# Bayesian Optimal Experimental Design for Vaccine Trials

Joshua Bean

April 14, 2022

*Thesis submitted for the degree of*

*Masters of Philosophy*

*in*

*Applied Mathematics*

*at The University of Adelaide*

*Faculty of Engineering, Computer and Mathematical Sciences*

*School of Mathematical Sciences*


THE UNIVERSITY
*of* ADELAIDE

# Contents

# Signed Statement

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint award of this degree.

I give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Signed: ..... Date: .....

# Acknowledgements

First of all I would like to thank my supervisors, Professor Joshua Ross and Associate Professor Gary Glonek for their constant help and encouragement. If it weren't for their advice and wealth of knowledge this thesis would not be near where it is today, especially in such uncertain times as 2020.

Second I would like to thank my loving and supportive parents who were always happy to talk to me or let me vent, keep me motivated, and well fed. Similarly to my Anma who is a never ending source of positivity, encouragement and delicious food. To my brother Simon, who is always proud of my achievements and work while not understanding why I bother doing 'so much work'.

Finally to my wife Miriam, whom not only got me through my undergraduate degree with support, companionship and just the right amount of competition. Moving into postgraduate research, she became an essential part of my research team by encouraging me to work through the difficult stages with love and support.

# Chapter 1

# Introduction

Between 2014 and 2016, West Africa experienced a devastating outbreak of the Ebola Virus Disease (EVD) that resulted in over 11,000 deaths across Guinea, Liberia and Sierra Leone [1]. The 2014-2016 outbreak was the most deadly EVD outbreak in history, and the rapid spread resulted in a world-wide humanitarian effort to help the affected countries. One line of defence against the outbreak was the development of an effective vaccine.

On 29$^{th}$ September 2014, the World Health Organisation (WHO) convened in Geneva to discuss two potential vaccines to the EVD [15]. It was concluded that after the vaccine passed early safety trials, the vaccine effectiveness would be tested in a full scale trial in Guinea with nearly 12,000 participants. The vaccine was administered in a ring formation where contacts of infected individuals or those at risk of infection were vaccinated [13]. This trial showed the vaccine was effective and the outbreak ended soon after. It is unclear whether the vaccine ended the outbreak or an increase in awareness of the disease and how it spread led to its control. The vaccine trial was implemented late into the outbreak and this highlighted a need for expedited trials and hence a framework for quickly finding an optimal vaccine trial design.

This scenario formed the motivation of this research; creating models for

a novel disease to investigate methods for finding an adaptive trial design that is optimal. However, mid-way through this research project, another fatal disease broke out, although this time globally, the SARS-CoV-2 2019 virus. This virus spread, ravaging the world population and, in some cases, effectively shut down whole countries' economies. As such, there has been a lot of interest in finding effective vaccines and trialling them on the general population. Like with the EVD outbreak, there was much discussion about expedited trials on the general population. These would have the goals of both learning about the vaccine efficacy itself and slowing and eventually stopping the spread of the disease.

The COVID-19 global pandemic in 2020 had disastrous effects globally, causing a massive loss of life and completely shutting down economies. However, its timing was fortuitous with regards to this research project. We decided to improve on many assumptions made in the previous work based on the EVD outbreak, including increasing the complexity of the disease model to more closely match a COVID model [26]. However, by increasing the complexity of the disease model considered, we quickly discovered computing issues. We found that the exact methods of estimating model parameters was many orders of magnitudes too slow to get a meaningful result and this led to the development of a new method. We adapted an existing method to get a nearly 10,000 times improvement in computation speed for this specific problem.

As mentioned above, the EVD outbreak highlighted a need to expedite vaccine trials, where the vaccine can be tested on the infected population while slowing and eventually ending the spread of the disease. So we constructed adaptive vaccine trials, where the trial begins with all participants in the control group and members of the population are vaccinated gradually. As we learn about the vaccine efficacy, we adjust the number of participants vaccinated each day accordingly. We first considered a simple model, where

individuals start as susceptible to the disease, then upon infection become infectious (able to spread the disease). Once infected, individuals remain infectious for a time with known mean. They then either recover from the disease or die. Since we were labelling individuals as belonging to the control or vaccinated group in the model, this led to a two-arm model, where one arm tracked the control group and the other arm tracked the vaccinated group. In order for the design to learn about the vaccine, we recorded all the information (the status of every compartment and the time and nature of each transition) from the model over a day and then updated the knowledge in the form of the posterior distribution of the vaccine effectiveness. Based on this posterior distribution we used four design parameters to decide the number of individuals to vaccinate. The first design parameter is the number of individuals initially vaccinated at the onset of the trial. The second is the number of individuals vaccinated if the vaccine is deemed beneficial. The third and fourth are a cutoff which the vaccine efficacy must exceed to be considered beneficial, and a threshold for the probability that the efficacy exceeds this cutoff. A vaccine is deemed beneficial if the probability that the vaccines effectiveness is less than the cutoff, exceeds this threshold; where the probability is estimated from the posterior distribution. Note that in this model, we have assumed the vaccine is multiplicative on the infection rate and that it can range from 0 to 2. Where 0 corresponds to a 100% effective vaccine and 2 corresponds to a vaccine that doubles the rate of infection for vaccinated individuals. Although, we would normally assume that vaccines at the trial phase would be at worst harmless, this would lead the optimisation to conclude any vaccine is acceptable. We ultimately want to compare the efficacy of two vaccines so we must consider harmful vaccines in order to find the optimal solution.

In an attempt to ensure computational feasibility, we made many assumptions about the model, some of which are not overly realistic or practical.

To improve the model, we pivoted to a COVID-19 based model with model parameters informed by data and that modelled how the disease progresses through individuals more accurately. One of the most significant changes was the assumption about the vaccine effect; we assumed the vaccine ranges from 0 to 1. This change assumes the vaccine is not dangerous to the individual and is at most ineffective, which aligns better with the safety precautions that would be taken. However, this modification meant that there was no disadvantage to vaccinating every individual from the onset, and to account for this we decided to model two competing vaccines simultaneously. These designs would be useful in the earlier stages of the outbreak where there are two safe candidate vaccines however their efficacy is unknown. This further increased the complexity of the model and hence slowed down the computation even more. Another crucial assumption we changed is that we only observe partial data from the outbreak, and this led to the need for faster computation methods. We found that a common method for estimating the likelihood function for partially-observed data was intractably slow for such a complex model in the context of optimal experimental design. As such, we took the principles from another method of estimation and through further approximations we were able to obtain a roughly 1000 times increase in speed for a minimal reduction in precision.

We have explored optimal experimental design for vaccine trials and developed a framework and understanding on a simple model in Chapter 3. In Chapter 4 we looked into methods of likelihood estimation for partially-observed data, and developed a new method that is orders of magnitude faster for a minor sacrifice in accuracy. Chapter 5 is a culmination of these two chapters where we apply the new likelihood method on a more sophisticated model. In this chapter we are interested in comparing two vaccines simultaneously, where we administer the vaccine to two groups simultaneously and compare their outcome. This simultaneous comparison allows for

a design that administers more of the optimal vaccine which will decrease the negative impact of the virus. The outcome of such a trial will be finding the more effective of the two vaccines as well as an accurate estimate of the vaccines efficacy, while simultaneously reducing the impact of the virus on the population.

# Chapter 2

# Background

## 2.1 Continuous-Time Markov chains

A *continuous-time Markov chain* (CTMC) defined on a countable state space $\mathcal{S}$, is a time-dependent stochastic process, $\{X(t)\}_{t\geq 0}$, that obeys the Markov property. The Markov property is defined as

$$P(X(t+s) = j | X(s) = i, \{X(u) : 0 < u < s\}) = P(X(t+s) = j | X(s) = i),$$

for all $s, t, u \in [0, \infty)$ and $i, j, k \in \mathcal{S}$.

Note that although we have defined a CTMC on a countable state space, this thesis will focus on finite state space CTMCs.

A CTMC, $\{X(t)\}_{t\geq 0}$, is time-homogeneous if,

$$P(X(t+s) = j | X(s) = i),$$

is independent of $s$ for all $s, t \in [0, \infty)$ and $i, j \in \mathcal{S}$. That is, if

$$P(X(t+s) = j | X(s) = i) = P(X(t) = j | X(0) = i),$$

for all $s, t \in [0, \infty)$ and $i, j \in \mathcal{S}$.

Let $p_{ij}(t) = P(X(t) = j | X(0) = i)$, for $s, t \in [0, \infty)$ and $i, j \in \mathcal{S}$, be the *transition (probability) function* for the transition from state $i$ to state

$j$. That is, $p_{ij}(t)$ is the probability of the process being in state $j$ at time $t$, given it started in state $i$ at time 0. The *transition function* $P(t)$ is an $|\mathcal{S}| \times |\mathcal{S}|$ matrix, whose $(i,j)^{\text{th}}$ entry is $p_{ij}(t)$, for $s, t \in [0, \infty)$ and $i, j \in \mathcal{S}$.

We can combine time-homogeneity and the Markov property to obtain the *Chapman-Kolmogorov* equations.

**Theorem 2.1.1** (Chapman-Kolmogrov)**.** *For all $s, t \in [0, \infty)$,*

$$p_{ij}(t+s) = \sum_{k \in \mathcal{S}} p_{ik}(t) p_{kj}(s).$$

When dealing with CTMCs, we are are unable to observe the transition function, so we observe the generator matrix, denoted $Q$. This generator matrix contains the information about the rates of transitioning between states as the process evolves. The elements of $Q$ are defined as follows,

$$\begin{aligned}
q_{ij} &= \lim_{t \to 0^+} \frac{P(X(t+s) = j | X(s) = i) - P(X(s) = j | X(s) = i)}{t} \\
&= \lim_{t \to 0^+} \frac{P(X(t+s) = j | X(s) = i)}{t} \\
q_{ii} &= \lim_{t \to 0^+} \frac{P(X(t+s) = i | X(s) = i) - P(X(s) = i | X(s) = i)}{t} \\
&= \lim_{t \to 0^+} \frac{P(X(t+s) = i | X(s) = i) - 1}{t}.
\end{aligned}$$

Note the quantities $q_{ii} \leq 0$ and $q_{ij} \geq 0$ for all $i, j \in \mathcal{S}$. The entries $q_{ij}$ $i \neq j$, represent the rate of transition of the process from state $i$ to state $j$. Similarly, the diagonal entries $q_{ii}$ represent the total rate of leaving state $i$.

The generator matrix $Q$, on a state space $\mathcal{S}$, is a matrix $Q = [q_{ij}]_{i,j \in \mathcal{S}}$ satisfying the following conditions:

1. $-\infty \leq q_{ii} \leq 0$, for all $i \in \mathcal{S}$;
2. $q_{ij} \geq 0$, for all $i \neq j$; and
3. $\sum_{j \in \mathcal{S}} q_{ij} \leq 0$, for all $i \in \mathcal{S}$.

Note the inequality in the third condition becomes an equality if the CTMC is *conservative*.

A CTMC has exponential holding times. That is to say that for every state the CTMC visits, it stays in that state for a time sampled from an exponential distribution with rate given by $q_i = -q_{ii}$;

$$P(X(s) = i, s \in [0, t] | X(0) = i) = e^{-tq_i}$$

Given the holding time has expired, then the distribution of where the process transitions to, is as follows,

$$P(\text{transitions to state } j \neq i | \text{transitions out of state } i) = \frac{q_{ij}}{q_i}.$$

Similarly, we know that the probability that the process transitions from state $i$ to state $j$, in a time interval of length $h$, is

$$P(X(h) = j | X(0) = i) = q_{ij}h + o(h),$$

where $o(h)$ is a function of $h$ such that $\lim_{h\to 0} \frac{o(h)}{h} = 0$.

The Chapman-Kolmogorov equations are required for deriving the following equations,

$$\frac{dP(t)}{dt} = P(t)Q \tag{2.1}$$

$$\frac{dP(t)}{dt} = QP(t) \tag{2.2}$$

known as the Kolmogorov forward and backward differential equations, respectively. These equations give a relationship between the generator matrix and the transition function $P(t)$. Note that the backwards equation always holds, however the forward equation holds if the state space is finite. All the CTMCs in this thesis have a finite state space, so we will assume it always hold. We can solve Equations (2.1) and (2.2) to get,

$$P(t) = e^{Qt}.$$

Note that this is a matrix exponential.

### 2.1.1 Simulating Continuous-Time Markov Chains

One of the common methods for simulating a realisation of a CTMC is, the stochastic simulation algorithm (SSA) [11]. This algorithm is outlined in Algorithm 2.1.

**Algorithm 2.1** Stochastic Simulation Algorithm

**Input:** Initial state $X(0)$, parameters $\boldsymbol{\theta}$, T

1: Initialise $t$
2: **while** $t \leq T$ **do**
3: Calculate the transition rates $q[j] = q_{ij}$, for all $j$ from the current state $i$ of the chain.
4: Sample the time of the next transition $t' \sim \text{Exp}(-q_{ii})$.
5: Increment $t = t + t'$.
6: For $k \in S$, sample a transition from the jump chain, having transition probabilities $p_{i \to k} = \dfrac{q_{ik}}{-q_{ii}}$.
7: Update the state, $X$, of the Markov Chain by transitioning from state $i$ to state $k$.
8: **end while**

**Output:** The time of each transition and the state of the Markov Chain after these transitions.

## 2.2 Epidemiological Models

We follow [14] for an overview of fundamental epidemiological models. In order to track the course of a disease through a population, we could track individuals as they progress through a series of *compartments*, where each compartment characterises important aspects of the disease. These models are referred to as compartmental models.

Compartmental models were first introduced by Kermack and McKendrick

[17] as deterministic models. In these models, the number of individuals in any given compartment is the solution to a set of ordinary differential equations (ODE). The individuals are classified into one of these compartments at any time throughout the duration of the epidemic. Some examples of these compartments are:

- $S$: where an individual is *Susceptible* to infection.
- $E$: where an individual is *Exposed* to the infection (infected but not infectious).
- $I$: where an individual is *Infectious* (able to infect susceptible individuals).
- $R$: where an individual has *Recovered* from being infectious (has immunity to the infection).

The choice of which compartments to include in the model will depend on both the characteristics of the disease, and what information we wish to glean from studying the outbreak. Consider the common cold, named rhinovirus. Transmission of a single strain can be modelled as an SIR model, see Section 2.2.2, as infectious individuals develop immunity to that strain upon recovery. However, there are many strains and the virus is continuously mutating, so immunity to one strain does not necessarily imply immunity to other strains. So, instead we might classify the common cold as an SIS, see Section 2.2.1, where upon recovery from the infection they are then susceptible to another strain of rhinovirus.

The work in this thesis is primarily concerned with CTMC representations of epidemic models; a special class of stochastic epidemic models (such as those considered by Barlett [3] and Kendall [16], for example). However, in Chapter 4 we utilise the deterministic representation of epidemic models,

which leads to faster estimation of the likelihood function. As the deterministic model can be used as an approximation of the mean of all realisations of the same CTMC model (this is shown to be true when the population is asymptotically large in Section 2.2.3), this can be used for a computationally fast method for approximate inference. Although the approximation is not true unless the population is asymptotically large, for the purpose of this project (where many impractical and sometimes unrealistic assumptions are made) it will suffice for demonstrating the methods.

### 2.2.1 Deterministic Models

In its simplest form, a deterministic epidemic model is a system of ODEs where each ODE describes the number of individuals in the population at given time, $t$. Since differential equations are continuous in both time and state, we think of these as modelling the proportion of the population in each compartment.

Throughout this thesis, we will be using models similar to the ones discussed in this chapter and unless otherwise mentioned we will be making the following fundamental modelling assumptions, as follows:

- The population is of a fixed size, that is to say there is no migration in or out of the population and similarly no births, nor deaths.
- The population is homogeneous, i.e., every individual is treated the same in terms of how they interact with the disease. In particular, every individual has the same probability of contracting the disease from an infected individual and they have the same level of infectiousness and are infectious on average for the same time.
- The population mixes homogeneously, where each individual has the same rate of interacting with another individual and hence the same probability of interacting with an infectious individual.

**An Example: The SIS model**

A Susceptible-Infectious-Susceptible (SIS) model could be used to model a disease such as some sexually-transmitted infections, where there is no period of immunity and the individual immediately returns to being susceptible to the virus upon recovery. The deterministic epidemiological models are intuitive as they are simply the difference between the flux into and out of each compartment. Let $t$ represent the time variable in the ODE and $s(t)$ be the proportion of the population that are susceptible at time $t$ and let $i(t)$ represent the proportion of the population that are infectious at time $t$. Let $\beta$ be the transmission parameter (the rate at which a susceptible individual becomes infectious when they interact with an infectious individual) and $\frac{1}{\gamma}$ be the average infectious period. Then the system of ODEs is,

$$\frac{ds}{dt} = -\beta is + \gamma i$$
$$\frac{di}{dt} = \beta is - \gamma i.$$

Since this is a very simple model an exact solution can be calculated. However in models used throughout this thesis, it is not possible to find exact solutions. Instead we rely on using numerical differential equation solvers.

### 2.2.2 Stochastic Models

The fundamental difference between deterministic and stochastic epidemic models is that the deterministic model does not capture the randomness that is accounted for in stochastic models. This is particularly significant at the beginning of the outbreak as a deterministic model does not capture the probability of the disease fading out before it takes off in the population. Note that while a deterministic approach models the proportion of the popu-

lation in each compartment and hence is continuous, a CTMC model counts individuals in each compartment and as such has a discrete state space. Both models are continuous in time.

### An example: SIR model

Note that the SIS model used an an example for deterministic models also has an analogous CTMC model, but to avoid repetition and provide a wider array of examples, we use a slightly different example here.

The Susceptible-Infectious-Recovered (SIR) model is representative of a disease with immediate infectiousness upon infection and with complete and permanent immunity upon recovery. An SIR model would be appropriate for modelling diseases that result in effectively life-long immunity after infection, such as measles, mumps or rubella. The homogeneous population of individuals is divided into three compartments defined by the infection status of each individual: *Susceptible* (S), *Infectious* (I), or *Recovered* (R).

The model makes the same assumptions as mentioned above. A diagram of the three compartments and the transitions between them are in Figure 2.1.

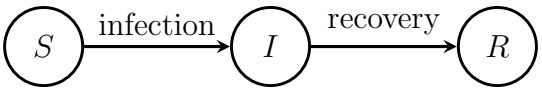


Figure 2.1: Compartmental diagram for an SIR model.

Let $S(t)$ and $I(t)$ represent the number of susceptible and infectious individuals in the population at time $t$. Let $N$ be the fixed total population size. Then $R(t) = N - S(t) - I(t)$ is the number of recovered individuals. The CTMC, $\{S(t), I(t)\}_{t\geq 0}$, models the SIR process with state space,

$$\mathcal{S} = \{(S, I) | S, I \in \mathbb{N}, 0 \leq S + I \leq N\},$$

where $\mathbb{N}$ is the set of non-negative integers. The transition rates from the state (S,I), are given in Table 2.1 below.

Table 2.1: SIR compartmental model transition rates, where $\beta$ is the transmission parameter, and $\frac{1}{\gamma}$ is the average infectious period.

| Event | Transition | Rate |
|---|---|---|
| Infection | $(S, I) \to (S-1, I+1)$ | $\frac{\beta IS}{N-1}$ |
| Recovery | $(S, I) \to (S, I-1)$ | $\gamma I$ |

Note in Table 2.1 the rate of infection is $\frac{\beta IS}{N-1}$. This is because the probability of a susceptible individual coming into contact with an infected individual is $\frac{S}{N-1}$ if there are $S$ infected people in the population and $N$ people in the population, as we exclude the susceptible person who is making the contact. It is important to note this is not consistent with deterministic models, but that is because they are equivalent to the mean of a CTMC model if the population is large enough (Section 2.2.3), where $\frac{1}{N} \approx \frac{1}{N-1}$.

### SEIR model

The Susceptible-Exposed-Infectious-Recovered (SEIR) model is an extension to the SIR model above, where we have added an *Exposed* (E) compartment that represents individuals that have contracted the disease but are not yet infectious. The inclusion of the Exposed state allows the model to include an incubation period (time from infection to infectiousness).

Let $S(t)$, $E(t)$ and $I(t)$ represent the number of susceptible, exposed and infectious individuals in the population at time $t$, and $R(t) = N - S(t) - E(t) - I(t)$, the recovered individuals. The CTMC, $\{S(t), E(t), I(t)\}_{t\geq 0}$, models the SEIR process with state space,

$$\mathcal{S} = \{(S, E, I) | S, E, I \in \mathbb{N}, 0 \leq S + E + I \leq N\}.$$

The transition rates from the state (S,E,I), are given in Table 2.2 below.

Table 2.2: SEIR compartmental model transition rates, where $\beta$ is the transmission parameter, $\frac{1}{\sigma}$ is the average incubation period, and $\frac{1}{\gamma}$ is the average infectious period.

| Event | Transition | Rate |
|---|---|---|
| Exposure | $(S, E, I) \to (S-1, E+1, I)$ | $\frac{\beta IS}{N-1}$ |
| Infection | $(S, E, I) \to (S, E-1, I+1)$ | $\sigma E$ |
| Recovery | $(S, E, I) \to (S, E, I-1)$ | $\gamma I$ |

### 2.2.3 Large-Population Approximation

As already discussed, we know intuitively that the mean of the stochastic models is asymptotically given by the deterministic models as the number of individuals in the population tends towards $\infty$. However in Chapter 4 we require the variance of these stochastic models. We take the work of Pollett *et al.* [24] that furthered work by Kurtz [20] on the single-dimensional deterministic approximation. They show that the mean and variance of a *density-dependent* stochastic model is given by a system of differential equations on a single-dimensional problem and mention that it can be easily extended to a multi-dimensional model. We follow [27] for the extension to multi-dimensional models.

**Definition 2.2.1** (Population process)**.** $\{\boldsymbol{X}(t)\}_{t\geq 0}$ is a population process if:

1. Each state $\boldsymbol{x}$ partitions a finite population of $N$ individuals into a finite number of compartments.

2. The only positive transition rates $q_{\boldsymbol{x},\boldsymbol{x}+\boldsymbol{\ell}}$, are ones for which $\boldsymbol{\ell}$ is either

$$\pm\boldsymbol{e}_i, \qquad \text{or} \qquad \boldsymbol{e}_i - \boldsymbol{e}_j,$$

where $\boldsymbol{e}_i$, is the unit vector with 1 as its $i^{\text{th}}$ element.

The CTMC models considered in this thesis track each individual in the population and as such may be thought of as a population process. Thus, the elements of the state space of the CTMC denote the number of individuals in each compartment, and the possible transitions of the CTMC reflect the event that an individual arrives/departs from compartment $i$ ($\pm\boldsymbol{e}_i$), or an individual transitions from compartment $j$ to compartment $i$ ($\boldsymbol{e}_i - \boldsymbol{e}_j$).

Most CTMC models used in epidemiology are population processes in which the compartments reflect an individual's stages of infection and the transitions of the model represent events such as an individual becoming infectious or an individual recovering.

The definition of *density dependence* refers to the family of CTMCs $\{\boldsymbol{X}^{(\nu)}(t)\}_{t\geq 0}$, $\nu \geq 0$, which takes values in $\mathcal{S}^{(\nu)}$, the set of all possible states of the process $\boldsymbol{X}^{(\nu)}$. This is a way of constructing a relationship between the CTMC, $\{\boldsymbol{X}(t)\}_{t\geq 0}$ and a particular *scaling parameter* $\nu \geq 0$; $\nu$ usually represents some measure of the size of the system, for epidemic models we will take $\nu$ to be the known population size $N$.

**Definition 2.2.2** (Density Dependence)**.** Suppose $\{\boldsymbol{X}^{(\nu)}(t)\}_{t\geq 0}$, $\nu \geq 0$, has a corresponding family of continuous functions $f^{(\nu)}(\boldsymbol{x}, \boldsymbol{\ell})$, for $\boldsymbol{x}$ in $E$ with $E \subseteq \mathbb{R}^K$ and $K \in \mathbb{Z}_+$, such that

$$q^{\boldsymbol{X}^{(\nu)}}_{\boldsymbol{y},\boldsymbol{y}+\boldsymbol{\ell}} = \nu f^{(\nu)}\left(\frac{\boldsymbol{y}}{\nu}, \boldsymbol{\ell}\right),$$

for all $\boldsymbol{y}$ in $\mathcal{S}^{(\nu)}$ and $\boldsymbol{\ell} \neq \mathbf{0}$; in which case, the stochastic process $\{\boldsymbol{X}^{(\nu)}(t)\}_{t\geq 0}$ is density dependent. Define the vector

$$F^{(\nu)}(\boldsymbol{x}) = \sum_{\boldsymbol{\ell}} \boldsymbol{\ell} f^{(\nu)}(\boldsymbol{x}, \boldsymbol{\ell}).$$

The family of CTMCs is said to be asymptotically density dependent if there exists a function $F(\boldsymbol{x})$ such that

$$\lim_{\nu\to\infty} F^{(\nu)}(\boldsymbol{x}) = F(\boldsymbol{x}).$$

In other words, a density dependent CTMC is one whose transition rates depend on the current state $\boldsymbol{y}$ only through the density $\boldsymbol{y}/\nu$. It can be shown that the stochastic process $\boldsymbol{X}^{(\nu)}(t)/\nu$, for $t \geq 0$, as $\nu \to \infty$, converges uniformly in probability (over finite time intervals) to a unique deterministic trajectory $\boldsymbol{x}(t, \boldsymbol{x}_0)$ with time derivative $F(\boldsymbol{x})$ and $\boldsymbol{x}_0$ in $E$.

**Theorem 2.2.1.** *Suppose that $F(\boldsymbol{X})$ is Lipschitz continuous on $E$, then if*

$$\lim_{\nu\to\infty} \frac{\boldsymbol{X}^{(\nu)}(0)}{\nu} = \boldsymbol{x}_0,$$

*we have that for finite $t$,*

$$\lim_{\nu\to\infty} P\left(\sup_{s\leq t} \left| \frac{\boldsymbol{X}^{(\nu)}(0)}{\nu} - \boldsymbol{x}(t, \boldsymbol{x}_0) \right| > \varepsilon \right) = 0, \qquad 0 \leq s \leq t$$

*for all $\varepsilon > 0$ , and for every trajectory $\boldsymbol{x}(\cdot, \boldsymbol{x}_0)$ satisfying*

$$\begin{aligned} \boldsymbol{x}(0, \boldsymbol{x}_0) &= \boldsymbol{x}_0 \\ \boldsymbol{x}(s, \boldsymbol{x}_0) &\in E, \qquad 0 \leq s \leq t \\ \frac{\partial}{\partial s}\boldsymbol{x}(s, \boldsymbol{x}_0) &= F(\boldsymbol{x}(s, \boldsymbol{x}_0)). \end{aligned}$$

This shows that the time derivative of the unique deterministic trajectory is given by $F(\boldsymbol{x}(s, \boldsymbol{x}_0))$, which aligns with the intuitive description given in Section 2.2.1. This approximation describes the average behaviour of the corresponding density-dependent stochastic model, but provides no indication of the variability, so we look to the diffusion limit.

**Theorem 2.2.2** (Diffusion Limit)**.** *Suppose that $F(\boldsymbol{x})$ is bounded and Lipschitz continuous on $E$. Suppose also that the family of continuous functions $G^{\nu}(\boldsymbol{x})$, where $\nu > 0$ and $\boldsymbol{x}$ is in $E$, is a $K \times K$ matrix, where $dim(\boldsymbol{X}^{(\nu)}(t)) = K$, with elements*

$$q_{ij}^{(\nu)}(\boldsymbol{x}) = \sum_{\boldsymbol{\ell}} \ell_i \ell_j f^{(\nu)}(\boldsymbol{x}, \boldsymbol{\ell}).$$

*Here, $\ell_i$ denotes the $i^{th}$ entry of the vector $\ell$, which converges uniformly to $G(\boldsymbol{x})$, where $G(\boldsymbol{x})$ is bounded and uniformly continuous on $E$. Then provided,*

$$\lim_{\nu \to \infty} \sqrt{\nu} \left( \frac{\boldsymbol{X}^{(\nu)}(0)}{\nu} - \boldsymbol{x}_0 \right) = z,$$

*for $\boldsymbol{x}_0$ in $E$. The family of Markov processes $\boldsymbol{Z}^{(\nu)}(t)$, for $t \geq 0$, defined by*

$$\boldsymbol{Z}^{(\nu)}(s) = \sqrt{\nu} \left( \frac{\boldsymbol{X}^{(\nu)}(s)}{\nu} - \boldsymbol{x}(s, \boldsymbol{x}_0) \right), \qquad 0 \leq s \leq t,$$

*converges weakly in $D[0,t]$ (the space of right-continuous, left-hand limited function on $[0,1]$) to a diffusion process, $\boldsymbol{Z}(t)$, with initial value $\boldsymbol{Z}(0) = \boldsymbol{z}$ and with characteristic function, $\Psi = \Psi(s, \boldsymbol{\theta})$ which satisfies*

$$\frac{\partial}{\partial s}[\Psi(s,\boldsymbol{\theta})] = -\frac{1}{2} \sum_{j,k} \boldsymbol{\theta}_j g_{jk}(\boldsymbol{x}(s,\boldsymbol{x}_0) \boldsymbol{\theta}_k \Psi(s,\boldsymbol{\theta}) + \sum_{j,k} \boldsymbol{\theta}_j \frac{\partial}{\partial x_k}[F_j(\boldsymbol{x}(s,\boldsymbol{x}_0))] \frac{\partial}{\partial \boldsymbol{\theta}_k}[\Psi) s, \boldsymbol{\theta})].$$

The theorem then stipulates that a $\sqrt{\nu}$ scaling of the difference $\boldsymbol{X}^{(\nu)}(t)/\nu - \boldsymbol{x}(s, \boldsymbol{x}_0)$ converges weakly in $D[0,t]$ to the diffusion process $\boldsymbol{Z}(t)$ as $\nu \to \infty$.

The following is taken verbatim from [27], which provides an explicit expression for the variance of this Gaussian diffusion and fully describes its characteristics.

*In particular, if one denotes $\nabla F(\boldsymbol{x})$ as the matrix of first partial derivatives of $F(\boldsymbol{x})$, that is $[\partial F_i / \partial x_j]$, and denotes $B(t) = \nabla F(\boldsymbol{x}(s, \boldsymbol{x}_0))$, then $E[\boldsymbol{Z}(t)] = M(t)\boldsymbol{z}$, where $M(t)$ is the unique solution to $dM(t)/dt = B(t)M(t)$, with initial value $M(0) = \boldsymbol{I}$. Similarly, the covariance matrix $cov(\boldsymbol{Z}(t)) = \Sigma(t)$ is the unique solution to,*

$$\frac{d\Sigma(t)}{dt} = B(t)\Sigma(t) + \Sigma(t)B(t)^T + G(\boldsymbol{x}(s, \boldsymbol{x}_0)), \tag{2.3}$$

*with* $\Sigma_0 = 0$.

*Equation 2.3 provides us with a system of differential equations, whose unique solution is the covariance matrix for the multi-dimensional diffusion process.*

This will be utilised in Chapter 4.

### Our epidemiological models are density dependent

The large-population approximation theory is only applicable if the CTMC is density dependent. All epidemic models used in this thesis are simply extensions of an SEIR model, by including more compartments that function similarly to those in the base model. We will focus on showing that each transition in the SEIR model is density dependent and hence including more compartments that function similarly will still result in a density-dependent model. To show that a model is density dependent, we must be able to write the transition rates satisfying Definition (2.2.2). In the SEIR model there are two types of transition rates, the first is the rate of exposure which is a constant multiplied by the mass action term, the second is a constant multiplied by the number of individuals in a compartment, for instance a recovery. First, we will consider an SEIR model with fixed population size $N$, where the exposure rate is given by,

$$q_{(S,E,I,R)\to(S-1,E+1,I,R)} = \frac{\beta SI}{N-1},$$

where $\boldsymbol{x} = (S, E, I, R)$ describes the number of individuals in each compartment. Note that we can describe this transition as $\boldsymbol{x} + \boldsymbol{\ell}$, where $\boldsymbol{\ell} = (-1, 1, 0, 0)$. If we take the fixed population size to be the scaling parameter $\nu$ and let $s = S/N$ and $i = I/N$, we can write this transition in the following

form,

$$q_{\boldsymbol{x},\boldsymbol{x}+\boldsymbol{\ell}} = N\frac{N}{N-1}\beta si.$$

However since this theory only holds for asymptotically density-dependent models, at the limit as $N \to \infty$, the ratio $N/(N-1)$ converges to 1, so this specific transition can be written in the form outlined in Definition (2.2.2). The other transition we need to consider is the rate of infection, this rate is expressed as,

$$q_{(S,E,I,R)\to(S,E-1,I+1,R)} = \sigma E,$$

For this transition, $\boldsymbol{\ell} = (0,-1,1,0)$, and by letting $e = E/N$, we get the following,

$$q_{\boldsymbol{x},\boldsymbol{x}+\boldsymbol{\ell}} = Ne.$$

This once again satisfies the condition for density dependence. In this model there is one last transition, the recovery, however it has the exact same structure as the infection discussed above, so it will satisfy the condition. Thus the SEIR model is indeed density dependent as all the transitions in the model satisfy the conditions for Definition (2.2.2). In this thesis every model we use will be extensions on this model, where the transition rates have the same structure as the ones discussed above. Therefore we know that all models used in this thesis are density dependent and hence we can use the result in Equation 2.3 to estimate the variance of our stochastic models.

## 2.3 Bayesian Inference

There are two main frameworks for statistical inference, frequentist and Bayesian. The frequentist approach considers the parameters as fixed but unknown. The Bayesian framework assumes the parameters are random variables. Further, the Bayesian approach assumes the prior knowledge about the parameters is described by a prior distribution, and also assumes that the

prior information can be combined with the data by assuming the parameter to be a realisation from the prior distribution. Here we provide some theory relating to both frameworks, however the research in this thesis is focused strongly on Bayesian inference.

In both the frequentist and Bayesian frameworks, a key function is the *likelihood function*. Following Casella and Berger [6], we provide the definition of the likelihood function. Let $\boldsymbol{\theta}$ be a vector of length $p$ of parameters $(\theta_1, \theta_2, \ldots, \theta_p)$.

**Definition 2.3.1.** Let $f(\boldsymbol{x}|\boldsymbol{\theta})$ denote the joint probability density function (pdf) or probability mass function (pmf) of the sample $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$, conditioned on the parameters $\boldsymbol{\theta}$. Then, given that $\boldsymbol{X} = \boldsymbol{x}$ is observed, the function of $\boldsymbol{\theta}$ defined by

$$L(\boldsymbol{\theta}|\boldsymbol{x}) = f(\boldsymbol{x}|\boldsymbol{\theta}),$$

is called the *likelihood function*.

This likelihood function describes the probability that $\boldsymbol{\theta}$ is the value of the parameter that produced the observed data $\boldsymbol{x}$.

### 2.3.1 Fundamentals

In Bayesian statistics we aim to make inference about the parameters, using information from two sources. We use the initial belief about the parameter values, which is represented in the *prior distribution*, and then obtain data from experimentation, which gives information about the parameters in the form of the likelihood function. Bayesian statistics provides a method for combining these two sources of information, to update the belief about the parameters, which is called the *posterior distribution*. We follow Gelman *et al.* [10] for the following definitions and theorems.

**Theorem 2.3.1.** *(Bayes' Rule). Let $X$ and $Y$ be continuous random variables with corresponding pdfs $f_X(\cdot)$ and $f_Y(\cdot)$. Then*

$$f_{X|Y}(x|y) = \frac{f_X(x)f_{Y|X}(y|x)}{f_Y(y)}. \tag{2.4}$$

In Equation (2.4), let $Y$ be the random variable representing the data, hence $\boldsymbol{y}$ is the observed data. Similarly, let $X$ be the random variable representing the parameters, $\boldsymbol{\theta}$. This gives the following definition of the posterior distribution.

**Definition 2.3.2.** The posterior distribution is given by,

$$\begin{aligned} f(\boldsymbol{\theta}|\boldsymbol{y}) &= \frac{f(\boldsymbol{\theta})f(\boldsymbol{y}|\boldsymbol{\theta})}{f(\boldsymbol{y})} \\ &\propto f(\boldsymbol{\theta})f(\boldsymbol{y}|\boldsymbol{\theta}). \end{aligned} \tag{2.5}$$

In Definition 2.3.2, $f(\boldsymbol{\theta})$ represents the prior information about the parameters, $\boldsymbol{\theta}$. Similarly, $f(\boldsymbol{y}|\boldsymbol{\theta})$ represents the likelihood of the parameters $\boldsymbol{\theta}$, given the data, $\boldsymbol{y}$.

The posterior distribution, $f(\boldsymbol{\theta}|\boldsymbol{y})$, represents the belief about the distribution of parameters, $\boldsymbol{\theta}$, by updating the belief from the prior distribution using the new information from the observed data, $\boldsymbol{y}$. We usually refer to the proportional expression (Equation (2.5)) as the posterior distribution, since evaluating the normalising constant, $f(\boldsymbol{y}) = \int_{\boldsymbol{\theta}} f(\boldsymbol{\theta})f(\boldsymbol{y}|\boldsymbol{\theta})d\boldsymbol{\theta}$, is often impossible or computationally infeasible. In the following we cover computational methods for sampling from the un-normalised posterior distribution.

### 2.3.2 Computational Methods

Direct evaluation of the posterior distribution is often not possible for complex processes, hence we require computational techniques. Here, we discuss a fundamental method for estimating the posterior distribution, the

*Metropolis-Hastings algorithm.* This method is an example of Markov chain Monte Carlo (MCMC), that samples from a target function, proportional to the posterior density (i.e., $f(\boldsymbol{\theta})f(\boldsymbol{y}|\boldsymbol{\theta})$). We discuss the theory and implementation of this method.

### Metropolis-Hastings Algorithm

The Metropolis algorithm was outlined by Nicholas Metropolis in [21]; Wilfred Hastings took this principal and extended it in [12], to create the Metropolis-Hastings algorithm. The Metropolis-Hastings algorithm allows us to sample from a target density (known up to normalisation), by generating a discrete-time Markov chain (referred to as the Metropolis-Hastings chain) with stationary distribution equal to the target distribution. At each step we propose a new state 'near' the previous state, from the proposal density $q(\boldsymbol{x}, \boldsymbol{y})$ which is the probability of proposing $\boldsymbol{y}$ given the current state $\boldsymbol{x}$, then decide whether to accept this proposed state based on an acceptance probability. In our case the target distribution is the posterior distribution so we can use the un-normalised posterior distribution given in Equation (2.5). Note that the algorithm will always yield a sample from the target distribution, provided the initial sample is from the domain of the target distribution. The Metropolis-Hastings algorithm is given in Algorithm 2.2.

The Metropolis-Hastings chain is guaranteed to reach the stationary distribution, however this may take a number of iterations. These early samples are not necessarily from the stationary distribution, so they are discarded and we refer to them as burn-in samples. The number of burn-in samples is one of many choices that need to be made when tuning the Metropolis-Hastings algorithm.

The Metropolis-Hastings algorithm can be proven to always converge to the target distribution, regardless of choice of proposal density. Hence, the proposal density does not affect the stationary distribution of the Metropolis-

**Algorithm 2.2** Metropolis-Hastings Algorithm

**Input:** Proposal density $q(\cdot,\cdot)$, parameters $\boldsymbol{\theta}$, number of samples $N$, burn-in samples $m$.

1: Randomly sample, from the prior, an initial sample $\boldsymbol{\theta}^{(0)}$
2: **for** $i = 1 : (N + m)$ **do**
3: Generate $\boldsymbol{\theta}^* \sim q(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(i-1)})$
4: Calculate $\alpha = \min\left\{1, \frac{f(\boldsymbol{\theta}^*)f(\boldsymbol{x}|\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^{(i-1)},\boldsymbol{\theta}^*)}{f(\boldsymbol{\theta}^{(i-1)})f(\boldsymbol{x}|\boldsymbol{\theta}^{(i-1)})q(\boldsymbol{\theta}^*,\boldsymbol{\theta}^{(i-1)})}\right\}$
5: Generate $u \sim U(0, 1)$
6: **if** $u < \alpha$ **then**
7: Set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^*$
8: **else**
9: Set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$
10: **end if**
11: **end for**
12: Discard first $m$ burn-in samples $\{\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(m)}\}$

**Output:** Sampled set of parameters $\{\boldsymbol{\theta}^{(m+1)}, \ldots, \boldsymbol{\theta}^{(N+m)}\}$.

Hastings chain, as long as it meets some basic criteria such as it admits the resulting Markov chain is reversible. However, choosing a suitable proposal density is still important as it affects the speed of convergence and mixing of the chain. Generally, the algorithm performs best when the proposal density matches the shape of the target density. If the proposal density is symmetric, the ratio in the acceptance probability can be simplified to,

$$\alpha = \min\left\{1, \frac{f(\boldsymbol{\theta}^*)f(\boldsymbol{x}|\boldsymbol{\theta}^*)}{f(\boldsymbol{\theta}^{(i-1)})f(\boldsymbol{x}|\boldsymbol{\theta}^{(i-1)})}\right\},$$

as $q(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(i-1)}) = q(\boldsymbol{\theta}^{(i-1)}, \boldsymbol{\theta}^*)$. This is the simpler acceptance probability outlined by Metropolis [21].

A simple way to think about the choice of proposal density is to consider the variance of the distribution centred on the current value in the chain. A distribution with large variance will propose values far from the current

value, but may result in a small acceptance probability, thus rejecting many proposed parameters. This can also induce autocorrelation as the chain may remain in each state for many steps until a proposal is accepted. A small variance leads to proposed values that are closer to the current value, and thus likely to have a larger acceptance probability; however, the chain will take much longer to fully explore the state space. A smaller variance may also induce autocorrelation as the proposed values are closely located to the previous values. This induced autocorrelation will cause us to underestimate the variance in the sample mean. If there is significant autocorrelation in the chain, then the *effective sample size* (ESS) of the final sample will be smaller than the actual number of samples.

If the samples from the Metropolis Hastings chain were completely independent, then the effective sample size would be the same as the number of samples. However as the dependence in the sample increases, the estimate becomes less accurate and so more samples are required. We use ESS as a measure of the equivalent number of independent samples, compared the number of dependent samples. That is, as the dependence in the sample increases, the ESS decreases. The ESS of a sample from an MCMC algorithm defined by [8] is defined as,

$$\text{ESS} = \frac{N}{1 + 2\sum_{k\geq 0} \rho_k(\boldsymbol{\theta})},$$

where, $\rho_k(\boldsymbol{\theta})$ is the autocorrelation at lag $k$ of $\{\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(N)}\}$. Note that the autocorrelation of the Metropolis-Hastings chain must be calculated at each lag before the ESS can be calculated.

## 2.4 Particle Filter

The likelihood function is a common component in all the computation methods used for evaluating the posterior distribution that are considered in this

thesis. In many cases evaluation of the likelihood function is not straightforward and a number of different methods, each with advantages and disadvantages, can be applied. In Chapter 4 we will explore estimating the posterior distribution for parameters from partially-observed data of an outbreak. In this thesis we consider partially-observed data to be the observed number of particular events in a day. For instance, an SIR model might have partially-observed data as the number of recoveries that have occurred in a day.

Calculating the likelihood function for partially-observed data in an exact way is nearly impossible from a computational perspective, however it is quick and easy to simulate these outbreaks. This is where *particle filters* are very useful. In its simplest form, a particle filter will simulate a realisation from the stochastic process and see how 'well' it matches the observed data. If a particle matches the data over some time period (we will work in days) well, then that *particle* will have a large weight and will likely survive into the next generation. However, particles that do not match the data well, have a small weight, and are unlikely to survive. We then sample the next generation of particles from this weighted distribution to continue forward to the next day. We are able to use these weights after every day to estimate the likelihood of the observed data. Algorithm 2.3 outlines this particle filter method, from [9].

### 2.4.1 Bootstrap Particle Filter

The bootstrap particle filter is the simplest form of a particle filter, where each successful particle is equally weighted. In the bootstrap particle filter, propagating a particle forward a day (step 6), is done by using a stochastic simulation algorithm similar to Algorithm 2.1. Then, from this propagation, we count the number of the event type being observed, namely $z_t$. For an SIR model, $z_t$ could be the number of infections over a day, for example. Then

**Algorithm 2.3** Particle Filter Algorithm

**Input:** Number of particles, N, initial state, $x_0$, number of days in the outbreak, T, observed data for each day, $\boldsymbol{y}$.

1: Initialise N particles, $\boldsymbol{X}$
2: Initialise weights, $\boldsymbol{w}$ with $w_i = \frac{1}{N}$
3: Initialise log-likelihood, $\ell = \ln\left(\frac{1}{N}\right)$
4: **for** $t = 1 : T$ **do**
5:  **for** $n = 1 : N$ **do**
6:  Propagate particle $n$ forwards a day
7:  Calculate the weight, $w_n$, of particle $n$
8:  **end for**
9:  Update log-likelihood, $\ell \leftarrow \ell + \ln(\frac{1}{N}\sum_{i=1}^{N} w_i)$
10:  Normalise the weights, $\boldsymbol{w} \leftarrow \boldsymbol{w}/\sum_{i=1}^{N} w_i$
11:  Sample (with replacement) $N$ particles with weights $\boldsymbol{w}$
12: **end for**

**Output:** log-likelihood $\ell$

the weight of this particle (step 7) is simply 1 if this matches the observed data and 0 otherwise;

$$w_n = \begin{cases} 1 & \text{if } z_t = y_t \\ 0 & \text{otherwise.} \end{cases}$$

The bootstrap particle filter will produce an unbiased estimate of the likelihood function of the data. However as the model grows in complexity, the probability of simulating a specific trajectory that agrees with the data becomes less likely. To remedy this, we can increase the number of particles used in the filter, but this will quickly become computationally intractable. As such, we will use an importance sampling particle filter that uses the

principles of importance sampling to significantly decrease the variance in the estimate of the likelihood.

### 2.4.2 Importance Sampling

Importance sampling is a variance reduction technique, and has applications when estimating rare event probabilities [18]. The standard setting is the estimation of a quantity,

$$\ell = \mathbb{E}[h(\boldsymbol{X})] = \int h(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x},$$

where $h$ is a real-valued function and $f$ is the probability density. Let $g$ be another probability density such that $hf$ is dominated by $g$; that is, $g(\boldsymbol{x}) = 0 \Rightarrow h(\boldsymbol{x})f(\boldsymbol{x}) = 0$. Using the density $g$, we can represent $\ell$ as

$$\ell = \int h(\boldsymbol{x}) \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} g(\boldsymbol{x}) d\boldsymbol{x}.$$

This estimator is called the importance sampling estimator and $g$ is called the importance sampling density. The ratio of densities,

$$w(\boldsymbol{x}) = \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})},$$

is called the likelihood ratio.

Fundamentally, importance sampling takes a distribution and changes the shape of this distribution according to $g(\boldsymbol{x})$. We then take a sample from this modified distribution and, using the weights, convert this sample back to a sample from the original distribution. An application of this is to assist with sampling from the tails of distributions (rare events), and in the context of our problem, rare outcomes for a realisation.

We will follow [5] for an outline of importance sampling particle filters.

### 2.4.3 Importance Sampling Particle Filter

An importance sampling particle filter follows the same principle of the bootstrap particle filter, however the method of propagating forward the particle

and calculating the weights is different. Here we utilise importance sampling to ensure that every particle always matches the observed data over a day exactly and as such, will need fewer particles when the observed events become infrequent. For an importance sampling particle filter, we take Algorithm 2.3 and modify the propagation in step 6 by Algorithm 2.4.

We will explain the algorithm by following a simple example of an SIR model and we only observe infections in the population (i.e. $S \to I$) at daily intervals. Then the data we have will be $\{z_0, z_1, \ldots, z_T\}$, where we have $T+1$ days of observations. Say we are interested in the propagation step at time $t$, then we know the initial state, $\boldsymbol{X}(0)$ and observed events $z_t$. Initially, we will take a sample of size $z_t$ from a standard uniform distribution and sort them $\{u_0, u_1, \ldots, u_{z_t}\}$. These are the times that the observed events will happen in the simulation.

Now we continue to simulate the outbreak following the SSA (2.1), but once the simulation moves past time $u_0$, we disregard that suggested transition and instead force the system to make the observed transition, $S \to I$. This is repeated until all observed events have been forced. It is important to note that we can ensure there are exactly $z_t$ observed events on day $t$ by making the rate of the observed event (in the SSA) equal to 0, so the transition occurs only when forced.

In Algorithm 2.4 we refer to the system needing to be *consistent*. This can be an issue for complex models that have long chains of events. To observe this, consider an SEIR model, where the observed transition is $E \to I$ which is at end of a chain of events $S \to E \to I$. Once again, let the observed data on day $t$ be $z_t$. We sample from the standard uniform distribution the times of these events, $\{u_0, u_1, \ldots, u_{z_t}\}$. For a system to be consistent, we need to be able to force the observed event at the sampled time. For example, consider a day that starts in state $(S, E, I, R) = (N-1, 0, 1, 0)$, and we need to force an event at time $u_0$. This is not a consistent state because the

compartment $E$ has 0 individuals which means we are unable to force the observed transition. To fix this we have to force an interim transition, that exposes an individual so that the exposed individual can then be infected at time $u_0$. The time, $T$, of this interim forced event is sample from a truncated exponential distribution,

$$P(T = t \mid t < u_0, \lambda) = \lambda e^{-\lambda t} \left(1 - e^{-\lambda u_0}\right)^{-1},$$

where $\lambda$ is the rate of leaving the current state in the CTMC. The truncated exponential distribution ensures that the interim forced event occurs before the observed forced event at time $u_0$. This is then repeated until all observed forced events have occurred.

Within Algorithm 2.4, we mention modified rates of the system, which are used in the calculation of importance sampling weights. It is important to modify these rates because this can ensure the system does not reach an impossible state. As discussed, the rate of the observed event is always zero. Another issue is that the simulation within the particle filter cannot end before all observed events have happened. Thus we also modify the rates so that there are enough infected individuals in the population until all observed events have occurred. As the complexity of the model increases, the number of modifications required increases and the chains of interim forced events get longer. We will discuss in detail an implementation of the importance sampling particle filter on a sophisticated model with many such cases in Chapter 4.

## 2.5 Optimal Experimental Design

Experiments are important tools for learning about the world. However, even under controlled conditions, experiments are not exempt from random variation. As such, an efficient design of experiments is crucial for accurately understanding the process of interest.

Optimal experimental design is a branch of statistics that aims to determine the experimental design that maximises an experiments utility. Utility can be defined specifically for the problem at hand, but is fundamentally an integral over the unknown parameters in the trial. A design is a set of parameters that affect how the experiment will be run, so changing the design, will change the outcome of the experiment. These designs are drawn from a multidimensional space (dependent on the number of parameters in the design), that will be restricted in each dimension by the physical restrictions of the experiment. This space from which designs are drawn is called the *design space* and is specific to each experiment. The design that results in the maximum utility is referred to as the *optimal experimental design*.

### 2.5.1 Bayesian Optimal Experimental Design

Bayesian optimal experimental design, much like Bayesian inference, is concerned with the posterior distribution. The purpose of experimental design is to determine the optimal design when the sample has not yet been observed. In Bayesian experimental design, the optimal design is one that affects the posterior distribution in such a way that it optimises the utility. The utility is a measure calculated from the results of the experiment, usually from the posterior distribution. An example of this is the Kullback-Leibler Divergence (KLD) [19], which is a measure of how different the posterior distribution is compared to the prior distribution. This is just one example of a utility that is measure of how much the design learned about model parameters.

Traditional utilities that measure how well the design facilitates learning about the model were not the focus of this project. For the vaccine trials in this thesis, we are more interested in the clinical outcomes of the trials. As such, we will develop our own utilities that are calculated from the amount of negative impact that the disease had on the population. We discuss these utilities in detail in Chapters 3 and 4.

### Induced Natural Selection Heuristic

A method for exploring the design space in higher dimensions is outlined by Price *et al.* [25], where they propose the *Induced Natural Selection Heuristic* (INSH). The INSH algorithm is outlined in Algorithm 2.5.

In Algorithm 2.5, the perturbation function can either be continuous or discrete, or a combination of the two. This function takes a current design and moves it slightly around the design space, centered at the current design; similar to the principal of the proposal distribution in the Metropolis-Hastings algorithm. This is where the exploration occurs in the method, and the variance of this function needs to be tuned appropriately. Price *et al.* suggest some different acceptance criteria; for example, taking the set of designs that have a utility greater than some percentage of the current optimal utility or taking the best $r$ designs.

If the variance of the perturbation function is too small, the algorithm is likely to get 'stuck' in a local minimum and hence not properly explore the design space. Similarly, if the variance is too large, the perturbed design may be too far from the current optimal, resulting in inefficient convergence to the local minimum.

For our purposes, we will be using the acceptance criteria where we take the best $r$ designs in the current set and then perturb each of them $m$ times. Consider fixing the number of designs, $|D| = r \times m$. If $r$ is small, then at each generation we save fewer designs, which will cause the algorithm to converge quickly to a local minimum, but explore the design space less extensively. Whereas if $r$ is large, then at each generation we save more designs, which will cause the algorithm to converge more slowly and hence explore the space more thoroughly. Like most optimisation routines, the parameters in this model must be tuned appropriately. The choice of perturbation function must also be tuned to achieve the most efficient convergence to the optimal design.

**Algorithm 2.4** Importance Sampling Particle Filter propagation algorithm

**Input:** Number of particles, N; initial state, $x_0$; observed data for day, $y$.

1: *Initialise*: set the initial condition, $\boldsymbol{X}(0)$; generate times of the $y$ observed events, of observed event type, from a uniform random variable over interval $[0, 1]$. Sort these times and add them to the stack, $\psi$, set $e_n$ (the next event type on the stack), $t_n$ (the next time on the stack) and set $t = 0$.

2: Set the initial importance weight $w = \ln(y!)$

3: **while** $|\psi| > 0$ **do**

4: Calculate the rates of the original process at the current state $a_i(\boldsymbol{X}(t))$, $i = 1, \ldots, M$, given the current state and $\alpha_0 = \sum_{i=1}^{M} a_i$.

5: Check the consistency of the system given the next forced event $e_n$. If inconsistent, then force an event to fix in the interval $[t, t_n]$. Assuming this event is of type $\ell$, generate,
$s \sim \text{TruncExp}(a_\ell, 0, t_n - t)$.
Update weight by $w \leftarrow w - \ln(a_\ell) + a_l s + \ln\big(1 - \exp(-a_\ell(t_n - t))\big)$.
Push $t + s$ and event type $l$ onto the stack and update $e_n$ and $t_n$.

6: Calculate the rates of the modified process, $b_i(\boldsymbol{X}(t), e_n)$, $i = 1, \ldots, M$, which depend on the current state and the next forced event. Calculate total rate, $\beta_0 = \sum_{i=1}^{M} b_i$.

7: Propose a time to the next event $t' \sim \text{Exp}(\beta_0)$.

8: **if** $t + t' < t_n$ **then**

9: Choose an event index, $j \in \{1, \ldots, M\}$ with probability $P(j = i) = \frac{b_i}{b_0}$ and update $\boldsymbol{X}(t)$ by event $j$ and $t \leftarrow t + t'$.
Update weight by $w \leftarrow w + \ln\left(\frac{a_j}{b_j}\right) - (\alpha_0 - \beta_0)t'$.

10: **else**

11: Implement the next forced event at time $t_n$. Update $X(t)$ by event $e_n$, $t \leftarrow t_n$ and
update weight by $w \leftarrow w + \ln(a_{e_n}) - (\alpha_0 - \beta_0)(t_n - t)$.

12: **end if**

13: **end while**

14: $w = \exp(w)$

**Output:** Weight of the particle over this day, $w$.

**Algorithm 2.5** INSH Algorithm

**Input:** An initial set of designs $D$ (usually a coarse grid of design points in the design space). Number of generations $W$, pertubation function $f(d|d')$, an acceptance criteria, number of perturbations on accepted designs.

1: **for** $w = 1 : W$ **do**
2: **for** $d^i \in D$ **do**
3: Sample parameters $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$, and simulate $\boldsymbol{y}^i$ from the model.
4: Evaluate utility $u(d^i)$.
5: **end for**
6: Set $D'$ to be the designs that satisfy the acceptance criteria, and the current optimal design $d^*$.
7: Sample $m$ designs from $f(d|d')$, for each $d' \in D'$. Set $D$ to be the newly sampled designs.
8: **end for**

**Output:** Set of designs $d$, and corresponding approximate utilities $u(d)$ (and hence, the optimal design $d^* = \arg\max_{d \in D} u(d)$).

# Chapter 3

# Optimising Vaccine Trial for a Novel Fatal Pathogen

## 3.1 Introduction

Vaccines are very powerful and can be used to control or even eradicate a disease. However, when a new vaccine is proposed it must be tested, in a vaccine trial. Typically, after development of the vaccine, these trials are performed in three stages in strictly controlled environments. A brief summary of these stages has been outlined below:

- Phase 1 - A small group of trial participants receive the trial vaccine.
- Phase 2 - The clinical study is expanded and the vaccine is given to people with characteristics that are similar to those of whom the vaccine is intended.
- Phase 3 - The vaccine is given to thousands of people and is further tested for safety and efficacy [7].

The first two phases are predominantly for testing the safety of the vaccine and the final phase is for learning about the efficacy. However, in an emer-

gency, this method is very time-consuming. In the 2014-16 Ebola epidemic, the World Health Organisation (WHO) declared a world-wide emergency and declared that Phase 3 of the vaccine trials were to be performed on the general population in an at-risk community [15].

In this chapter, we will take this concept, simplify and consider designs that maximise certain objectives. In particular, we want to learn about the efficacy of the vaccine under trial, while simultaneously trying to minimise the number of deaths in the population. To achieve this, we require an epidemic model that tracks the deaths in the population as well as the other compartments related to the transmission. We modify a standard SIR epidemic model described in Section 2.2.2, to contain a death compartment, where infected individuals die with a certain probability, or recover with the complementary probability.

### 3.1.1 SIRD Model

The state transitions and the associated rates for the SIRD model are given in Table 3.1. Here, $\beta$ is the effective transmission rate, $\frac{1}{\gamma}$ is the average infectious period, and $p$ is the probability of an infected individual dying. We seek to test how vaccines affect the disease, so we need to separately track the individuals that have been vaccinated from those in the control group. This gives rise to the need for a model with parallel 'arms', one for each group.

**Assumptions** In order to construct this model, we make several assumptions. Firstly, we assume that the vaccine only affects the susceptibility of the vaccinated population (by the multiplicative parameter, $q$). Secondly, we assume that vaccinated and non-vaccinated individuals who are infectious have the same infectivity, $\beta$, and that they remain infectious for the same (average) length of time, $\frac{1}{\gamma}$. Finally, we assume that the probability

of death, $p$ is the same for both vaccinated and non-vaccinated individuals. All of these assumptions do not necessarily reflect the biology of vaccination accurately, as along with susceptibility, vaccines also the affect length of infection, transmissibility, severity of symptoms and probability of death. These assumptions are made to simplify the model and allow focus on the affect on susceptibility so that we can optimise the vaccine trials based on the parameter $q$.

Table 3.1: The events, state transitions, and rates in an SIRD model, where $\beta$ is the effective transmission rate, $\frac{1}{\gamma}$ is the average infectious period, and $p$ is the probability of an infected individual dying.

| Event | Transition | Rate |
|---|---|---|
| Infection | $(S, I, R, D) \to (S-1, I+1, R, D)$ | $\frac{\beta IS}{N-1-D}$ |
| Recovery | $(S, I, R, D) \to (S, I-1, R+1, D)$ | $(1-p)\gamma I$ |
| Death | $(S, I, R, D) \to (S, I-1, R, D+1)$ | $p\gamma I$ |

### 3.1.2 SIRD Model with Two Arms

The SIRD model discussed in Section 3.1.1 can be extended to have two arms, representing the vaccinated and unvaccinated sub-populations. Each arm consists of the SIRD compartments in parallel. The purpose of this structure is to record the movement of individuals from the unvaccinated to the vaccinated arm, via vaccination. All the individuals in the trial are from the same population, however we separate them into two sub-populations by placing them in one of the two parallel arms. These arms have their own separate compartments allowing us to keep track of the number of infections and deaths in each arm. Since the subjects are in the same population, there is however cross-infection between the two arms as there is a potential force of infection from both infectious compartments. See Figure 3.1 which

Table 3.2: The state transitions in an SIRD model with two arms, where $\beta$ is the effective transmission rate, $\frac{1}{\gamma}$ is the average infectious period, $p$ is the probability of an infected individual dying, and $q$ is the effect of the vaccine.

| Event | Transition | Rate |
|---|---|---|
| Infection of control | $(S_c, I_c) \to (S_c - 1, I_c + 1)$ | $\frac{\beta(I_c+I_v)S_c}{N-1}$ |
| Recovery of control | $(I_c, R_c) \to (I_c - 1, R_c + 1)$ | $(1-p)\gamma I_c$ |
| Death of control | $(I_c, D_c) \to (I_c - 1, D_c + 1)$ | $p\gamma I_c$ |
| Infection of vaccinated | $(S_v, I_v) \to (S_v - 1, I_v + 1)$ | $\frac{q\beta(I_c+I_v)S_v}{N-1}$ |
| Recovery of vaccianted | $(I_v, R_v) \to (I_v - 1, R_v + 1)$ | $(1-p)\gamma I_v$ |
| Death of vaccinated | $(I_v, D_v) \to (I_v - 1, D_v + 1)$ | $p\gamma I_v$ |

is a depiction of the transition diagram for this model, and Table 3.2 for the transition rates. Note that for simplicity, we only explicitly write the states that change for each transition. The compartments with a subscript $c$ contain individuals in the control arm (unvaccinated), and the subscript $v$ corresponds to individuals that have been vaccinated.

For simplicity, we have chosen the vaccine effect to be a multiplicative effect on the susceptibility to infection of the vaccinated individuals. That is, the vaccine only changes the rate at which vaccinated individuals become infected, and there are no further effects once the individual becomes infected. This effect is labelled $q$ in the vaccinated arm of the model. So, if $q$ is less than 1, the vaccine decreases the rate of a vaccinated susceptible individual becoming infected. If $q$ is greater than 1, the vaccine has an adverse effect by increasing this rate. The dashed line (transition) between the two Susceptible compartments corresponds to the vaccination scheme that is implemented, where decisions are based on the current knowledge about the parameter $q$. This decision about vaccination is the focus of our experimental design,

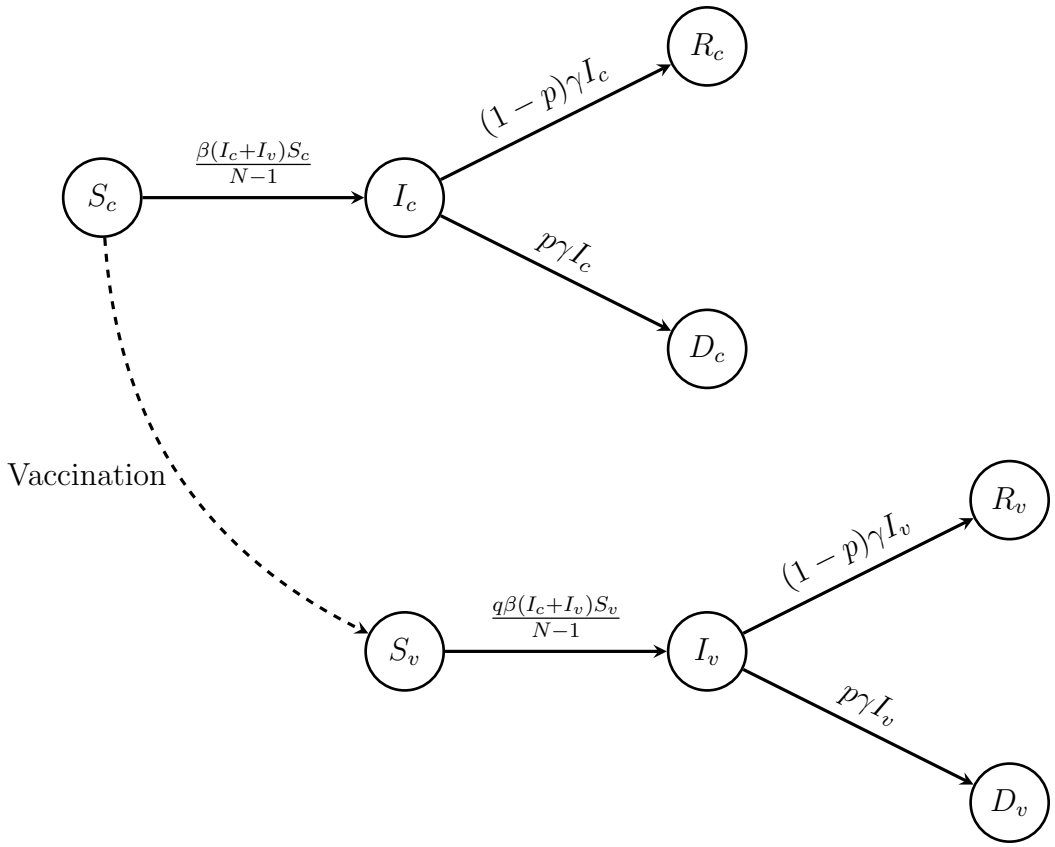


Figure 3.1: Transition diagram for the SIRD model with two arms, where $\beta$ is the effective transmission rate, $\frac{1}{\gamma}$ is the average infectious period, $p$ is the probability of an infected individual dying, and $q$ is the effect of the vaccine on effective transmission.

which we seek to optimise by learning about the vaccine and decreasing the number of deaths. The design elements we consider are discussed in Section 3.1.4.

### 3.1.3 Likelihood Estimation

We are interested in comparing adaptive designs, where design decisions are made at the end of each day and are based on the posterior distribution for the vaccination efficacy, $q$. At the end of a given day, the posterior distribution for $q$ is calculated, and the resulting actions are made at the same time. This requires the calculation of the posterior distribution based on the data collected, at the end of each day.

The focus of this chapter is to apply optimal experimental design to a scenario which is simplified compared to reality. We assume that we have full knowledge about the transmission dynamics of the disease and hence we only need to estimate the vaccine efficacy, $q$, resulting in more efficient computation. We further assume that every transition and the times of each transition are observed. This assumption allows us to calculate the likelihood exactly and efficiently. Relaxing this assumption will drastically increase the computational complexity and is considered in Chapters 4 and 5.

Let $\{X(t)\}_{t\geq 0}$ be the continuous-time Markov chain that describes the evolution of the disease. Let the random variable $T_i$ be the epoch of the $i^{\text{th}}$ transition in the Markov chain, and let the random variable $\boldsymbol{S}_i$ be the state after this transition. For simplicity, we set $T_0 = 0$, and let $\boldsymbol{S_0}$ be the initial state of the Markov chain. This model has 8 individual compartments and so the state of the Markov chain is the vector describing the number of individuals in each compartment, $\boldsymbol{s}_i = (S_c, I_c, R_c, D_c, S_v, I_v, R_v, D_v)$. Note we can remove one compartment from the model since the population is a fixed size, $N$.

The observed data is the set of tuples,

$$\boldsymbol{y}_n = \Big\{(t_i, \boldsymbol{s}_i)\Big| i = 1, \ldots, n\Big\},$$

up to and including the $n^{\text{th}}$ transition.

Recall the unnormalised posterior distribution is calculated as follows,

$$f(q|\boldsymbol{y}) \propto f(q) f(\boldsymbol{y}|q),$$

where $f(q)$ is the prior distribution for $q$, which is discussed briefly in Section 3.1.5, and $f(\boldsymbol{y}|q)$ is the likelihood of the data $\boldsymbol{y}$, given the parameter $q$. In order to calculate the likelihood, we utilise the fact that Markov chains have exponential holding times and that the transition rates of the compartmental models are determined by the number of individuals in each compartment, i.e., the state of the process.

Assume at some arbitrary time $t$, that the $n$ events represented by $\boldsymbol{y}_n = \Big\{(t_i, \boldsymbol{s}_i)\Big| i = 1, \ldots, n\Big\}$ have occurred. Then the likelihood can be expressed as,

$$f(\boldsymbol{y}_n|q) = \prod_{i=1}^{n} f\Big((T_i, \boldsymbol{S}_i) = (t_i, \boldsymbol{s}_i) \;\Big|\; (T_{i-1}, \boldsymbol{S}_{i-1}) = (t_{i-1}, \boldsymbol{s}_{i-1})\Big) P\Big((T_0, \boldsymbol{S}_0) = (0, \boldsymbol{s}_0)\Big),$$

where $f\big((T_i, \boldsymbol{S}_i)|(T_{i-1}, \boldsymbol{S}_{i-1})\big)$ is the conditional density of the transition into $\boldsymbol{s}_i$ occurring at time $t_i$, given the process transitioned into the state $\boldsymbol{s}_{i-1}$ at time $t_{i-1}$. We have,

$$
\begin{aligned}
&f\Big((T_i, \boldsymbol{S}_i) = (t_i, \boldsymbol{s}_i) \;\Big|\; (T_{i-1}, \boldsymbol{S}_{i-1}) = (t_{i-1}, \boldsymbol{s}_{i-1})\Big) \\
&\quad = \lim_{h\to 0^+} \frac{P\Big(X(t_i+h) = \boldsymbol{s}_i,\ \{X(u) = \boldsymbol{s}_{i-1} :\ t_{i-1} \le u < t_i\}\Big| X(t_{i-1}) = \boldsymbol{s}_{i-1}\Big)}{h} \\
&\quad = \lim_{h\to 0^+} \frac{P\Big(X(t_i+h) = \boldsymbol{s}_i\Big| X(t_{i-1}) = \boldsymbol{s}_{i-1}, \{X(u) = \boldsymbol{s}_{i-1} :\ t_{i-1} \le u < t_i\}\Big)}{h} \\
&\qquad\qquad \times P\Big(\{X(u) = \boldsymbol{s}_{i-1} :\ t_{i-1} \le u < t_i\}\Big| X(t_{i-1}) = \boldsymbol{s}_{i-1}\Big) \\
&\quad = \lim_{h\to 0^+} \frac{P\Big(X(t_i+h) = \boldsymbol{s}_i\Big| X(t_i^-) = \boldsymbol{s}_{i-1}\Big)}{h} \\
&\qquad\qquad \times P\Big(\{X(u) = \boldsymbol{s}_{i-1} :\ t_{i-1} < u < t_i\}\Big| X(t_{i-1}) = \boldsymbol{s}_{i-1}\Big),
\end{aligned}
$$

by the Markov property, and where $X(t_i^-) = \lim_{\tau \uparrow t_i} X(\tau)$.

From Section 2.1, the first term can be expressed as,

$$
\lim_{h\to 0^+} \frac{P\Big(X(t_i+h) = \boldsymbol{s}_i\Big| X(t_i^-) = \boldsymbol{s}_{i-1}\Big)}{h} = \lim_{h\to 0^+} \frac{q_{\boldsymbol{s}_{i-1},\boldsymbol{s}_i} h + o(h)}{h}.
$$

Similarly, the second term can be simplified to,

$$
P\Big(\{X(u) = \boldsymbol{s}_{i-1} :\ t_{i-1} < u < t_i\}\Big| X(t_{i-1}) = \boldsymbol{s}_{i-1}\Big) = e^{-q_{\boldsymbol{s}_{i-1}}(t_i - t_{i-1})}.
$$

Therefore, the likelihood component is given by

$$
\begin{aligned}
&f\Big((T_i, \boldsymbol{S}_i) = (t_i, \boldsymbol{s}_i) \;\Big|\; (T_{i-1}, \boldsymbol{S}_{i-1}) = (t_{i-1}, \boldsymbol{s}_{i-1})\Big) \\
&\qquad = q_{\boldsymbol{s}_{i-1},\boldsymbol{s}_i} e^{-q_{\boldsymbol{s}_{i-1}}(t_i - t_{i-1})}.
\end{aligned}
$$

Recall from Section 2.1, that $q_{\boldsymbol{s}_{i-1},j}$ is the transition rate from state $\boldsymbol{s}_{i-1}$ to state $j$, and $q_{\boldsymbol{s}_{i-1}} = -\sum_{j \neq \boldsymbol{s}_{i-1}} q_{\boldsymbol{s}_{i-1},j}$.

In the above calculation, we have ignored that at the end of every day, we make decisions about vaccinating individuals, dictated by the trial design. This is important as once these individuals have been vaccinated, there is a resulting change in the transition rates for subsequent events. This changes

our likelihood calculation when events span over a day. If this happens we need to multiply the probability that no more transitions happen until the end of the previous day, by the probability that the next transition happens on the current day (using modified transmission rates).

Recall the $i^{\text{th}}$ event occurs at time $t_i$, and the previous event occurs at time $t_{i-1}$. Now assume the end of the day is at time $t'$ such that $t_{i-1} < t' < t_i$ and that a vaccination occurs at time $t'$, moving the chain to state $\ell$. Note that in this scenario, it is not possible to transition from state $\boldsymbol{s}_{i-1}$ to state $\boldsymbol{s}_i$ in one transition, and that the movement from state $\boldsymbol{s}_{i-1}$ to state $\ell$ is the result of the vaccination regime. This changes the density slightly from the previous result to be

$$
\begin{aligned}
&f\Big((T_i, \boldsymbol{S}_i) = (t_i, \boldsymbol{s}_i) \;\Big|\; (T_{i-1}, \boldsymbol{S}_{i-1}) = (t_{i-1}, \boldsymbol{s}_{i-1})\Big) \\
&\quad = \lim_{h \to 0^+} \frac{P\Big(X(t_i + h) = \boldsymbol{s}_i,\ \{X(u) = \ell :\ t' \le u < t_i\} \Big| X(t') = \ell\Big)}{h} \\
&\qquad\qquad \times P\Big(X(t') = \boldsymbol{s}_{i-1},\ \{X(u) = \boldsymbol{s}_{i-1} :\ t_{i-1} \le u < t'\} \Big| X(t_{i-1}) = \boldsymbol{s}_{i-1}\Big) \\
&\quad = q_{\ell, \boldsymbol{s}_i} e^{-q_\ell (t_i - t')} \times e^{-q_{\boldsymbol{s}_{i-1}} (t' - t_{i-1})} \\
&\quad = q_{\ell, \boldsymbol{s}_i} e^{-q_\ell (t_i - t') - q_{\boldsymbol{s}_{i-1}} (t' - t_{i-1})}.
\end{aligned}
$$

### 3.1.4 Design Parameters

As mentioned in Section 3.1.2 we are interested in the optimal design that both learns about the vaccine efficacy while minimising the number of deaths over the whole outbreak. We consider designs that adapt to the learned knowledge about vaccine efficacy. To put it simply, if we believe the efficacy is less than 1, we should vaccinate many people and if we believe it is greater than 1, we should not vaccinate; but we must ascertain such a belief with some level of certainty.

We first need a rule for deciding if a vaccine is beneficial. This rule would

take the posterior distribution for vaccine efficacy ($q$) and return a measure of our confidence that the true $q$-value is less than 1. We chose to consider calculating the mass under the posterior distribution less than a cutoff, $c$. That is to say, the vaccine is deemed beneficial if $P(q \leq c \mid \boldsymbol{y}) > \alpha$; where $c$ is a cutoff point for the vaccine efficacy and $\alpha$ is considered a threshold probability. Thus, at the end of each day, we calculate the posterior of $q$, then calculate the probability that $q$ is less than $c$ and compare this probability to $\alpha$ to decide to how many people to vaccinate.

In this model, there are 5 key design parameters:

- $v$ - the initial number of individuals vaccinated at the onset of the trial;
- $v_1$ - the number of individuals vaccinated at the end of a day, given we believe the vaccine is beneficial, i.e., if $P(q \leq c \mid \boldsymbol{y}) \geq \alpha$;
- $c$ - the threshold that we require $\alpha$ of the mass in the posterior distribution for $q$, to be less than, to deem the vaccine as beneficial; and,
- $\alpha$ - the amount of mass in the posterior distribution we require less than a threshold $c$ to deem the vaccine beneficial.

We represent a design as the tuple, $(v, v_1, c, \alpha)$.

### 3.1.5 Prior Knowledge

In Bayesian inference, we require the prior distribution of the parameters we are estimating in order to calculate the posterior distribution. However, if prior knowledge is very limited, one can use an 'uninformative' prior. In this problem, we seek to estimate the vaccine efficacy $q$, which we have artificially restricted to be between 0 and 2. As such, we use prior distributions that are restricted to a domain of $[0, 2]$. In this case, a simple uninformative prior distribution is a uniform distribution, $U(0, 2)$.

## 3.2 Optimal Design Criteria (Utility)

To define optimal design, we require an experimental design criterion, often referred to as a utility function. That is, a metric we can calculate for each design and use to compare designs to decide which is optimal.

### 3.2.1 Expected Number of Deaths

First, we are interested in investigating how the vaccine efficacy, $q$, affects the total number of deaths over the outbreak. To do this, we ran a set of simulations with two extreme designs, $d_1 = (N, 0, c, \alpha)$ and $d_2 = (0, 0, c, \alpha)$. The design $d_1$ has all individuals vaccinated at the onset, and design $d_2$ has no individuals vaccinated at any point in the epidemic, and is hence equivalent to a standard SIRD outbreak. Figure 3.2 illustrates the expected number of deaths against different values of $q$, under these two designs. The key result we can draw from Figure 3.2 is that the relationship between the expected number of deaths and $q$ is monotonic, as expected, but the values themselves are not necessarily intuitive for design $d_2$. Since design $d_1$ has no individuals vaccinated during the trial, there is no dependence on vaccine efficacy, $q$; hence the horizontal line.

### 3.2.2 Oracle Design

We define the 'oracle' design to be the optimal design to reduce the expected number of deaths, given that the vaccine efficacy, $q$, is known exactly. If we know that $q \leq 1$, then the vaccine is beneficial, and the optimal design would be to vaccinate all individuals at the onset. Similarly, if we know that $q > 1$, then the vaccine is detrimental and the optimal design would be to never vaccinate. As such, the oracle design corresponds to the orange curve for $q \leq 1$ and the blue curve for $q > 1$, in Figure 3.2.

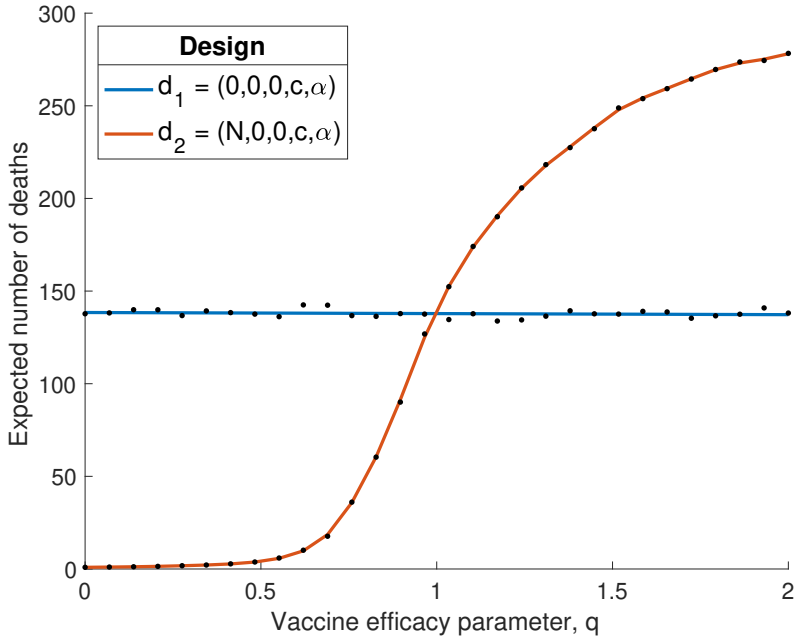


Figure 3.2: The expected number of deaths in an SIRD model against vaccine efficacy, $q$, under the two extreme designs, $d_1$ and $d_2$ where $N = 1000$. The model parameters are, $\beta = 0.3$, $\frac{1}{\gamma} = 5$ and $p = 0.3$. A smoothing spline has been fit across the 30 equispaced points over the domain for each design ($d_1$ in blue and $d_2$ in orange).

Since the oracle design is acting according to exact knowledge of the efficacy, it is the ideal outcome (with respect to minimising the expected number of deaths). Since we are estimating the vaccine efficacy as the outbreak progresses, the best design is one that most closely matches the oracle design. The measure of this difference between the oracle design and a given design under investigation, $d$, is given by

$$u(d) = \int_0^\infty \Big(D_d(q) - D_O(q)\Big) f(q) dq, \tag{3.1}$$

where $D_d(q)$ is the expected number of deaths under design $d$ for vaccine

efficacy, q; $D_O(q)$ is the expected number of deaths under the oracle design for vaccine efficacy, $q$; and $f(q)$ is the prior density for $q$. Note the quantity in the integral will always be positive as no design will minimise deaths as well as the oracle. The quantity $u(d)$ can be thought of as the expected additional number of deaths under design $d$, compared to the oracle design. This is defined to be the utility function. this novel utility function is easily interpretable by epidemiologists, unlike some other utility functions such as the Kullback-Liebler divergence. The optimal design is then defined to be,

$$d^* = \arg\min_d \ u(d).$$

In this model, we are considering Bayesian optimal experimental design. Hence, we make decisions based on the posterior distribution which requires the use of a prior distribution. We also need to sample values of the parameters for designs to test, which we call this the sampling distribution. The support of the sampling distribution must be contained in the support of the prior distribution and typically they are chosen to be the same.

The prior distribution appears in two places: the calculation of the posterior distribution, and the evaluation of the utility function. Note that in the utility function (Equation 3.1, we integrate the difference in the expected number of deaths with respect to the prior distribution This is to ensure we find the design that matches the oracle design most closely where there is a high proportion of mass in the sampling distribution.

The choice of the prior affects the calculation of the posterior distribution which in turn dictates how we vaccinate individuals. The posterior distribution combines the information from the prior distribution and the data; if we have little data, the prior has a significant effect on the posterior distribution. If we have many data points, the likelihood will dominate the prior. As such, the choice of prior distribution is significant in the early stages of an outbreak; however, as the outbreak progresses, the prior information tends

to be overpowered. Thus over an entire outbreak, the choice of prior distribution is unlikely to have a significant effect on the outcome. However, as the prior is used in the evaluation of the utility function, it does have a significant effect on the utility for each design.

We will use a uniform prior over the domain of the parameter $q$.

## 3.3 Results

We implemented the INSH algorithm [25], as discussed in Section 2.5.1 and outlined in Algorithm 2.5, for 10 generations with 100 initial designs. At the end of each generation we kept the 20 designs that performed best (including the current overall optimal design) and perturbed each of these designs 5 times. The perturbation kernel was a mixture of discrete-truncated uniform distributions (for design parameters $v$ and $v_1$) and normal distributions (for design parameters $c$ and $\alpha$):

$$\begin{aligned} v^{(w)} &= v^{(w+)} + U_d(-50, 50); \\ v_1^{(w)} &= v_1^{(w+)} + U_d(-30, 30); \\ c^{(w)} &= c^{(w+)} + N(0, 0.1); \\ \alpha^{(w)} &= \alpha^{(w+)} + N(0, 0.07). \end{aligned}$$

Note that when perturbing the set of designs, we take care to ensure that the sampled values remained within the limits outlined in Table 3.3. After 10 generations the best design was found to be, $d^* = (72, 250, 1.24, 0.75)$, with an expected additional number of deaths, $u(d^*) = 18.70$, that is to say that our design we expect 18.7 additional deaths more than the oracle design. In context, this design means that *72* individuals are initially vaccinated, Then if the estimated posterior distribution has at least 0.75 of the mass below 1.24 at the end of a particular day, 250 more people are vaccinated.

Table 3.3: Maximum and minimum values for the design parameters.

| Parameter | Min | Max |
|---|---|---|
| $v$ | 0 | $\frac{N}{2}$ |
| $v_1$ | 0 | $N - I_c$ |
| $c$ | 0 | 2 |
| $\alpha$ | 0 | 1 |

### 3.3.1 Assessing Convergence to Local Minimum

Now that we have implemented INSH, we will check that it did indeed converge to a local minimum. First we will check that it converged, then we will check that the convergence was to a local minimum. Figure 3.3 shows the algorithm converging to a set of designs. Each plot is a projection of the labelled design parameter onto one dimension. On the vertical-axis we are plotting the generations of the algorithm. The colour bar indicates the value of the utility function for each design, where blue is a poor design and red indicates better designs. It is important to note the range on the horizontal-axis is different for each plot.

In Figure 3.3, we can see that each parameter does converge to a narrower set of points. While this convergence happens at different rates for each parameter, all parameters seem to have converged completely by the $7^{\text{th}}$ generation.

The INSH algorithm converged to a set of best designs. It is difficult to know if this set contains the true optimal design, but we can assess whether it is indeed a local minimiser of the utility function. Figure 3.4 shows a set of plots assessing the convergence to a local minimum. In this figure for each plot we held the other design parameters at the optimal design and varied the parameter under investigation over a coarse grid on its support. We then evaluated the utility for these designs over the grid and plot the utilities

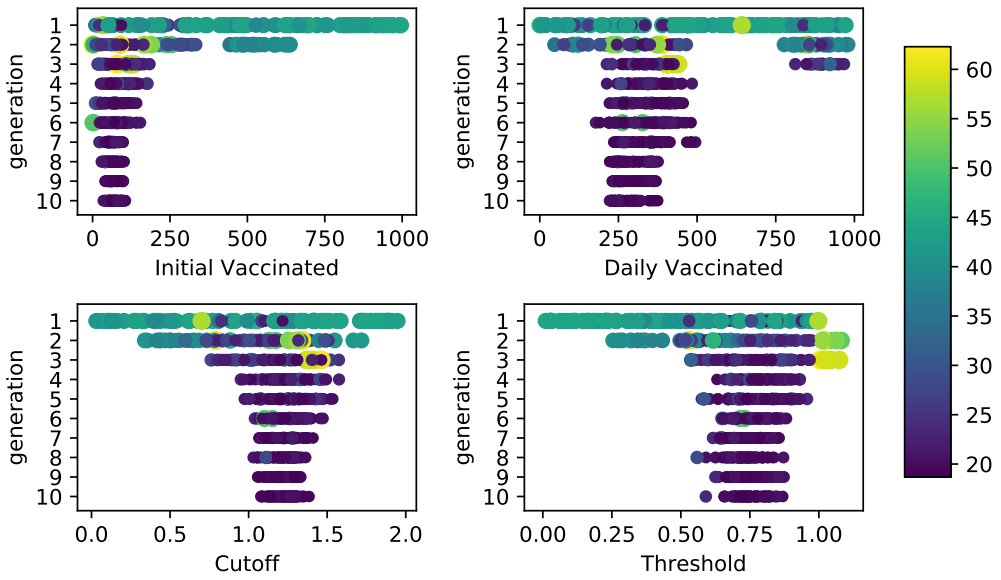


Figure 3.3: A plot that shows INSH converging to a set of designs, where the colourbar indicates the value of the utility function. The generation is plotted on the vertical-axis reading down the plot and the horizontal-axis is the value of that design parameter. Note that at each generation the 100 points that were considered by INSH are plotted and we see them converge to a set of points as the algorithm evolves. Again, the model parameters are, $\beta = 0.3$, $\frac{1}{\gamma} = 5$ and $p = 0.3$.

against that dimension of the design space.

In Figure 3.4 we can see that with regard to all design parameters, except the number of daily vaccinations, that we are indeed clearly at a local minimum. It is interesting to note that the plot for the daily vaccinated parameter shows that the utility is close to invariant to this parameter given the value of the other parameters. We can interpret this to mean that the optimal design waits until enough is known about the vaccine (given by $c$ and $\alpha$) and then vaccinate every individual as soon as possible. By *stopping time*, this means that if the amount of mass under the posterior less than the cutoff is greater than the threshold, then vaccinate every individual imme-

diately and end the trial. We can make this connection because even when we vaccinate 100 individuals each day, then in approximately 9 days (under this design) every individual would be vaccinated, thus effectively a stopping time.

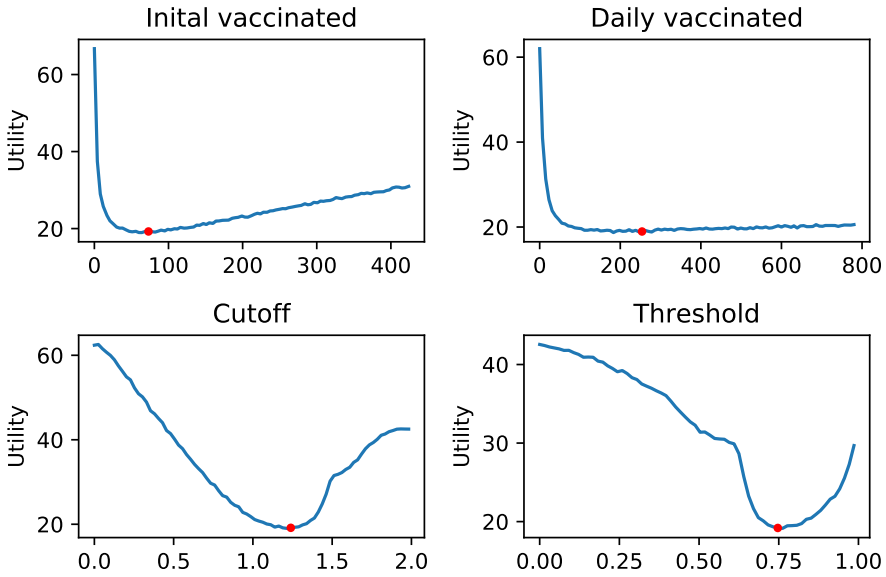


Figure 3.4: In each subplot we kept three design options held at the value from the current best design and changed the named parameter over the full range of possible vaccine values. The INSH-best design parameter value has been included as a red point in each subplot.

After considering these plots to assess convergence, the algorithm seems to have adequately explored the design space and found a local minimum. Figure 3.3 demonstrates that the algorithm has explored the design space well and converged efficiently to a set of optimal designs. Figure 3.4 shows that the optimal design is indeed a local minimum of the design space.

## 3.4 Getting the Most Out of the Super-Computer

A significant component of this work was utilising the full potential of the super-computer we had access to. Since trialling a design takes so many calculations, we required parallel computing and at times had to approach this in an unusual way. However to discuss parallelising the computation on the super-computer, we must first discuss what goes into investigating a design fully.

For a given design under investigation, we needed to test how that design performed under a range of potential $q$-values and then calculate the utility. As far as testing how a design performs, we would fix a value of $q$ and simulate the outbreak under that design. While the simulation was happening we are estimating posterior distributions and vaccinating accordingly. We then picked a new $q$ value and simulated an outbreak under that design again and so on. We chose to test the design for 30 different $q$-values between 0 and 2 (equispaced), we chose 30 because each node on the super-computer has 32 cores. This is reasonably fast, however to achieve a reasonable estimate of the utility, we had to repeat the 30 simulations under the different values of $q$ many times. Rather than setting a number of iterations, we instead ran the code for 6 hours and took how many were achieved in that time (this resulted in around 25,000 repetitions). This was important because if we requested a certain amount of time for the code to run and it finished earlier, then the job could be allocated lower priority on the queue.

The optimal experimental design is simply many nested loops of simulating an outbreak, the challenge here is finding which loop to parallelise. For instance we could have each core trial a design over the 30 values of $q$ or could have 30 cores perform one value of $q$ each. We chose to allocate a different value of $q$ to each core to reduce the number of files needed to save information, as writing many files can slow down the processes. We

noticed however that simulations when $q$ was near 0, ended very early (effective vaccine) while simulations when $q$ was nearer 2 took significantly longer. This was an issue because all 30 cores must complete their operations before they can move on to the next task resulting in idle cores for periods of time. If we order the values of $q$ under investigation as $\{q_1, q_2, \ldots, q_{30}\}$, then we noticed that time taken for $q_1$ and $q_{30}$ to complete was approximately the same as the time required for $q_2$ and $q_{29}$. We noticed this pattern continued for nearly all values. As such we tried allocating $q_1$ and $q_{30}$ to the same core and so on. This did increase our core efficiency, but we experimented further and found that allocating 6 values of $q$ to each node resulted in the highest efficiency. To use all 30 cores available we would then allocate 6 values of $q$ to each core resulting in 5 full evaluations once all cores had completed. For instance core 1 was allocated $\{q_1, q_6, q_{11}, q_{16}, q_{21}, q_{26}\}$, core 2 was allocated $\{q_2, q_7, q_{12}, q_{17}, q_{22}, q_{27}\}$ and so on. This was advantageous for two main reasons, first was the increased efficiency but the second was that we were able reduce the number of output operations by a fifth.

Another step in the optimal experimental design that requires consideration is the implementation of INSH. In each generation of INSH we trialled 100 designs and then needed to collate the information from these trials and then trial another 100 designs. As the super-computer is a competitive resource, it can take a lot of time getting to the top of the queue. So we set dependencies on the jobs we submitted so that once one was completed the others remained on top of the queue.

## 3.5 Limitations of the Model

In this chapter we made modelling assumptions that were very naive. First, for the sake of computation simplicity and efficiency, we assumed the data was fully observed. This assumption allowed us to calculate the likelihood

function exactly (using basic Markov chain theory), which is precise, and efficient. However, this assumption is not realistic since in a real outbreak, that kind of data is impossible to collect and hence represents a poor modelling assumption.

Another crucial modelling assumption we made was to assume that the vaccine effect can range between 0 and 2. We made this assumption to provide a simple illustrative example in which the adaptive design would learn about the vaccine and make decisions accordingly. If we had assumed the vaccine affect was between 0 and 1, then the optimal design would be to vaccinate everyone initially as that can be no worse than do nothing. Our assumption allowed the experimental design to converge to an optimal design and ensured we were able to distinguish between different designs. However, this is an unrealistic assumption as a vaccine that could have detrimental affects would not be admitted to the general population, as it would fail to pass the basic clinical trials. Although this may seem impractical it still achieves the intended purpose of illustrating the methods discussed.

The third modelling assumption we made was to choose a simple model (SIRD), which has few compartments and does not have exposed periods, or symptomatic and asymptomatic infectious individuals. We chose a simple model because this once again decreased computation time and allowed us to explore the optimal experimental design, without having to work through slow computing. In relation to the choice of epidemiological model, we also chose parameters for the model arbitrarily until we had a disease that spread well but not too quickly such that the design made a difference.

We address and relax all of these assumptions in Chapter 4.

# Chapter 4

# Computational Efficiency of Likelihood Estimation

## 4.1 Introduction

The work in Chapter 3 familiarised us with the context of optimal experimental design in vaccine trials and developed methods that we can build on. The assumptions made to enable computational efficiency in the preceding example are not realistic in practical situations and so, in the following chapters, we will be addressing these assumptions. We will be addressing the three main issues highlighted in Section 3.5 in this chapter.

**Fully observed data.** In the previous chapter it was assumed that we can fully observe the data from the outbreak. We made this decision for the sake of computational efficiency and simplicity. However, it resulted in an unrealistic problem as fully observing the data is almost impossible. In this chapter we will remove this modelling assumption by instead using partially-observed data; that is, we only observe certain types of transitions and we do not know the exact time of these transitions. We have assumed that we observe the data to daily precision, meaning at the end of each day we have

a count of the number of observed events that day. We will assume that we can only observe the transition from an individual being asymptomatic to showing symptoms (that is, $E \to I$). This might seem unrealistic, however as seen with COVID-19, once a positive test comes back the individual being tested is asked when they first showed symptoms (to a daily precision). It is true that the result from the test might be delayed by some number of days, and hence the patients response will be delayed, but for the sake of simplicity, we are assuming there is no delay in the observations. Further, some individuals become symptomatic and are never tested or reported, but for simplicity we do not account for these cases. We will go into more detail about the model we chose to use and the partially-observed data in Section 4.2.1.

**Unrestricted vaccine effect.** In Chapter 3, we made the assumption that the vaccine effect could range between 0 and 2. Since the effect was multiplicative on the infection rate, this means the vaccine could be 100% effective all the way to increasing your chance of infection by a factor of 2. This is very unrealistic as a vaccine that increases the rate of infection would never be administered to the population. In reality, before a vaccine is administered to the population, it would have to undergo three-stage clinical trials to assess the safety of the vaccine in a controlled environment. However we needed to make this assumption to ensure there is a trade-off between vaccinating every individual initially and waiting to learn about the parameters. In a global pandemic, such as the COVID-19 outbreak, these trials might be expedited [4] to achieve a swift and efficient roll-out of the vaccine. In this chapter, we assume that the vaccine effect is between 0 and 1.

**Simplistic model.** In Chapter 3 we chose to use a simple epidemiological model (the SIRD model, from Section 3.1.1). Not only was this model unsophisticated in regards to an actual virus, we also chose model parameters

arbitrarily to ensure the outbreak spread slowly, yet had a large enough first wave to allow the design to have an effect. This was acceptable because we were only interested in building a framework for optimal experimental design and learning about the process. However we are now interested in finding a practical method and hence solution to a more practical and realistic problem. Hence, in this chapter we will use a model that has been used to model COVID-19, with parameters from real data based upon [22]. We will go into more detail about this in Section 4.2.1.

## 4.2 Three-Arm Model with Partially-Observed Data

As discussed above, the model used here now assumes we can observe only one transition. Further, the previous previous model to be too simplistic and unrealistic, so we have taken inspiration from an early model for the COVID-19 virus from [22]. In this paper, they use an SEIIRH model, where the first infectious compartment is asymptomatic and the second is symptomatic.

**Assumptions** The SEIIRH model has similar assumptions to the SIRD model for simplicity. We continue to assume that all participants (regardless of vaccination status) have the same infectivity, $\beta$, and the same average length of infection (asymptomatic and symptomatic), $1/\gamma_a$ and $1/\gamma_s$. We also assume that all participants have the same probability of death, $p$. As with the SIRD model, these assumptions do not reflect the biology of vaccinations, but are made for simplicity and to focus on optimising in regards to the vaccines' efficacies in reducing transmissibility, $q_1$ and $q_2$.

### 4.2.1 The SEIIRH Model

In order to construct a more realistic model and take into account the relaxing of assumptions described above, we have taken inspiration from an early model for the COVID-19 virus from [22]. In this paper, they use an SEIIRH model, where the first infectious compartment is asymptomatic and the second is symptomatic. The structure of the SEIIRH model is very similar to the SIRD model in Section 3.1.1, however there is more compartments. Note that we have included a hospitalised compartment instead of deaths for this model, because [22] provided estimated hospitalisation rates for COVID-19. We will go into more detail on the parameter values in Section 5.1.1. Note that in the SEIIRH model, we depict the asymptomatic infectious compartment as $I_a$ and the symptomatic infectious compartment as $I_s$. Note here, that we are assuming the infectious rate is the same between symptomatic and asymptomatic individuals, this could be modified easily to mimic self-isolation, that we have seen is effective with the Covid-19 pandemic. See Table 4.1 for the transition rates for this model. Here, $\beta$ is the effective transmission rate, $\frac{1}{\sigma}$ is the average time spent in the exposed compartment, $\frac{1}{\gamma_a}$ is the average time spent in the asymptomatic infectious compartment, $\frac{1}{\gamma_s}$ is the average time spent in the symptomatic infectious compartment and $p$ is the probability an infectious individual is hospitalised.

### 4.2.2 SEIIRH Model with Three Arms

Similar to Chapter 3, we will be vaccinating individuals in the population and hence will need to label them differently, hence the need for a multi-arm model. However for this chapter, we will be changing the base model to an SEIIRH model as this model is a more realistic choice for the COVID-19 virus. Similarly we have decided to change the model to be a three-arm model, where one arm is the control ($c$), one arm is the individuals whom had

Table 4.1: The state transitions of the SEIIRH model, where we have included only the relevant states in the transition column.

| Event | Transition | Rate |
|---|---|---|
| Exposure | $(S, E) \to (S-1, E+1)$ | $\frac{\beta(I_a+I_s)S}{N-1}$ |
| Infection (asymptomatic) | $(E, I_a) \to (E-1, I_a+1)$ | $\sigma E$ |
| Infection (symptomatic) | $(I_a, I_s) \to (I_a-1, I_s+1)$ | $\gamma_a I_a$ |
| Recovery | $(I_s, R) \to (I_s-1, R+1)$ | $(1-p)\gamma_s I_s$ |
| Hospitalisation | $(I_s, H) \to (I_s-1, H+1)$ | $p\gamma_s I_s$ |

vaccine 1 administered ($v_1$), and the last arm is the individuals whom had vaccine 2 administered ($v_2$). In this scenario we are trialing the two vaccines simultaneously, to learn about their effectiveness and to minimise the spread of the disease. This method could be applied to a scenario where there are two viable vaccines, but it is unclear which is more efficacious. Ideally the optimal design would be one that waits for some time to see which vaccine is better and will vaccinate most people with that vaccine. Note that we are assuming once an individual has received a vaccine, they are ineligible for the other vaccine.

In this model we have included two vaccine affects: $q_1$ corresponding to vaccine 1, and $q_2$ corresponding to vaccine 2. We have assumed that the vaccine changes the rate of infection and that it has a multiplicative effect on the infection rate. See Figure 4.1 for a diagram of the three-arm SEIIRH model. Note that the three arms are labelled $c$ for control, $v_1$ for vaccine 1, and $v_2$ for vaccine 2. Similarly note, the subscript on the infectious compartments $a$ and $s$ correspond to asymptomatic and symptomatic infectious. Similarly, we define $I_t$ to be the total number of infectious individuals in the population, both symptomatic and asymptomatic in all arms.

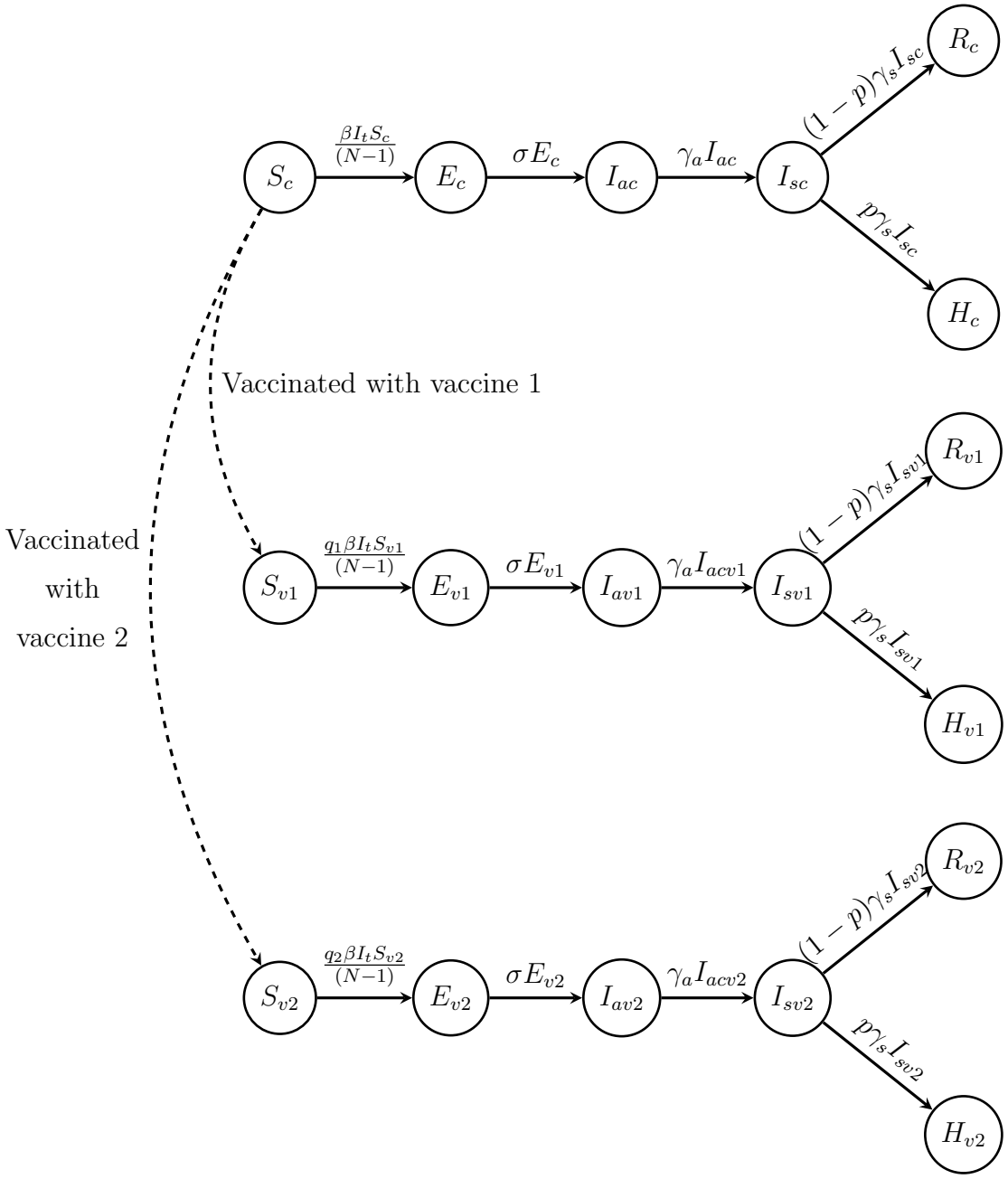


Figure 4.1: Transition diagram for the SEIIRH model with three arms, where $q_1$ is the effect of vaccine 1 and $q_2$ is the effect of vaccine 2. Note that $I_t$ is the total number of infected individuals in the population ($I_{ac} + I_{sc} + I_{av1} + I_{sv1} + I_{av2} + I_{sv2}$).

## 4.3 Required Calculations for Optimal Experimental Design

Before we can implement the three-arm SEIIRH model in an optimal experimental design framework, we must first find a method for calculating the likelihood function for the partially-observed data. The key requirement for this method is that it must be computationally efficient, as we need to perform many calculations to evaluate a design.

**Calculations required to evaluate the Posterior.** In Chapter 3, when evaluating the posterior distribution for the parameters, we found it was most efficient to calculate the unnormalised posterior over a range of parameter values and then normalise it. This was feasible as we were only estimating one parameter; in this model we are estimating both $q_1$ and $q_2$. We investigated using a Metropolis-Hastings algorithm to estimate this. However, we discovered we would need approximately 2500 samples to achieve a satisfactory exploration of the space. Since the computational effort to obtain each sample is very large, 2500 samples is much too large for a computationally efficient method. Instead, we chose to use a similar method to the previous chapter, where we will evaluate the unnormalised posterior over a grid of $q_1$ and $q_2$ values and then normalise by numerical integration.

Let $n$ be the precision on the grid of model parameters. For each pair $(q_1, q_2)$, we will need to evaluate the likelihood once; then we will need $n^2$ likelihood evaluations per posterior evaluation.

**Calculations required to implement a design.** To implement a design, we simulate an outcome (using the SSA outline in Section 2.1.1) and each week we collect data and evaluate the posterior up to that time. Based on that posterior we vaccinate the population. Hence for a pair of true values of $(q_1, q_2)$, we need to perform likelihood calculations for each

week the outbreak lasts. Let $\ell$ be the number of weeks and hence posterior calculations that must be performed.

**Calculations required to trial a design.** In order to fully trial a design for a pair of true vaccine effects, $(q_1, q_2)$, we need to implement the design many times and average the outcomes. Let $m$ be the number of times we implement a design, to then take the average. As $m$ is increased the precision in the estimate will increase.

**Calculations required for utility evaluation.** As discussed in Section 2.5, a utility used in optimal experimental design is an integral over the range of parameter values that are unknown. In the context of this problem, the unknown parameters are $q_1$ and $q_2$, so for a utility evaluation, we require a grid of these values to then integrate over. Let $N$ be the precision of this grid, then we must trial each design $N^2$ times. Note that for the evaluation in Chapter 5, we took $N = n = 20$, although $N$ and $n$ can different.

By putting all of these together we end up needing $n^2 \times \ell \times m \times N^2$ likelihood calculations per design we wish to trial. This number is very large and grows quickly as we strive for a more precise result. However this total is simply the number of likelihood evaluations required to evaluate a design. We then must integrate over the design space; for instance when we implement the INSH algorithm in Section 3.3, we evaluated 1000 designs.

This highlights our need for a computationally-efficient method of likelihood evaluation for this model, in the context of optimal experimental design. In the following sections we will discuss various likelihood methods and compare their computational speed and their accuracy.

## 4.4 Exact Likelihood Calculation

It is technically possible to perform exact likelihood calculations on partially-observed data, however for this problem we have an 18-compartment model

Markov chain. The partial data significantly increases the computational complexity as we need to infer all the other transitions that are not observed. This method is computationally intractable for a model with 18 compartments. Instead we must look for a method that estimates the likelihood rather than calculates it exactly.

## 4.5 Bootstrap Particle Filter

A common method for likelihood-based inference when it is difficult to calculate the likelihood exactly is to use a particle filter, which is advantageous when the process is fast and easy to simulate. For this model, we can efficiently simulate an outbreak through the SSA. However, the events we are observing are often very rare, and so for a bootstrap particle filter to be accurate we would need a very large number of particles, which decreases the efficiency significantly. Instead we can look to use an importance sampling particle filter, that utilises the principles of importance sampling to assist in sampling these rare events.

## 4.6 Importance Sampling Particle Filter

The importance sampling particle filter is a precise method for estimating the likelihood of partially-observed data; however as mentioned in Section 2.4.3 as the model complexity increases it becomes more difficult to implement. In Section 2.4.3, we demonstrated how one would theoretically implement an importance sampling particle filter for an SEIR model. This is relevant to our SEIIRH model as the SEIR model has a chain of infections $S \to E \to I$ and the SEIIRH has a chain, $S \to E \to I_a \to I_s$. Our model is even more complex due to having three arms with cross infection between them.

In this section we illustrate an importance sampling particle filter on a

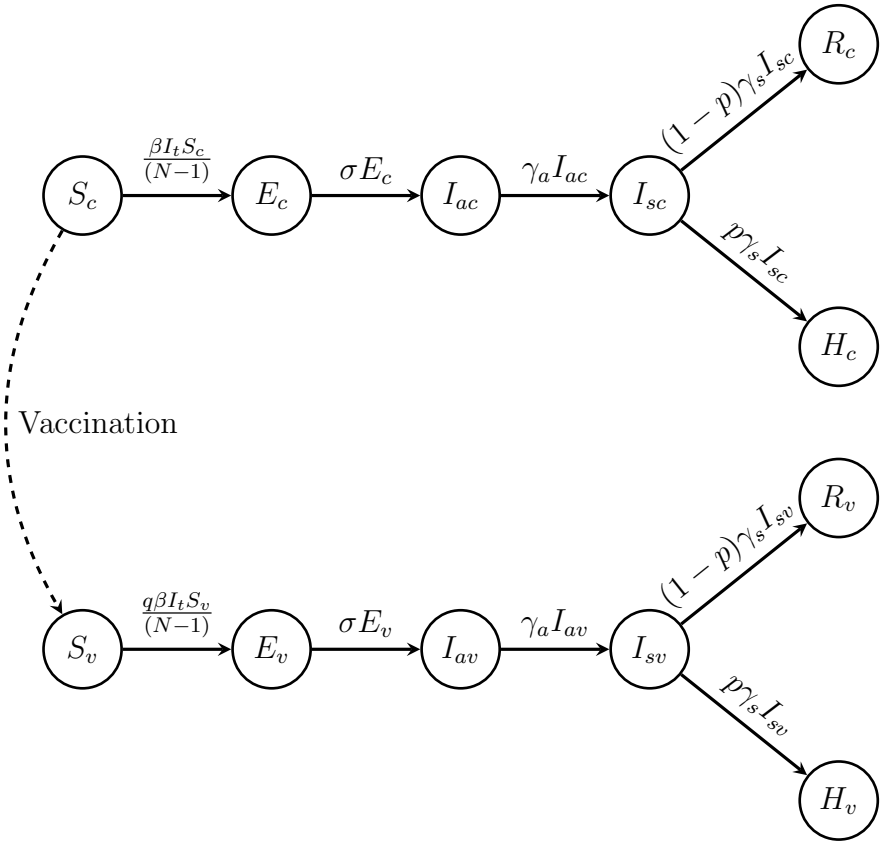


Figure 4.2: Transition diagram for the SEIIRH model with two arms, where $q$ is effect of the vaccine. Note that $I_t$ is the total number of infected individuals in the population ($I_{ac} + I_{sc} + I_{av} + I_{sv}$).

two-arm SEIIRH model; See Figure 4.2 for a diagram of this model containing a control group and a single vaccinated group. We chose to illustrate the method using only a two-arm model because it sufficiently demonstrates the difficulties of this method and sufficiently tests the computational efficiency. This also highlights a shortcomings of this method, where it cannot easily be modified to a different model as we need to account for all the 'edge' cases and the unique sequence of interim forced events (as discussed in Chapter 2. See Figure 4.2 for a diagram of this model.

### 4.6.1 Simulated Data

We are collecting the observed data to daily precision and every week updating our knowledge for the posterior distribution of the parameters. The structure of this data is an array, where each column represents the number of transitions from $I_a$ to $I_s$ in either arm that happened that day and each row represents a day. Thus the data will be a $7 \times 2$ array for each week. However for the purposes of demonstrating this method, we will be evaluating the posterior at the end of the outbreak, so the data will be a $2 \times T$ array, where the outbreak lasts for $T$ days. Note also we will not be implementing designs yet, as we are simply concerned with the behaviour of the likelihood method. Instead we will be using the method on a population where a third of the individuals are in each vaccinated group and that will not change over the course of the outbreak. Another important note to consider is that to evaluate the posterior distribution, we will be evaluating the likelihood over a range of values that the vaccine effect can take ($[0, 1]$) and then normalising.

In the following we will go into detail of how we adapted Algorithm 2.4 to this specific model, with close attention paid to edge cases.

Assume we have observed events $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_T\}$, then we will focus on day $t \in [0, T]$, as we are modifying the propagation step, and the following process is repeated for each day.

### 4.6.2 Forced Observed Events

In Algorithm 2.4 we generate the times of the observed events, assuming we have observed $(y_t^c, y_t^v)$ events on day $t$. Then we must force $y = y_t^c + y_t^v$ total observed events over this day. In Algorithm 2.4, we will sample $y$ random numbers from a standard uniform distribution and sort them. However we now need to order these forced events randomly, as the order in which they are forced will affect the result. Specific to this problem, the observed transition

in the control group $I_{ac} \to I_{sc}$ is event Type 3, and the observed transition in the vaccinated group is $I_{av} \to I_{sv}$ is event Type 8. To randomly order the forced events we sampled, without replacement, from a set that contains $y_t^c$ Type 3 events and $y_t^v$ Type 8 events with a total of $y$ entries. This sample is the beginning of the stack $\psi$, that keeps track of the next events that need to be forced. Note the stack $\psi$ is the event-type stack and it has entries that correspond to the elements of the time stack $\omega$.

### 4.6.3 Forced Interim Events

The next step that requires care is when we are checking if the system will remain consistent for the next forced event $e_n$. Without loss of generality assume the next forced event is in the control group (event Type 3) and at time $t_n$. Then at this point, there are four cases that need to be considered.

**Case 1.** In this case, there is at least one individual in the $I_{ac}$ compartment. If this is true, we do not need to force an interim event and instead can force the observed event at time $t'$, by adding event Type 3 to the top of $\psi$ and, $t'$ to the top of $\omega$.

**Case 2.** In this case, there are no individuals in $I_{ac}$, but there is at least one individual in $E_c$. In this scenario, we need to chain the proceeding events in order to force this sampled event. This is achieved by forcing the interim event $E_c \to I_{ac}$ (event Type 2); recall from Algorithm 2.4 this requires us to sample the time, $t'$, of this event from a truncated exponential distribution. Then we would add event Type 2 to the top of $\psi$ and $t'$ to the top of $\omega$. This ensures the interim forced event is implemented before the next forced observed event.

**Case 3.** In this case, there are no individuals in $I_{ac}$, and there are no individuals in $E_c$. The majority of the time this case occurs (see Case 4), we can simply chain together the two interim forced events; note that the order in which this happens is important. First, we will sample the time for the transition $S_c \to E_c$ (event Type 1) from the truncated exponential distribution, $t'$. We then add event Type 1 to the top of $\psi$ and $t'$ to the top of $\omega$.

**Case 4.** There is one caveat to Case 3 that needs to be addressed that is a result of having two arms that can cross infect. This is a very specific circumstance when we are in Case 3 in the control group and there is only one individual in the Exposed compartment in the vaccinated group $E_v$. This is an issue because we want to make the transition $S_c \to E_c$, however for that to happen there must be at least one individual in one of the other four infectious compartments. In this scenario, we must force the interim event $E_v \to I_{av}$ (event Type 7) before we force event Type 1, by adding event Type 7 to the top of $\psi$.

It is important to note here that once the top transition on the stack has been made (i.e., popped off the stack), we re-evaluate the consistency of the state. This is how we chain the events together. For instance, say we are in Case 2, and the top element of the stack is event Type 2. Once that transition has occurred, we once again check which case we are in, and in this scenario that would be Case 1. However in the propagation step, random events are still occurring, it is entirely possible that the random events may allow us to skip interim forced events. For instance, if we are in Case 4, and the transition is enforced, then after that transition, by random chance an individual transitions from $S_c$ to $E_c$, then we could jump to Case 2, bypassing Case 3. Thus care must be taken when implementing random events as to check if this has changed the sequence of events that must be forced.

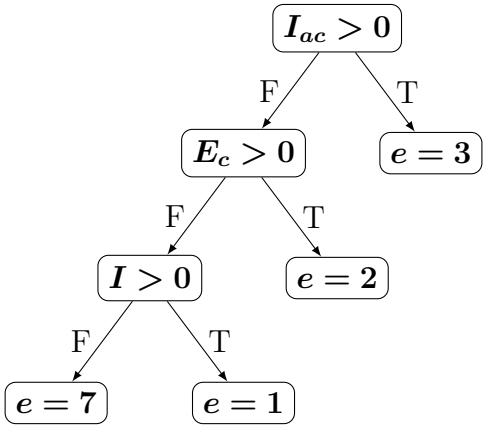


Figure 4.3: A decision tree for the interim forced events, where $e$ is the next forced event to be put on the stack and $I_t = I_{ac} + I_{sc} + I_{av} + I_{sv}$.

We have discussed these cases for the control group, however since both groups are identical (except for the vaccine effect) the same principles apply for the vaccinated group. In Figure 4.3 we have outlined all these cases in a decision tree.

### 4.6.4 Modified Rates

Another key component of Algorithm 2.4, is how we modify the rates of the underlying simulation process to ensure the system remains consistent. Recall, while propagating the particles forward a day, we have to force the number of observed events and still allow the process to have random transitions that follow the same rates as the model. This can lead to issues with regard to the system's consistency, for instance if all infected individuals recover before we have forced all the observed events (that have occurred in the data to date). There are some general rules for adjusting rates for particle filters. The first is that the rate of the observed event (in this model, event types 3 and 8) is always 0; this is important because that means the observations only happen at the forced times. Second, the rate of the next interim

forced event is always 0; this ensures the propagation step can not make that transition randomly before forcing it. Finally, if we observed $N_e$ total infections in the data, and if there is a total of only one infectious or exposed individual in the population then the rate of recovery or hospitalisation in either compartment is 0. Similarly if $E_c + I_{ac} + I_{sc} + E_v + I_{av} + I_{sv} = 1$, and the number of already forced events is less than $N_e$, then we set the rates of recovery and hospitalisation in both compartments to 0. These conditions are crucial, because otherwise the system can prematurely end the simulated outbreak and then that particle will not match the data.

### 4.6.5 Limitations

In Section 4.9, we have performed many likelihood evaluations for a range of q-values using this method and we have summarised the computational time and precision of the method. We show that for the two-arm model, this approximation method is reasonably accurate. However, it is too slow in the context of optimal experimental design. Note the times we have calculated are from using the two-arm SEIIRH model, which is a simpler and smaller model compared to the full model that will be used in Chapter 5. However, the computing time taken for a single likelihood evaluation using a particle filter on this simpler model is intractable, thus extending the model will only make this worse. Therefore the two-arm model highlights the need for a significantly faster approximation method for likelihood estimation.

In the following sections, we will demonstrate two methods that are significantly computationally faster than a particle filter, with a minor sacrifice of precision. We will discuss implementing them on the two-arm model and compare timing and accuracy of results to the particle filter in Section 4.9.

## 4.7 Gaussian Diffusion Approximation Method

As can be seen in Section 4.9, using a particle filter to approximate the likelihood is an accurate method, however it is very slow. In the context of optimal experimental design, we have a need for a significantly faster method, at the sacrifice of some precision.

We took the principles of the method outlined in [23], where they discuss using a density-dependent approximation to the stochastic process. Recall from Section 2.2.3, that for a large population ($N$), the density-dependent stochastic model can be approximated by a deterministic process. This method is significantly computationally less intensive as we are simply solving the differential equation outlined in [23] for a range of parameter values, which can be done efficiently and accurately using differential equation solvers.

### 4.7.1 Outline of the Method

Much like with a particle filter we are interested in the likelihood of the observed data given the parameters, and in this context the only unknown parameter is the vaccine effect $q$. Theorem 2.2.1, gives us a form for the mean outcome of all stochastic realisations for a set of fixed parameter values. Then the likelihood of each value of $q$ can be calculated by how 'closely' the stochastic realisation matches the deterministic trajectory.

Let's still assume we only observe the number of transitions from $I_a$ to $I_s$ for both arms of the model, at the end of each day. These observations can be represented as $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_T\}$, where $\boldsymbol{y}_t$ is a 2-element vector containing the counts for the number of transitions in the control and vaccinated compartments, respectively. To calculate the likelihood of the data over these $T$ days, this method would require us to solve the deterministic trajectory for a value of the unknown parameter $q$. This deterministic trajectory is a multi-

dimensional ordinary differential equation (ODE) which can be numerically solved using a solver (e.g., odeint from scipy in python) [28]. We will go into more detail on how we built the differential equation for this problem in Section 4.7.2.

From the solution of the differential equation, we have an approximation of the mean number of transitions between $I_a$ to $I_s$ compartments as well as an approximation of the variances. We will refer to the counts of the number of transitions in the deterministic solution as the deterministic observations and denote them as $\{\boldsymbol{z}_1^q, \boldsymbol{z}_2^q, \ldots, \boldsymbol{z}_T^q\}$, and the variances that accompany these observations can be represented as $\{\boldsymbol{\Sigma}_1^q, \boldsymbol{\Sigma}_2^q, \ldots, \boldsymbol{\Sigma}_T^q\}$. Note that the solution to the ODE provides us with the full covariance matrix for every compartment in the model, however we are only interested in the sub-covariance matrix that corresponds to the two observed transitions. Under this approximation, we have $f(\boldsymbol{y}_t|q) \sim N(\boldsymbol{z}_t^q, \boldsymbol{\Sigma}_t^q)$, and hence the likelihood of the data is,

$$\ell^q = \log(f(\boldsymbol{y}|q)) = \log\left(\prod_{t=1:T} f(\boldsymbol{y}_t|q)\right) = \sum_{j=1:T} \log(f(\boldsymbol{y}_t|q)).$$

### 4.7.2 Implementation

An issue we encountered when implementing this approximation, is trying to calculate the number of the observed transitions over a day from the solution of the ODE. Specific to this problem, it is possible to work this out because we know the flux in and out of each compartment over a day. However in order to obtain the variance of these numbers, we need an element of the differential equation to be counting the number of observed transitions. We know that the rate of transition between the two infected compartments is $\gamma_a I_{ac}$ (for the control group), and $\gamma_a I_{av}$ (for the vaccinated group), so we added two new DEs to the deterministic model, and label these counting

DEs by $C_c$ and $C_v$ respectively. These two differential equations count how many individuals moved between the infected compartments. However these will become a cumulative count after a day has progressed, which is not what we want.

To fix this we solved the ODE forward a day, took the state at the end of the day and then leaving the 12 states from the original model unchanged, we reset $C_c$ and $C_v$ to be 0. This results in those compartments counting the number that transitioned in the day, rather than cumulatively over some period of days. We save the counts in these components at the end of the day before setting them back to 0. The result of this was a solution to the deterministic model that also counted the number of transitions we are interested in.

Another consideration that needs to be made is when including the variance in the deterministic model, where all variances and covariances that correspond to $C_c$ and $C_v$ need to also be set to 0. The original model consisted of 12 compartments, and we added two more for counting, so the covariance matrix will have 196 ($14^2$) compartments, and we must also set the 52 elements that are the 13th and 14th rows and columns of the matrix to 0; resetting the values similarly to the mean components. This is because when we set $C_c$ and $C_v$ to 0, we are setting them to constants which have zero variance; then we can calculate the variance of the count over the day.

### 4.7.3 Limitations and Advantages

We have summarised the results in Section 4.9, where we see that this method is also computationally intractable in the context of optimal experimental design. In order to obtain the variances of the observations, we must calculate the covariance matrix for all components in the ODE; this results in a 210-dimensional ODE. An ODE of this size is still computationally intensive and unfortunately we were unable to find a way around this issue.

Another limitation of this method is that the stochastic process (observed data) is only one realisation of what could happen for given model parameters. These trajectories can vary significantly, which can be a problem if by natural variation the observed data matches a deterministic trajectory, generated from different parameter values, more closely. This is unfortunately unavoidable in this method, however we can minimise the effect by averaging over more simulations in the experimental design stages.

Similarly, another limitation of using deterministic models is that they tend to over-estimate the true mean. This is because the deterministic solution will not account for the probability of the disease fading out early (if $R_0 \geq 1$). This is a limitation of this method, because when the observed data is from an outbreak that faded out, we would not be able to make reasonable estimations of the likelihood. However, in the context of optimal vaccine trials, if a realisation of the outbreak ends early, one could say that the endeavour to slow down the disease and reduce deaths was successful and precise evaluation of the vaccine efficacy is not as important.

This is particularly true in Section 5.3, where the utility used does not involve the posterior estimates of the parameters. As such if the disease fades-out, the utility is not adversely affected by poor estimation and if it were to happen in a real-world scenario, we could estimate the likelihood using a slower and more precise method, for example a particle filter.

Overall, this method has the potential to be reasonable for approximating the likelihood, but it takes too long to solve the required system of differential equations. We did circumvent this by solving the system of differential equations for a range of parameter values and saving the covariance matrix and means over time. This sped up the process, but it will not be feasible when we implement the method in an experimental design framework. At the end of a week, a certain number of people are vaccinated, changing the initial state for the differential equation, making the saved information use-

less. So, we constructed a computationally-efficient method where we reduce the dimension of the ODE in Section 4.8.

## 4.8 Poisson Approximation Method

Taking the principles from the above method, we modify it simply to increase the computational efficiency, with an expected loss in precision. Previously, we were solving a 210-dimensional ODE and now seek to lower the number of dimensions while still calculating the likelihood in a similar manner.

Broadly, the modification involves using the mean trajectory from the system of differential equations, and assuming that the realisation from the stochastic model fits that of a Poisson distribution. Thos is elaborated further in Section 4.8.1. This is a significantly faster method as it requires us to solve only a 14-dimensional ODE.

### 4.8.1 Outline of the Method

The actual method is nearly identical to the previous one discussed; we solve the differential equation (now just the mean components), and estimate the likelihood of the data given the assumption that the data follows a Poisson distribution. We observed that if the rates of a Markov chain is held constant, then over a short period of time, the number of arrivals is Poisson distributed with mean given by the corresponding rate. This will allow us to approximate the Gaussian error structure with a Poisson distribution, where we only need to know the mean. In this problem, of course the state of the Markov chain is not held constant, however over the time frame of a day (the precision of our observed data) it typically does not change drastically.

Given the solution to the system of differential equations is the mean trajectory of all stochastic realisations (for a set of model parameters) asymptotically, then we can simply calculate the probability at time $t$ of observing

$\boldsymbol{y}_t$ infections, given the mean number of infections is $\boldsymbol{z}_t^q$ under a Poisson distribution. However, another assumption must be made: that the counts of the number of infections in both groups are independent. Given this assumption we can simply multiply the two probabilities at each time $t$. Let $f(y_t^c|q) \sim \text{Poi}(z_{t,c}^q)$ and $f(y_t^v|q) \sim \text{Poi}(z_{t,v}^q)$, be from independent Poisson distributions. Then the likelihood of the observed data is,

$$\ell^q = \sum_{j=1:T} \log(f(\boldsymbol{y}_t|q)) = \sum_{j=1:T} \log(f(y_t^c|q)) + \sum_{j=1:T} \log(f(y_t^v|q)).$$

### 4.8.2 Limitations and Advantages

Similar to the method outlined in Section 4.7, this method has the same limitations as far as the precision, but likely to be less precise. However, this method required us to assume that over a day, the state is approximately constant, which allows the use of a Poisson distribution for the error structure. With the population size being sufficiently large the Poisson assumption is reasonable, however care must be made to acknowledge that using a Poisson error structure is a further approximation on an already approximate method. Having said that, this method is significantly quicker than either of the two previous methods that we have explored and in the context of the optimal experimental design in Chapter 5, this is crucial. See Section 4.9 for the analysis of this method.

Both the Gaussian method outlined in Section 4.7, and this one have a significant advantage over the particle filter, in that they are easily adaptable to a specific problem. In the case of the particle filter we had to manually handle all of the forced and interim forced events as well as modifying the rates for the edge cases. As the complexity of the model increases, this becomes a very tedious and difficult task. However when we look at approximating the likelihood using the solution to a system of differential equations, pro-

vided we change the system of differential equations correctly, the methods are easily modified to any model. A major advantage of this method is its performance with regard to both the reasonable precision and importantly the significant improvement in computational efficiency, as presented in 4.9.

## 4.9 Results

In the previous sections in this chapter, we have explored three methods in detail for approximating the likelihood of the observed data given the model parameters. Each method stepped further away from statistical precision, due to requiring more assumptions and approximations. However it was discussed that this sacrifice in precision was made up for in the gain in computational efficiency. In this section we will discuss the accuracy and speed of these three methods in comparison to each other.

In the following results, we took a true value of $q$ (we chose to test the results on five different $q$-values: 0.1, 0.3, 0.5, 0.7 and 0.9) and generated data from the two-arm SEIIRH model using the SSA. We took the observed data (number of transitions from asymptomatic to symptomatic in both groups in each day) over the whole outbreak and evaluated the posterior distribution. Note that to evaluate the posterior distribution we evaluated the likelihood at 30 $q$-values equi-spaced from 0 to 1, and then normalised these 30 points to get a normalised posterior distribution. We then find the *maximum a posteriori* (MAP) that is a measure of the single-value summary of central location for the posterior distribution. Note that in the particle filter we need to set the number of particles used as a parameter. We chose to use 500 particles as we decided this led to an acceptably smooth and precise posterior distribution.

In Figure 4.4 we have included a set of side-by-side boxplots of the estimated MAPs (from 2500 samples) for each method and over a range of

$q$-values the data was generated using. We included this plot to demonstrate the median and variance in the estimates of the MAPs. Note, we have included horizontal red lines that correspond to the true values of $q$ that were used in the generation of the data to assist in assessing the results. We can observe from Figure 4.4 that all methods seem to underestimate the true value, however this is probably unavoidable and is not a result of the particular method. It is worth noting that for large values of $q$, the particle filter performs much better at achieving an unbaissed estimate for $q$, we are not entirely sure why this is, but possibly when $q$ is small, the outbreaks end quickly and insufficient data is collected. We can also observe that the median MAPs of the Gaussian and Poisson methods are identical for each value of $q$. However, the Poisson method has a slightly larger variance in the estimate than the Gaussian method. All three methods seem to be appropriate in estimating the MAP of the unknown parameter, however the particle filter is, as we expected, the most precise.

The precision of the Poisson method is negatively impacted by the approximations and assumptions we made. However, we made this sacrifice for the sake of computation efficiency. Table 4.2 displays the mean MAP estimate and time for each method, over the range of the values of $q$. From this table we can see that the particle filter took approximately 6000 seconds per posterior evaluation, while the Gaussian method took approximately 25000 seconds per posterior evaluation. Both of these are too slow, especially when applied to a more sophisticated model and when the posterior requires more likelihood evaluations. We can also see that the Poisson method took approximately 2.5 seconds per posterior evaluation, which is a significant improvement in computational speed for an apparently minor sacrifice in precision.

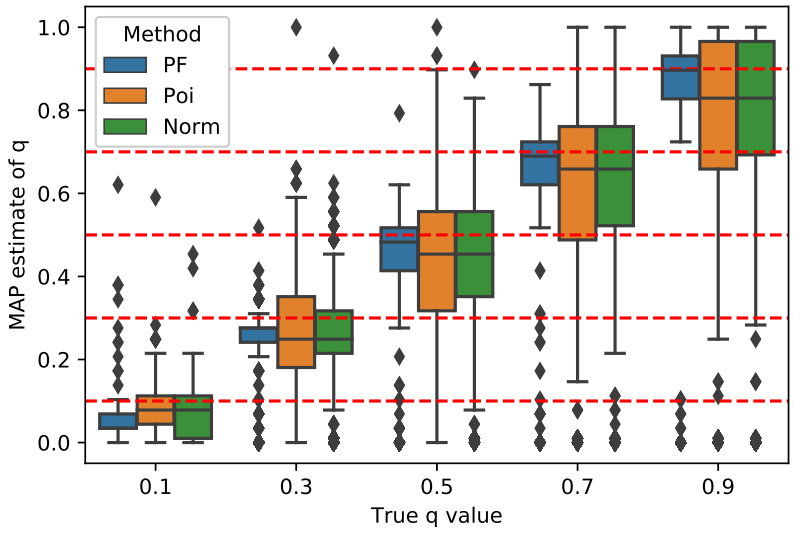


Figure 4.4: Side-by-side boxplots of the MAP estimates of the unknown model parameter $q$. The data was generated with $q$ values, 0.1, 0.3, 0.5, 0.7 and 0.9. We have included the three methods under investigation and the red horizontal lines represent the true value of $q$.

## 4.10 Summary

In this chapter we have introduced a three-arm SEIIRH model and discussed partially-observed data and methods of estimating the likelihood. We reduced the complexity of the model to a two arm model which simplifies the process of exploring three likelihood estimation methods in detail. We investigated the performance of these three methods with respect to the accuracy in MAP estimation and computation time. We showed that the Poisson modification to the method outlined by [23], showed significant improvements in the computation speed for a minor sacrifice in MAP estimation accuracy. It should be noted that we demonstrated the method on a simplified model, because the Particle filter and Gaussian method become too computation-

Table 4.2: The average MAP estimates from the three methods investigated of the model parameter $q$. The $q$ value column is the value of $q$ the data was generated from and the averages are from 2500 unique datasets. The time is the time taken for the full posterior evaluation and is in seconds.

| | **Particle Filter** | | **Gaussian** | | **Poisson** | |
|---|---|---|---|---|---|---|
| q-value | MAP | Time (s) | MAP | Time (s) | MAP | Time (s) |
| 0.1 | 0.057 | 5605 | 0.086 | 25482 | 0.092 | 2.45 |
| 0.3 | 0.265 | 6027 | 0.277 | 25323 | 0.285 | 2.36 |
| 0.5 | 0.476 | 6405 | 0.467 | 25447 | 0.473 | 2.32 |
| 0.7 | 0.684 | 6507 | 0.659 | 24037 | 0.660 | 2.17 |
| 0.9 | 0.894 | 6405 | 0.836 | 22781 | 0.832 | 2.05 |

ally demanding on the full model. This would be further emphasised when implemented in an optimisation routine. The results above are sufficient to justify using the modified Poisson method on the three-arm SEIIRH model. As such in Chapter 5 we will explore using this method for optimal experimental design on a three-arm SEIIRH model with competing vaccines.

# Chapter 5

# Optimising Vaccine Trial for Competing Vaccines

## 5.1 Introduction

This chapter is the culmination of the previous two chapters; we will take the optimal experimental design principles we developed in Chapter 3 and apply them to a more realistic problem using the fast likelihood evaluation method developed in Chapter 4. As discussed in Section 4.1, the assumptions made in Chapter 3 limited the practical application of the model, but had the benefit of reducing the complexity and computational requirement. By making these assumptions we were able to develop understanding of how optimal experimental design might be applied to adaptive vaccine trials. In this chapter we will be performing similar optimal experimental design, and will modify these assumptions resulting in a more practical model. These modified assumptions include having partially-observed data, that is we can only observe some of the transitions in the compartmental model, which led us to develop a fast likelihood approximation method outlined in Chapter 4.

Table 5.1: The model parameters used in this chapter for the SEIIRH model, with values based on a COVID-19 model in [22].

| Parameter | Value |
|---|---|
| $\beta$ | 0.4 |
| $\sigma$ | 0.3125 |
| $\gamma_1$ | 0.5 |
| $\gamma_2$ | 0.130 |
| $p$ | 0.0515 |

### 5.1.1 The Model

In Section 4.2 we developed and discussed a three-arm SEIIRH model with partially-observed data that addresses many of the unrealistic modelling assumptions made in Chapter 3. We chose this SEIIRH model since previous works ([22]) suggested this model may be appropriate for COVID-19, and provided estimated parameters for this model. See Table 5.1 for a table of parameters used for modelling COVID-19.

In Table 5.1, $\beta$ is the effective infection rate parameter of the virus, $1/\sigma$ is the average time spent in the exposed compartment, $1/\gamma_1$ is the average time an individual spends in the asymptomatic infectious compartment, and $1/\gamma_2$ is the average time spent in the symptomatic infectious compartment. Finally, $p$ is the probability an individual is hospitalised, given they have been infected with the virus.

Note that the $\beta$ value estimated from the paper was significantly larger than quoted above, however the original value resulted in outbreaks that spread through the population so quickly that there wasn't time for the adaptive designs to have an affect. We artificially lowered the $\beta$ value to slow down the initial wave of infections, and we can justify this by assuming

individuals wore face masks and socially distanced, for example. This will also lead to many extensions in the model with regards to isolation and lock down jurisdictions that will be discussed in Section 6.1.

Recall that the model in Section 4.2 assumed partially-observed data. Specifically this means we are observing the number of individuals that transition from asymptomatic infectious to symptomatic infectious over a day. At the end of each day, we have a count for this number in the control group, and the groups corresponding to the two vaccines under trial.

Recall the vaccine effect is a multiplicative effect, $q$, on the infection rate that corresponds to the vaccine administered to the individuals. From Chapter 3 we changed the vaccine effect to be restricted to be between 0 and 1 (100% effective to completely ineffective, respectively). For this chapter we change this assumption to move towards a more realistic vaccine effect. However, as discussed in Section 4.2, this lead to the development of the three-arm model. This model allows us to compare two vaccines ($q_1$ and $q_2$) simultaneously and the adaptive design will inevitably decide which vaccine is better (or potentially they are approximately equally beneficial) and ideally end up vaccinating a majority of the individuals with the superior vaccine.

### 5.1.2 Design parameters

The design component of this model will be similar in nature to that of Chapter 3, however since we have two vaccinated groups now the way we determine how to administer the vaccines is different. We will be performing analysis of the data every week and then vaccinating accordingly again at the end of a week, this will be discussed in more detail in Section 5.2. We made this change because it is unlikely to practically be able to make decisions based on daily data.

Since we are comparing two vaccines, we have to estimate the joint posterior for the vaccine effect ($q_1$ and $q_2$) of both vaccines. From this joint

posterior distribution we are able to extract the marginal distributions and hence the maximum a posteriori (MAP) estimate of the individual vaccine effects. Note that this is equivalent to considering the bivariate MAP as the two distributions are independent. Any decisions made during the experiment about which vaccine we believe is superior is made based on the MAP estimates. At the end of each week, we will estimate the joint posterior distribution of $q_1$ and $q_2$ and hence MAP estimates, and then administer a proportion of the weekly vaccinations based on a ratio, $r = \frac{q_1}{q_1+q_2}$, where we administer $r \times 100\%$ of the weekly vaccines into group 2 and $(1-r) \times 100\%$ of the weekly vaccines into group 1. This ratio is reasonable, however this ratio does not contain any design parameters and hence we cannot explore designs that modify this.

In the scenario when the two vaccine effects are very similar, the ratio will allocate approximately half the individuals into each vaccinated group. We introduced a design parameter $\omega$; this parameter modifies the ratio so that the proportion of individuals allocated to the better vaccine proportionally is larger. This parameter modifies the ratio by raising each element to it, $r_\omega = \frac{q_1^\omega}{q_1^\omega+q_2^\omega}$. We will restrict $\omega$ so that it can range between 0.2 and 5. To demonstrate how $\omega$ affects this ratio, let's assume without loss of generality that $q_1 < q_2$. Then if $\omega = 1$, then we have the ratio $r$ from earlier. If $\omega > 1$, then the proportion allocated to group 1 is larger than if $\omega = 1$. Similarly if $\omega > 1$ the proportion is more sensitive to the difference between $q_1$ and $q_2$ and hence will be more subtle in the allocation to the vaccine groups. The purpose of this parameter is to allow the optimisation routine to explore how sensitive the ratio should be to the difference in the two vaccine efficacy values.

As discussed in Chapter 4, computational speed is a major issue for a model of this complexity and the partially-observed data modelling assumptions we have made. As such we require a small design space so that we can

find an optimal design in reasonable time. We have decided on three design parameters as follows:

- $v_0$ - The number of individuals vaccinated initially in each vaccinated group.
- $v_1$ - The number of vaccines to be administered at the end of each week (these vaccines are shared between the two vaccines).
- $\omega$ - Power we raise the MAP estimates in the ratio, $r$, to.

Note that we can think of $\omega$ as a substitute for the two parameters $c$ and $\alpha$ from Chapter 3.

## 5.2 Likelihood

In Chapter 4 we explored different likelihood methods for partially-observed data, and compared these methods with respect to both computational speed and accuracy. We developed a method for estimating the likelihood that is acceptably efficient and precise. In the previous chapter, we implemented this method on a two-arm SEIIRH model and data from a full outbreak. However when considering a design, we need to estimate the posterior each week and then make decisions based on that. This leads us to potential issues and considerations that need to be addressed.

**Three-arm model extension.** Now that we are estimating the likelihood for the three-arm model, there are two parameters we need to estimate, resulting in a joint posterior distribution. As discussed in Section 4.3 for a three-arm model, we need to estimate the likelihood of the data over a grid of parameter values. In this situation, we chose a $20 \times 20$ grid of $q$-values as a trade-off between computational time and precision in the utility estimate.

**Weekly likelihood calculations.** Since the posterior is estimated at the end of every week and we update our estimate each week, we needed to adapt how the posterior is estimated. First, the ODE is solved forward one week, which provides the expected trajectory of the week we have observed. Then we can calculate the log-likelihood contributions from each day over the grid. These contributions are calculated for each day of the week and we add them together to obtain a log-likelihood estimate for each pair of $q$ values in the grid.

Note that the normalised log-likelihood is proportional to the log-posterior since we chose a standard uniform prior distribution. Also, during a simulation there are two different state spaces we need to manage. The first is the state space from the stochastic simulation of the outbreak, which we will refer to as the stochastic state. The second is the state space that arises from the ODE solution used in the log-likelihood evaluation which we will refer to as the DE state. After solving the DE forward a week for a pair of $q$-values, the state at the end of this week must be saved so that when we evaluate the posterior the following week; we can solve the DE starting at the DE state saved from the end of the previous week. This is particularly important when we begin vaccinating individuals which changes the stochastic state, and hence we must modify the DE state for consistency.

**Weekly vaccinations.** In Chapter 3 we chose arbitrary model parameters that resulted in an outbreak that spread fast. As a result we made decisions every day. However for this COVID-19 model, the outbreak is slightly slower and in the interest of realism, we will be vaccinating individuals at the end of each week. Due to the partially-observed data, we do not know how many individuals are in each of $S_c$, $E_c$ or $I_{ac}$ compartments, we only know the sum of all three. Hence at the end of each week, we must sample without replacement from the total, which can result in vaccinating a combination

of Susceptible, Exposed or Infectious Asymptomatic individuals. However this can lead to inconsistency in the DE state as the DE state can be different to the stochastic state. For the log-likelihood evaluation, we only know how many individuals have been vaccinated ($\ell_1$ into vaccine group 1 and $\ell_2$ into vaccine group 2), we do not know how many of each type (i.e., which compartments). It is important to note that if an Exposed or Infectious Asymptomatic individual is vaccinated, it has no affect on them and their transition to infectious symptomatic will be counted in the compartment they have been moved to. With regard to the DE state, we need to move $\ell_1$ individuals to vaccine group 1 from the control group. To approximate how many from each compartment moved to each vaccine group we find the proportion of susceptible, exposed and infectious asymptomatic in the control group, and move individuals according to those proportions. This process is then repeated for the second vaccinated group, however these two states may not be always consistent, for instance there may be less than $\ell_1 + \ell_2$ in the control group in the DE state; if this is the case, we vaccinate everyone available and follow the principle above.

This will result in the DE state corresponding to vaccinating the expected numbers from each compartment, respecting the number of individuals available. This is important because the DE is, asymptotically, the expected trajectory of the stochastic model and combined with the expected numbers vaccinated in each compartment results in the approximate mean outcome for all stochastic realisations.

## 5.3 Utility

As discussed in Section 2.5, in optimal experimental design we require a utility function that can be evaluated on a design that is a measure of its effectiveness. Recall a utility is in principal an integral over the parameters

we are estimating and in this problem we want a design that minimises the expected number of hospitalisations over the outbreak in all groups. As discussed in Section 4.3, we will trial each design over a $20 \times 20$ grid of possible parameter values and then integrate the volume under this surface. The utility is then the integral under the surface of these grid of points, which can be evaluated numerically. The density of this grid affects the precision of the estimated utility. However, as we increase the number of points in this grid, we drastically increase the computational load, which can be problematic in this problem.

## 5.4 Results

We implemented the INSH algorithm [25], as discussed in Section 2.5.1 and outlined in Algorithm 2.5, for 6 generations with 100 initial designs. At the end of each generation we kept the 20 designs that performed best (including the current overall optimal design) and perturbed each of these designs 5 times. The perturbation kernel was a mixture of discrete-truncated uniform distributions (for design parameters $v$ and $v_1$) and normal distributions (for design parameter $\omega$):

$$\begin{aligned} v^{(w)} &= v^{(w+)} + U_d(-25, 25); \\ v_1^{(w)} &= v_1^{(w+)} + U_d(-25, 25); \\ \omega^{(w)} &= \omega^{(w+)} + N(0, 0.07). \end{aligned}$$

Note that when perturbing the set of designs, we take care to ensure that the sampled values remained within the limits outlined in Table 5.2. After 6 generations the best design was found to be, $d^* = (299, 363, 0.901)$, with an expected number of deaths, $u(d^*) = 2.979$, that is to say that under our design we expect 2.979 individuals to be hospitalised out of the population of 1000.

Table 5.2: Maximum and minimum values for the design parameters, where $N$ is the total population (1000), $I_c a$ is the number of individuals in the asymptomatic control compartment and $I_c s$ is the number of individuals in the symptomatic control compartment.

| Parameter | Min | Max |
|---|---|---|
| $v$ | 0 | 300 |
| $v_1$ | 0 | $N - (I_c a + I_c s)$ |
| $\omega$ | 0 | 1 |

### 5.4.1 Assessing Convergence to Local Minimum

Now we have implemented INSH, we will check that it did indeed converge to a local minimum. First we will check that it converged, then we will check that the convergence was to a local minimum. Figure 5.1 shows a set of plots that show the algorithm converging to a set of designs. Each plot is a projection of the labelled design parameter onto one dimension. On the vertical-axis we are plotting the generations of the algorithm. The colour bar indicates the value of the utility function for each design, where blue is a poor design and red indicates better designs. It is important to note the range on the horizontal-axis is different for each plot.

In Figure 5.1, we can see that each parameter does converge to a narrower set of points. While this convergence happens at different rates for each parameter, all parameters seem to have converged completely by the $6^{\text{th}}$ generation. Noting that the parameter, $v_1$ and $\omega$ did not converge until very late in the optimisation routine.

The INSH algorithm converged to a set of best designs. It is difficult to know if this set contains the true optimal design, but we can assess whether it is indeed a local minimiser of the utility function. Figure 5.2 shows a set of plots assessing the convergence to a local minimum. In this figure for each

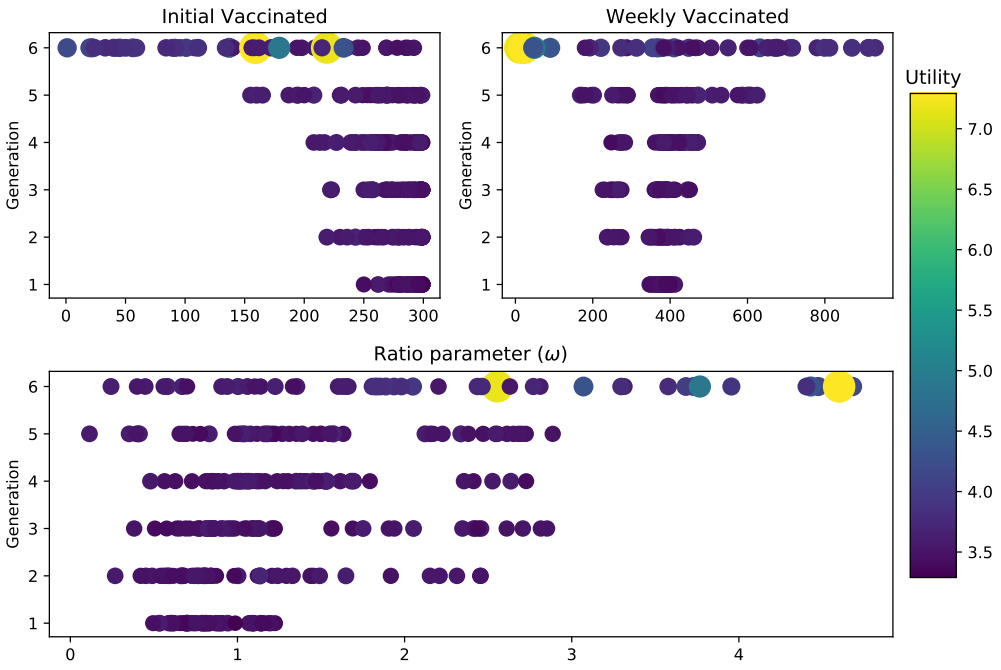


Figure 5.1: Shows INSH converging to a set of designs, where the colourbar (and size of point) indicates the value of the utility function. The generation is plotted on the vertical-axis reading down the plot and the horizontal-axis is the value of that design parameter. Note that at each generation the 100 points that were considered by INSH are plotted and we see them converge to a set of points as the algorithm evolves.

plot we held the other design parameters at the optimal design and varied the parameter under investigation over a coarse grid on its support. We then evaluated the utility for these designs over the grid and plot the utilities against that dimension of the design space.

In Figure 5.2, we can see that the value of the utility function is rather insensitive to the value of the parameters. This is in stark contrast to the equivalent plot in Section 3.3.1, where we observe that all parameters converged to a local minimum. This result can be due to many factors. First is the model structure and parameters do not allow the design to learn about

the model and vaccinate accordingly. Second is that our utility is not calculated to a high enough precision, this is most likely in the case of the parameter $\omega$, where we can see the trace plot is very inconsistent. However this is not likely to be the cause for the other two parameters under investigation, as the plot follows a clear trend wth some noise. Note that in the initial vaccinated subplot the plot curves up after 500, this is because this parameter is the number of individuals allocated to each vaccine group. When this value is greater than $\frac{N}{2}$, we allocated $v$ to vaccine group 1 and then $N - v$ to vaccine group 2. This is why we see an increase in the utility on the plot past $v = 500$ as half the time allocating the majority of the individuals to vaccine group 1 will not be beneficial.

After considering these plots to assess convergence, the algorithm seems to have explored the design space, however unfortunately it does not seem to have achieved satisfactory convergence. Figure 5.1 demonstrates that the algorithm explored the design space well, however Figure 5.2 shows that the optimal design is not very sensitive to the parameter values.

## 5.5 Getting the Most Out of the Super-Computer

Similar to Chapter 3 computation was a significant focus in this chapter. Care was taken to ensure we used the super computer resource to its full potential. Similar to Section 3.4 there are two main places we can improve the efficiency on the super computer, the way we parallelise the code over each node and how we submit jobs to the queue.

Firstly, it is important to note that optimal experimental design is essentially a set of nested loops and as far as computing is concerned, we can parallelise over any of these loops. Choosing which loop to parallelise over is a tradeoff between simplicity of data storage and the number of file writing operations. As such we chose to parallelise on the step where we trial a design

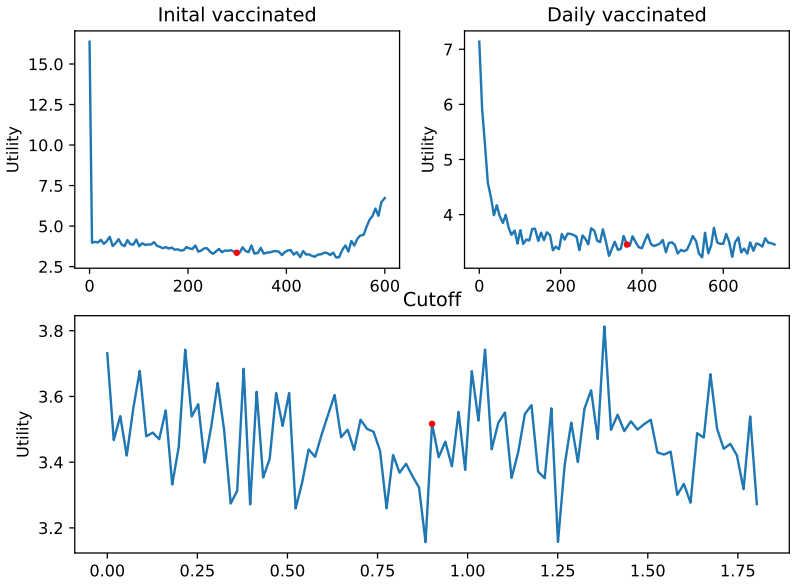


Figure 5.2: In each subplot we kept two design options held at the value from the current best design and changed the relevant parameter over the full range of possible values. The INSH-best design parameter value has been included as a red point in each subplot.

over a grid to calculate the utility. From Section 4.3 we know that to test a design fully, we trial it over pairs of $q$-values over a grid and then repeat this many times. We chose to have a node completely test a design over a grid. This led to easily writing files of information at the end of a full trial, and importantly if the super-computer crashed or our script ran out of time, we would still have saved the previous successful simulations.

Similarly, as discussed in Section 4.3 the super computer is a competitive resource so a significant bottleneck is getting to the top of the queue and running code within the allocation. It is important to note that the super computer we had access to restricted jobs to a maximum of 3 days and each

account is able to run at most approximately 60 jobs at a time, dependent on competition from other users. As such we could at most run 60 scripts for 3 days at any given time; this was an issue because we found that in order to get a precise estimation of the utility, we needed at least 10 days of code running full time per design. Thus we ensured to run each design on 4 different nodes for the full 3 days. In an ideal world, we would be able to run an unlimited number of scripts at any given time, then we could run 10 nodes per design for a full day and finish a generation in INSH in a day. However this is not possible so we found a compromise that lead to a final result.

## 5.6 Discussion

In this chapter we addressed many of the naive assumptions made in Chapter 3 and developed a new model and modelling framework to manage this. We implemented these changes and using the likelihood estimation method refined in Chapter 4, the applied the INSH optimisation routine to find an optimal design. After achieving convergence from the algorithm we investigated whether we had found a local minimum and unfortunately found that the optimal design is insensitive to most of the parameters in the design. We also discuss possible causes of this which we suggest would be worth pursuing. Another method of identifying what caused this unfortunate result, would be to run the optimisation routine again from different starting points to show this result and hence method is robust. However with the computing resources available, this is infeasible within the time frame of this project.

# Chapter 6

# Conclusion

2020 has highlighted how significant a threat global pandemics are and will continue to be, to every individual and organisation worldwide. As such when an outbreak of a virus similar to COVID-19 inevitably happens again, we as a society need a way of testing a potential vaccine's efficacy rapidly. In this thesis we have studied a concept of an adaptive vaccine trial, where vaccines are administered to the population according to knowledge that has been gained from observing the infections. That is to say if we observe that the vaccine is highly effective, then it will be administered more quickly. This concept for a trial will allow us to learn about the vaccine's effectiveness as well as slow the spread of the disease in that we will be putting vaccines out into the population.

In Chapter 3 we explored optimal experimental design for vaccine trials in a Bayesian framework, on two different models of increasing sophistication. Initially we looked at an SIRD model that could be used for a virus that killed infectious individuals with a fixed probability. We assumed the vaccine effect could range between complete immunity and making an individual twice as susceptible to the virus. We also assumed that all the data was observable with the intention of making the problem more computationally feasible. To trial the vaccine, the population all started in the control group and at

the end of each day, we collected the data we had observed, estimated the posterior distribution for the vaccine effect and promptly vaccinated some number of individuals into the vaccinated group based on this posterior. The design was a tuple of four numbers that dictated how the vaccine would be administered based upon the posterior distribution. Once this framework was developed we implemented the optimisation routine, INSH, see Section (2.5.1), and found a set of optimal design and the best design the algorithm had found in 10 generations. To confirm that the optimal design was indeed in the local minimum, we tested each design parameter was in a trough (assuming the other parameters were held at the optimal achieved by INSH) and produced a plot that showed we had successfully found a local minimum. We found that even though we had made naive assumptions about the model complexity and data observations, we encountered significant computational hurdles. This resulted in the use of the high performance computing (HPC) resource available, and care was taken to ensure we received the most out of this resource as possible.

Having demonstrated the feasibility of the approach on a simplified model, we sought to generalise it to a more realistic context. In doing so we developed the three-arm SEIIRH model with partially-observed data. This model was based on the COVID-19 outbreak that began half-way through this project, we used the epidemiological model outlined in [22]. As such we had addressed the issues of an overly simplistic model as well as model parameters that were fictitious, and we added the third arm to the model to allow us to compare two vaccines simultaneously. Trialling two vaccines simultaneously seemed more practical in that we know both are beneficial, but the model is trying to find which vaccine is most effective and the optimal design will administer more vaccines to the better group. Note that we also only observed one type of transition, from asymptomatic infectious to symptomatic infectious, which is consistent with COVID-19 testing in that people get tested when

they show symptoms. We did however assume we know the day that an individual makes this transition. Changing to observing partially-observed data had a significant impact on the likelihood estimation method, where exact calculations are impossible. We also discovered that a common method, the particle filter, for estimating the likelihood for partially-observed data was computationally infeasible in this context. As such, we took principles from a method outlined in [23], and in relaxing some assumptions we were able to develop a method that showed orders of magnitudes of improvement in computation time and for minimal sacrifice in accuracy.

In Chapter 5, we took the knowledge and framework we had developed on the simplified model in Chapter 3 and applied it to the sophisticated model outlined in Chapter 4, using the fast likelihood estimation method developed in Chapter 4 also. At this point, all that had to be done was tune the optimisation routine and set it running. Although we had access to the HPC resource and had experience in implementing similar problems, we had to overcome further challenges to obtain a result in reasonable time. With the HPC running non-stop (with the exceptions of when it was stopped for maintenance), it took over two months for the optimisation routine to converge to a set of optimal designs and hence an optimal solution.

## 6.1 Future Work

In Chapter 3 we made many modelling assumptions as this was simply an exploration step, to obtain early insights on the concept and limitations of optimal experimental design. In Chapters 4 and 5, we address many of these assumptions, however the model still leaves many further avenues of research to further this result.

First and foremost, we assumed the model parameters were known and fixed as estimating them as well as the vaccine efficacy would increase the

computation burden significantly. However we believe that if the other model parameters are known, then the optimal design will be cautious in how the quickly the vaccines are rolled out. This can potentially lead to further work where the disease dynamics are not known fully and the method may even estimate the number of compartments in the model.

Second, we assumed the population is homogeneous, however this is obviously not true particularly with the severity of sickness from the virus. For the COVID-19 model, we assumed each individual had the average hospitalisation rate. However it would be interesting to see if the population was stratified by age and gender to see how the optimal design would change with the elderly being vaccinated early. This leads further to questions about the homogenous mixing assumption, where we assume all individuals mix evenly, which is also not appropriate in that a 20-30 year old would mix with the population significantly more than the elderly and children. We can further this by implementing self isolation, so once an individual has shown symptoms and received a positive test, their interaction with the population is significantly lower.

Third, the models considered in this work did not include spatial information. It may be interesting to apply a household model to this problem where the household level can be household, postcodes, cities, states or even countries. In this we would model transmission within and between the 'households', that would allow for more sophisticated designs, that can implement travel bans. This would be similar to how Australia handled new cases, by blocking certain postcodes from mixing. Of course care would need to be taken with implementing this with vaccine trials simultaneously.

Finally, we assumed the vaccine affected an individual by modifying their susceptibility to the disease. However this may not be the case for all diseases, so we could consider modifying where in the model the vaccine effect is and how that changes the results with regard to optimal design.

# Appendix A

# Tandem queue processes

All code used for simulations can be found at my Github repository

# Bibliography


[1] CDC releases detailed history of the 2014-2016 Ebola response in MMWR. *Centers for Disease Control and Prevention*, Jul 2016.

[2] N. T. J. Bailey. A simple stochastic epidemic. *Biometrika*, 37(3/4):193–202, 1950.

[3] M. S. Bartlett. Some evolutionary stochastic processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 11(2):211–229, 1949.

[4] S. Bell, R. Clarke, S. Mounier-Jack, J. L. Walker, and P. Paterson. Parents' and guardians' views on the acceptability of a future COVID-19 vaccine: A multi-methods study in england. *Vaccine*, 38(49):7789–7798, 2020.

[5] A. J. Black. Importance sampling for partially observed temporal epidemic models. *Statistics and Computing*, 29(4):617–630, 2019.

[6] G. Casella and R. L. Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

[7] Centers for Disease Control and Prevention. Vaccine testing and approval process, May 2014.

[8] A. Doucet, N. Freitas, and N. Gordon. An introduction to sequential monte carlo methods. *Sequential Monte Carlo Methods in Practice*, pages 3–14, 2001.

[9] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

[10] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

[11] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4):403–434, 1976.

[12] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. 1970.

[13] A. M. Henao-Restrepo, A. Camacho, I. M. Longini, C. H. Watson, W. J. Edmunds, M. Egger, M. W. Carroll, N. E. Dean, I. Diatta, M. Doumbia, B. Draguez, S. Duraffour, G. Enwere, R. Grais, S. Gunther, P.-S. Gsell, S. Hossmann, S. V. Watle, M. K. Kondé, S. Kéïta, S. Kone, E. Kuisma, M. M. Levine, S. Mandal, T. Mauget, G. Norheim, X. Riveros, A. Soumah, S. Trelle, A. S. Vicari, J.-A. Røttingen, and M.-P. Kieny. Efficacy and effectiveness of an rVSV-vectored vaccine in preventing Ebola virus disease: final results from the Guinea ring vaccination, open-label, cluster-randomised trial (Ebola ça suffit!). *The Lancet*, 389(10068):505–518, 2017.

[14] H. W. Hethcote. Three basic epidemiological models. *Biomathematics*, pages 119–144, 1989.

[15] R. Kanapathipillai, A. M. Henao Restrepo, P. Fast, D. Wood, C. Dye, M.-P. Kieny, and V. Moorthy. Ebola vaccine—an urgent international priority. *New England Journal of Medicine*, 371(24):2249–2251, 2014.

[16] D. G. Kendall. Deterministic and stochastic epidemics in closed populations. In *Proc. 3rd Berkeley Symp. Math. Statist. Prob*, volume 4, pages 149–165, 1956.

[17] W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 115(772):700–721, Aug 1927.

[18] D. P. Kroese, T. Taimre, and Z. I. Botev. *Handbook of Monte Carlo methods*, volume 706. John Wiley & Sons, 2013.

[19] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[20] T. G. Kurtz. Solutions of ordinary differential equations as limits of pure jump Markov processes. *Journal of applied Probability*, 7(1):49–58, 1970.

[21] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[22] R. Moss, J. Wood, D. Brown, F. Shearer, A. Black, A. Cheng, J. McCaw, and J. McVernon. Modelling the impact of COVID-19 in Australia to inform transmission reducing measures and health system preparedness. *medRxiv*, 2020.

[23] D. Pagendam and P. Pollett. Optimal design of experimental epidemics. *Journal of Statistical Planning and Inference*, 143(3):563–572, 2013.

[24] P. Pollett and A. Vassallo. Diffusion approximations for some simple chemical reaction schemes. *Advances in Applied Probability*, 24(4):875–893, 1992.

[25] D. J. Price, N. G. Bean, J. V. Ross, and J. Tuke. An induced natural selection heuristic for finding optimal Bayesian experimental designs. *Computational Statistics & Data Analysis*, 126:112–124, 2018.

[26] S. Purkayastha, R. Bhattacharyya, R. Bhaduri, R. Kundu, X. Gu, M. Salvatore, D. Ray, S. Mishra, and B. Mukherjee. A comparison of five epidemiological models for transmission of sars-cov-2 in india. *BMC Infectious Diseases*, 21(1), Jun 2021.

[27] N. P. Rebuli. Hybrid methodology for Markovian epidemic models, 2018.

[28] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.