Imperial College London

Imperial College London

Department of Physics

Epidemiological Modelling of COVID-19

College ID:
01531221

Supervisor:
Carlo R. Contaldi

Assessor:
Gavin Davies

Project Code:
THEO-Contaldi-2

Word Count: 9998

Monday 2nd May, 2022

Examining the COVID-19 Pandemic Through the Lens of Mathematical Modelling

COVID-19 you say? Are we still talking about it? It may feel like the pandemic is slowly drifting into the realm of the past, yet scientists all over the world are still heavily invested in careful monitoring of the COVID-19 pandemic. With more than 6 million COVID-19 related deaths and over half a billion reported cases worldwide, the importance of careful examination of pandemics, in general, has become very clear over the past couple of years.

For this reason, all kinds of experts – doctors, biologists, epidemiologists, mathematicians, even physicists – have worked hard to develop advanced mathematical models to help them understand what drives the pandemic and therefore predict what will happen next. With a clear understanding of the state of the pandemic, scientists are in a better position to devise policies that help save lives and reduce the impact of the pandemic.

The first step in doing so involves building a model for the pandemic i.e. a mathematical description of its various aspects – how many people who get COVID-19 will get hospitalised, how many people could die, how many people would need to be vaccinated to achieve herd immunity, and so forth. The most common types of model used are *Compartmental Models*. With these, we categorise people into groups – *compartments* – such as infected, vaccinated, recovered, and model how people move between them. If we can accurately describe the way these compartments interact, we can make predictions about the progression of the pandemic, by, for example, examining the compartment of hospitalised people to predict the expected pressure on the healthcare system. The hardest part of mathematical modelling is ensuring that our descriptions about the interactions of these compartments are accurate, which involves fine-tuning the *parameters* of these models.

Intuitively, the best way to 'learn' what these parameters are is to observe the pandemic as it evolves, and use real data to yield parameter values – a process known as *Inference*. By developing sophisticated inference methods, scientists can obtain accurate parameters and use them to make predictions.

In our MSci project, we built basic compartmental models and investigated the various advantages and limitations of making them more and more complex. We then developed a mathematical framework to infer model parameters from COVID-19 data, using a method known as *Bayesian Inference*. However, we soon realised that the equations we derived for the probability distributions of these parameters are so complex that they cannot be solved analytically. So, instead, we developed numerical methods that approximate our solutions using random sampling. Two key takeaways with regards to optimising these numerical methods were that they can be made significantly more efficient by taking into account correlations between parameters, and the results yielded made considerably more precise by incorporating prior knowledge about the parameters, e.g. from literature.

Finally, we tested our methods by inferring parameters from real COVID-19 data from autumn of 2020, making a prediction about the trajectory of the pandemic following the lockdown in Christmas 2020. Our results successfully predicted the turning point and decrease in cases in January 2021. However, we observed that the uncertainty in our predictions grows so much with each additional day predicted that only the very near future can be reliably predicted.

Acknowledgements

I am grateful to my Supervisor, Carlo R. Contaldi, for offering me this research project opportunity and for the insightful discussions that we had. I would also like to thank my friend Nick for reading my draft report and giving me useful advice.

Abstract

The COVID-19 pandemic has affected human lives on an unprecedented scale, highlighting the necessity for accurate modelling and timely estimation of key parameters of a pandemic, which is crucial to devising scientifically informed mitigating strategies. This report details the development of basic compartmental models used in epidemiological modelling, and investigates the advantages and limitations of more complex compartmental models. A mathematical framework of Bayesian inference is then developed, tailored to the compartmental models used, and bespoke Monte Carlo Markov Chain algorithms are designed to numerically obtain probability distributions for inferred model parameters. The algorithms were tested by inferring parameters from simulated data, yielding results with a mean deviation of 4.62% from the simulated model parameters, describing the model underlying the simulated data with a p-value of $p = 0.9988$. Finally, model parameters were inferred from real COVID-19 data from September 20^{th} 2020 to December 28^{th} 2020 and used to predict the trajectory of the pandemic for a subsequent 25-day period, within a 2σ parameter uncertainty range. Compared against the actual data for that period, the model successfully predicted a turning-point and a decrease in cases.

Contents

Chapter 1

Introduction

The effects of the COVID-19 pandemic have been unprecedented. When the first few cases of the 'novel' coronavirus SARS-CoV-2 were detected in Wuhan, China in early January 2020 [1], no one could have predicted the catastrophic ways in which the unfolding pandemic would shock our world. As of April 13$^{\text{th}}$ 2022, there have been over 6 million COVID-19 related deaths and almost 500 million reported cases worldwide [2]. What's more, the global economy came to an abrupt halt as a result of lock-downs and other mitigating strategies – with major, long-lasting socioeconomic repercussions.

In light of this unparalleled effect on human lives, the necessity for accurate modelling of the pandemic quickly became apparent – in particular the timely estimation of key parameters, crucial to devising scientifically informed strategies to suppress the impact of the pandemic. Many groups of scientists have been developing models to better understand the pandemic and predict its trajectory, including the Medical Research Council (MRC) Centre for Global Infectious Disease Analysis at Imperial College London, which contributes its findings to the UK Scientific Advisory Group for Emergencies (SAGE) [3]. Such research is of paramount importance to decision-making authorities and can be used to manage future pandemics more effectively.

Motivated by this, we seek to build basic epidemiological models of COVID-19 and then apply them to real data to infer parameters of the pandemic, allowing us to make predictions about its evolution. In order to achieve this, we decide to use *Compartmental Modelling*, which has been extensively applied to model infectious diseases and whose effectiveness in doing so is well-established. It is also computationally cheaper compared to other modelling techniques, like agent-based modelling, thereby enabling us to work with larger datasets. We then fit our models to COVID-19 data to infer key parameters, using *Bayesian Inference* – a method naturally suited to this kind of problem, given the pandemic's ever-changing nature (new variants, vaccination, etc.), allowing for inferred parameters to continually update based on new data.

While building basic compartmental models is relatively straight-forward, inference can become quite involved, especially using real COVID-19 data, which is often biased and noisy. Thus, the complexities and assumptions of the models and data will be considered carefully, ensuring clear statistical treatment of uncertainties in the inference process.

In summary, we aim to:

1. Build and understand basic compartmental models of COVID-19.
2. Use our models to infer parameters of simulated data, ensuring our inference method works correctly.
3. Fit our models to real COVID-19 data to infer the parameters of the pandemic, including their uncertainties.
4. Make a short-term prediction about the course of the pandemic.

1.1 Report Layout

This report begins with Chapter 2 on Compartmental Modelling, detailing how basic compartmental models are constructed, programmed and run – and then how these can be extended to more complex models, with their various advantages and disadvantages.

The report continues with Chapter 3 on Bayesian Inference, deriving a relevant mathematical framework used to infer parameters, followed by Chapter 4 on numerical methods, used to obtain parameter probability distributions.

Then, Chapters 5 and 6 show the application of the models and inference methods developed to infer parameters from simulated and real COVID-19 data – the former to test our inference method, and the latter to make a short-term prediction about the progression of the pandemic.

The report ends with a summary of the aims, objectives, and methods developed (Chapter 7), outlining central conclusions and ideas for future research.

Chapter 2

Compartmental Modelling

To model the COVID-19 pandemic, we employed a modelling technique commonly used in epidemiology, whereby a population is divided into labelled groups – *compartments* – hence its name: *Compartmental Modelling*. In essence, we model the flow of people between compartments – whose time-evolution provides a model for the pandemic.

Compartmental models originate from the works of Ronald Ross and Hilda Hudson [4, 5, 6] who were the first to formulate the mathematics of these models, and later, from the works of Kermack and McKendrick [7], who outlined the well-known SIR model. Ross realised over 100 years ago that "behind these [problems of epidemiology] there must be causes which are of profound importance to mankind and which probably can be ascertained only by those principles of careful computation which have yielded such brilliant results in astronomy, physics, and mechanics" [4]. About 10 years after the Spanish Flu broke out, Kermack and McKendrick posed certain important questions, such as "whether the termination [of an infectious disease] occurs only when no susceptible individuals are left, or whether the interplay of the various factors of infectivity, recovery and mortality, may result in termination, whilst susceptible individuals are still present in the unaffected population" [7]. To contextualise these questions, we start by analysing the most basic compartmental model, the *SIR model*.

2.1 SIR Model

To form a compartmental model, we must first decide what compartments to partition a population into – and then how these interact with each other. In Figure 2.1, we see the most basic compartmental model formalised by Kermack and McKendrick in 1927 – the SIR model, wherein the population is partitioned into just 3 compartments:

- $S(t)$**:** Susceptible
- $I(t)$**:** Infected
- $R(t)$**:** Recovered

After deciding what compartments to use, we determine how these interact, expressing rates of flow of people between them – i.e. rates of change of compartments – by writing down a set

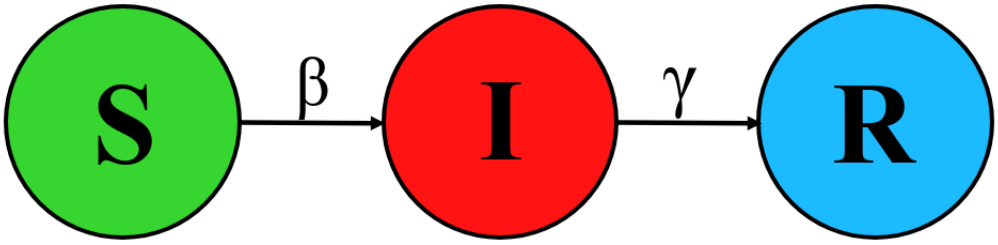

Figure 2.1: A flowchart representing the SIR model, as put forward by Kermack and McKendrick in 1927 [7], consisting of 3 compartments: $S(t)$ - Susceptible, $I(t)$ - Infected, $R(t)$ - Recovered, with 2 model parameters: β and γ, determining the rates of flow between compartments. The arrows depict the direction of flow of people between compartments.

of coupled ordinary differential equations (ODEs) with specific parameters, in this case β and γ, which encode information particular to a given infectious disease and population.

First, we note a few key assumptions of the SIR model – most importantly, a fixed population: the sum of all compartments $N = S(t) + I(t) + R(t)$ is constant, such that:

$$\frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt} = \frac{dN}{dt} = 0. \tag{2.1}$$

This represents a population where birth and death rates are considered negligible and are disregarded. We also assume that:

1. Every susceptible person has the same probability as every other susceptible person of becoming infected.
2. A single infection provides total immunity for the infected individual (no re-infections).
3. The parameters are constant.

Using these assumptions, we create the SIR model, obtaining a set of coupled ODEs that describe rates of flow of people between compartments. We start by considering the rate at which susceptible people $S(t)$ leave their compartment – the rate of infections. Given that $S(t)$ can only change by susceptible people becoming infected, we need:

1. a contact between a susceptible and an infected person,
2. the transmission of disease upon such a contact.

Thus, we need to know how frequently contacts occur between susceptible and infected people, and how likely the transmission of disease is upon such a contact. Letting c be the average number of contacts a person makes per unit time, then any susceptible person makes on average c contacts per unit time, of which $c\frac{I}{N}$ are contacts with an infected person. Thus, susceptible people collectively make $Sc\frac{I}{N}$ contacts with infected people per unit time. If each such contact has probability p of transmission, then there are $pSc\frac{I}{N}$ total infections per unit time i.e. the rate of infections is given by:

$$\frac{d(\text{Infections})}{dt} = pc\frac{SI}{N}. \tag{2.2}$$

Since the rate of change of $S(t)$ is negative the rate of infections, as for every additional infection there is a reduction in the population of susceptible people, then the rate of change of susceptible people is given by:

$$\frac{dS}{dt} = -\frac{d(\text{Infections})}{dt} = -pc\frac{SI}{N} =: -\beta\frac{SI}{N}, \tag{2.3}$$

where $\beta := cp$ is the first parameter of the SIR model. Next, we consider the rate of change between infected and recovered people, "assumed to be proportional to the number of infected individuals" [8],

$$\frac{dR}{dt} = \gamma I, \tag{2.4}$$

where γ is the fraction of infected people that recover per unit time. This second parameter of the SIR model can also be considered as an inverse measure of the length of time for which an infected individual remains ill: $\gamma \propto 1/D$ where D is the expected duration of illness.

Finally, we use Equation 2.1 to get the rate of change of infected people, using the two rates of change derived above (Equations 2.3 and 2.4), to obtain:

$$\frac{dI}{dt} = \beta\frac{IS}{N} - \gamma I. \tag{2.5}$$

These 3 equations form a set of 3 coupled ODEs defining the SIR model, providing a complete description for the flow of people between the 3 compartments, characterised by parameters β and γ, defined above. We note that this makes the SIR model, and all other compartmental models, deterministic; as they are defined by a system of ODEs. These constitute one of two big classes of epidemiological models, the other being *stochastic models* like agent-based ones, which formulate probabilistic models for how disease spreads from one individual to another.

As an aside, we can use parameters β and γ to define a commonly used composite parameter R_0, known as the basic reproduction rate:

$$R_0 := \frac{\beta}{\gamma} \propto cpD, \tag{2.6}$$

where c, p and D are defined above. If we let the 'time unit' considered by c equal the average length of time D for which a person remains infected, then Equation 2.6 simplifies to *the average number of contacts per person, times the probability of transmission upon a contact.* Thus, R_0 gives the expected number of new infections a single infected person causes during their illness, assuming every person they come into contact with is susceptible. This is a metric that incorporates both the behavioural characteristics of a population and the nature of the pathogen itself, hence is very useful for assessing a disease's potential to spread in a population.

2.1.1 Programming & Solving the SIR Model

Having defined the SIR model with the 3 coupled ODEs – Equations 2.3, 2.4, and 2.5 – we want to obtain $S(t)$, $I(t)$, and $R(t)$ explicitly as functions of t to get the time-evolution of each compartment. Together, these represent the model of the pandemic. We notice, however, that this system of coupled ODEs is non-linear, hence analytical solutions do not exist. Despite that, various efforts have been made to develop analytical solutions in the form of *approximants*, involving numerically solved integrals. The most recent such examples include the works of Tiberiu Harko [9], Joe C. Miller [10, 11], as well as Martin Kroger and Reinhard Schlickeiser [12, 13].

Here we present the most commonly employed technique for solving the SIR model: numerical integration of the system of ODEs – namely using Python package SciPy's highly-optimised function *scipy.integrate.solve_ivp* [14]. To do this, we consider this task as an initial-value problem, determining the number of people in each compartment at $t = 0$ (the initial values of each variable), and the model parameter values, which are constant. We then define a method in Python to represent the system of coupled ODEs, and continue by defining a second method that invokes the *scipy.integrate.solve_ivp* routine to solve the SIR model, providing the following 5 arguments:

1. The Python function representing the system of coupled ODEs, i.e. the SIR model.
2. The integration interval – a tuple consisting of the initial and final values of t: (t_0, t_f).
3. The initial values of the compartments: $S(t = 0), I(t = 0), R(t = 0)$.
4. The values of the model parameters β and γ.
5. An array t_{eval} of points in time for which the solutions for all compartments are to be evaluated.

We use the SciPy routine's default Runge-Kutta method of order 5 [15], although SciPy offers other numerical integration methods, as well [14]. This returns a list of arrays, one for each compartment, with values calculated at each time point in t_{eval}.

To run our first SIR model and verify that it works correctly, we must decide what arguments to pass to the SciPy routine. Firstly, we set a population size $N = 10^6$ – a rather arbitrary decision, but necessary nonetheless.

Next, we determine the initial number of infected individuals – set to 0.1% of the population i.e. $I(t = 0) = 1000$, based on the fraction of the English population infected with COVID-19 when the pandemic hit the UK in March of 2020, as suggested by detailed findings from the Coronavirus Infection Survey for England by the Office for National Statistics [16]. Given that we consider 'initial values' of a pandemic model, we can assume that no people have recovered from the disease yet, setting $R(t = 0) = 0$, thus the initial value for susceptible people is: $S(t = 0) = N - I(t = 0) - R(t = 0) = 999000$.

Having set initial values, we then determine model parameter values β and γ. Again, to run the model and verify that it works, the parameter values can be arbitrary; we choose $\beta = 0.5$ and $\gamma = \frac{1}{6}$, giving $R_0 = 3$, consistent with early COVID-19 data [17]. Finally, we decide to

evaluate the values for each compartment at daily intervals for a total of 50 days, yielding the solutions for the SIR model demonstrated in Figure 2.2.

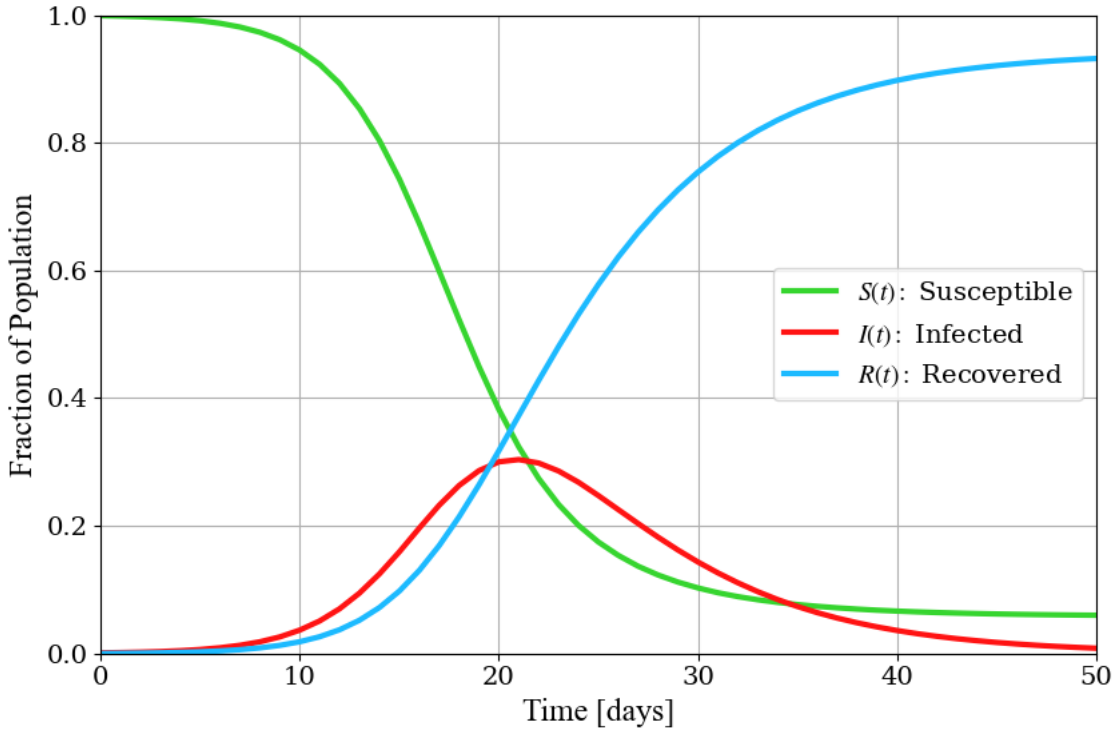

Figure 2.2: Solutions for the SIR model described by Equations 2.3, 2.4, and 2.5, demonstrated as the time-evolution of compartments – $S(t)$ in green, $I(t)$ in red, and $R(t)$ in blue, with: $\beta = 0.5$, $\gamma = \frac{1}{6}$, $S(t = 0) = 999000$, $I(t = 0) = 1000$, and $R(t = 0) = 0$. The values of the compartments are represented as a **fraction** of the population (y-axis), as population size chosen is arbitrary.

To verify that the model ran correctly, we check that the key assumption of the SIR model: constant population size, summarised by Equation 2.1, holds. Indeed, by adding the 3 compartment values at any point in time, we always get $N = 10^6$, confirming that the routine works correctly and that the model runs successfully.

We observe from Figure 2.2 that, initially, the number of susceptible people quickly drops, as they become infected and then recover. The number of infected individuals then peaks, and infected people start decreasing, due to the pool of susceptible people – i.e. people that can become infected – decreasing such that rate of recovery exceeds rate of infection. Finally, a pandemic 'ends' when most of the population is recovered and a small fraction remains susceptible; the ratio can vary, depending on the model parameter values.

While we have successfully built a basic compartmental model, the SIR model, we note that it is relatively simple and does not paint a very accurate and realistic picture of the pandemic. For example, we assume people cannot die from the disease, which is untrue for COVID-19. Thus, we want to explore how we can extend this basic model to capture more of the complexities of a real pandemic, forming more complete models.

2.2 More Complex Compartmental Models

While the SIR model is useful for a general picture, we would like to create models with a more detailed representation of the various complexities in a pandemic, offering more specific information – like how many people may be expected to get hospitalised or die. To construct these, we need to better understand the various dynamics in a pandemic, and then we can better partition the population into more compartments.

For example, early COVID-19 data revealed that people are not infectious from the moment they contract the virus [18]; there is a short interim period called *latency* where individuals are infected but not yet infectious, usually represented with a compartment called **exposed**. In addition, we may also separate infected individuals into **asymptomatic** and **symptomatic**, or consider **hospitalised** and **intubated** people, who are separated from the rest of the community and hence have different infectivity rates. Moreover, we could consider people who die from a disease and place them into a **deceased** compartment, as they do not contribute to epidemiological dynamics, while we could group **vaccinated** people into their own compartment, as their probabilities of infection, hospitalisation, and death are significantly reduced [19].

Considering the additional compartments (in bold, above), we could form a more complex and accurate model of the pandemic, depicted below in Figure 2.3. While this more complex

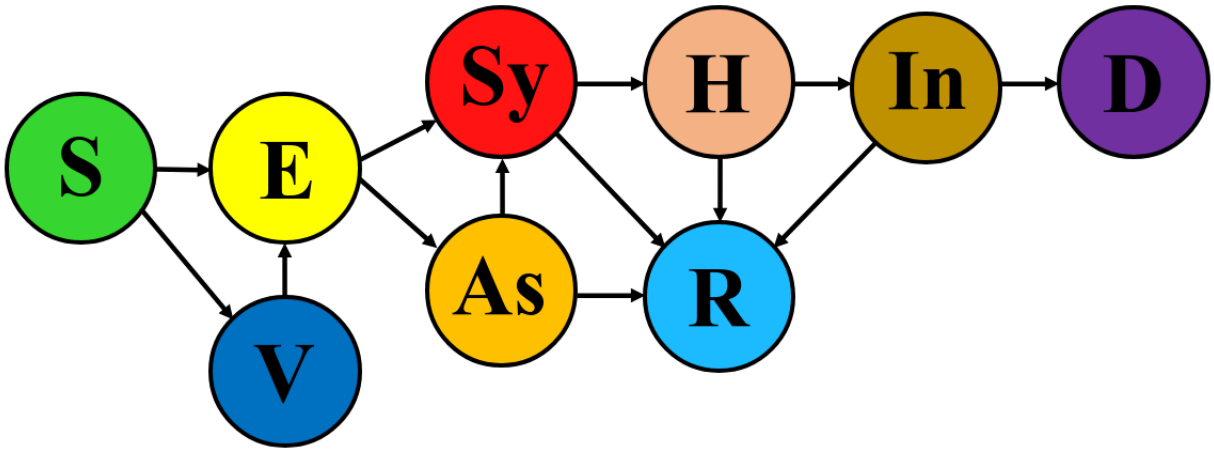

Figure 2.3: A flowchart representing a more complex compartmental model, consisting of 9 compartments: $S(t)$ - Susceptible, $E(t)$ - Exposed, $V(t)$ - Vaccinated, $Sy(t)$ - Symptomatic, $As(t)$ - Asymptomatic, $R(t)$ - Recovered, $H(t)$ - Hospitalised, $In(t)$ - Intubated, $D(t)$ - Deceased. The arrows depict the direction of the flow of people. The model parameters that would be on top of the arrows have been omitted, as their naming convention is rather arbitrary; the key goal of this figure is to depict the structure of an example of a more complex model.

model is indeed more realistic than the SIR model, it is limited by the precision and accuracy of the additional information used to form it. That is, with each compartment added, we inevitably make assumptions on its interactions with the rest of the model, and on the values of new model parameters introduced. If we are very certain about their accuracy, then such complex models will be better representations of the pandemic. If, however, there is some level of uncertainty in them, it propagates through multiple compartments – resulting in potentially very uncertain and inaccurate predictions. In fact, detailed analysis of the propagation of errors in various models has shown that "simpler [compartmental] models are less sensitive

to uncertain parameters than more complex models" [20] as modelling a greater number of more specific aspects of a pandemic is more prone to errors than modelling fewer, more general aspects. This effect can at times be so significant that the SIR model proves more accurate in predicting real COVID-19 data than more complex models [21].

As a result, when choosing a model, especially for model-parameter inference and predictions, we want to optimise the trade-off between increased detail and increased prediction uncertainty of more complex models.

2.2.1 Compartmental Model Used for Inference & Prediction

Considering the detail-uncertainty trade-off of complex compartmental models, we decided to build a model to infer parameters from COVID-19 data and make predictions which, despite being very simple, with only 4 parameters, captures the most important aspects of a pandemic. Our model, the SIRHD model, consists of the following 5 compartments:

- $S(t)$**:** Susceptible
- $I(t)$**:** Infected
- $H(t)$**:** Hospitalised
- $R(t)$**:** Recovered
- $D(t)$**:** Deceased

demonstrated in Figure 2.4. We want to capture the key aspects of the pandemic, including:

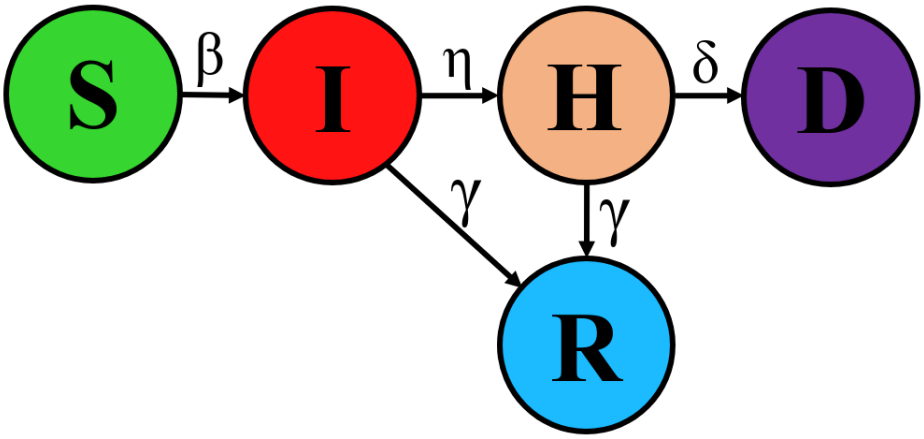

Figure 2.4: A flowchart representing the SIRHD model, consisting of 5 compartments: $S(t)$ - Susceptible, $I(t)$ - Infected, $H(t)$ - Hospitalised, $R(t)$ - Recovered, $D(t)$ - Deceased, with 4 model parameters: β, η, γ, and δ, determining the rates of flow between compartments. The arrows depict the direction of the flow of people.

the total number of infected people – a measure of the prevalence of the virus; the number of

hospitalised people – which provides an indication of the pressure on the healthcare system; and finally the number of deceased people, which in turn is a measure of the severity of a pandemic.

As for the SIR model, we start with the key assumption of population conservation, which, in the case of the SIRHD model we are now using, also includes the deceased compartment:

$$\frac{dS}{dt} + \frac{dI}{dt} + \frac{dH}{dt} + \frac{dR}{dt} + \frac{dD}{dt} = \frac{dN}{dt} = 0. \tag{2.7}$$

Using the same arguments used for the SIR model to derive the system of coupled ODEs, we get the same rate of change for susceptible people:

$$\frac{dS}{dt} = -\beta \frac{SI}{N_d}, \tag{2.8}$$

where β is defined in Equation 2.3 and N_d represents the *dynamic* population, i.e. excluding the deceased compartment. Similarly, we get the same interaction between the infected and recovered compartments – namely, infected people are assumed to recover at some constant rate γ. However, in this case, we get an additional interaction term arising from infected people getting hospitalised at some constant rate η, resulting in:

$$\frac{dI}{dt} = \beta \frac{SI}{N_d} - \gamma I - \eta I. \tag{2.9}$$

The rate of change of hospitalised people is obtained by considering, per unit time, the number of infected people who get hospitalised ηI, the number of hospitalised people who die, and the number of hospitalised people who recover, resulting in:

$$\frac{dH}{dt} = \eta I - \gamma H - \delta H, \tag{2.10}$$

where δ is the fraction of hospitalised people who die per unit time. Finally, the rates of change of recovered and deceased people are obtained simply by considering the population conservation summarised by Equation 2.7, yielding:

$$\frac{dR}{dt} = \gamma I + \gamma H, \tag{2.11}$$

and

$$\frac{dD}{dt} = \delta H. \tag{2.12}$$

The set of Equations 2.8 to 2.12 forms a system of 5 coupled ODEs that define the SIRHD model, characterised by parameters β, γ, η, and δ defined above. This model is programmed and run with the same steps analysed in Section 2.1.1 – in this case providing the initial values of 5 compartments and the values of 4 model parameters. Moreover, we verified that the SIRHD model works correctly similarly to how we did for the SIR model, by running the model to check that the population size N is conserved. As before, we set a population size of $N = 10^6$, assuming 0.1% of the population is initially infected, i.e. $I(t = 0) = 1000$, and assume that at the 'beginning' of a pandemic there are no hospitalised, recovered, and deceased people, setting $H(t = 0) = 0$, $R(t = 0) = 0$, and $D(t = 0) = 0$, and hence $S(t = 0) = 999000$. Parameters β and γ were set to $\beta = 0.5$ and $\gamma = \frac{1}{6}$ as before, while the additional parameters were set to

$\eta = \frac{1}{7} \times 0.024 \approx 0.003429$, and $\delta = \frac{1}{10} \times 0.25 = 0.025$, assuming that: an average of 2.4% of infected people get hospitalised, and that this happens an average of 7 days after infection; and that an average of 25% of hospitalised people die, and this occurs an average of 10 days after being hospitalised, consistent with early COVID-19 data [22, 23]. We calculated the values for each compartment at daily intervals for a total of 50 days, producing the solutions for the SIRHD model demonstrated in Figure 2.5. Adding together the 5 compartments for any point in time always gives $N = 10^6$, confirming that the model runs successfully, with N being conserved.

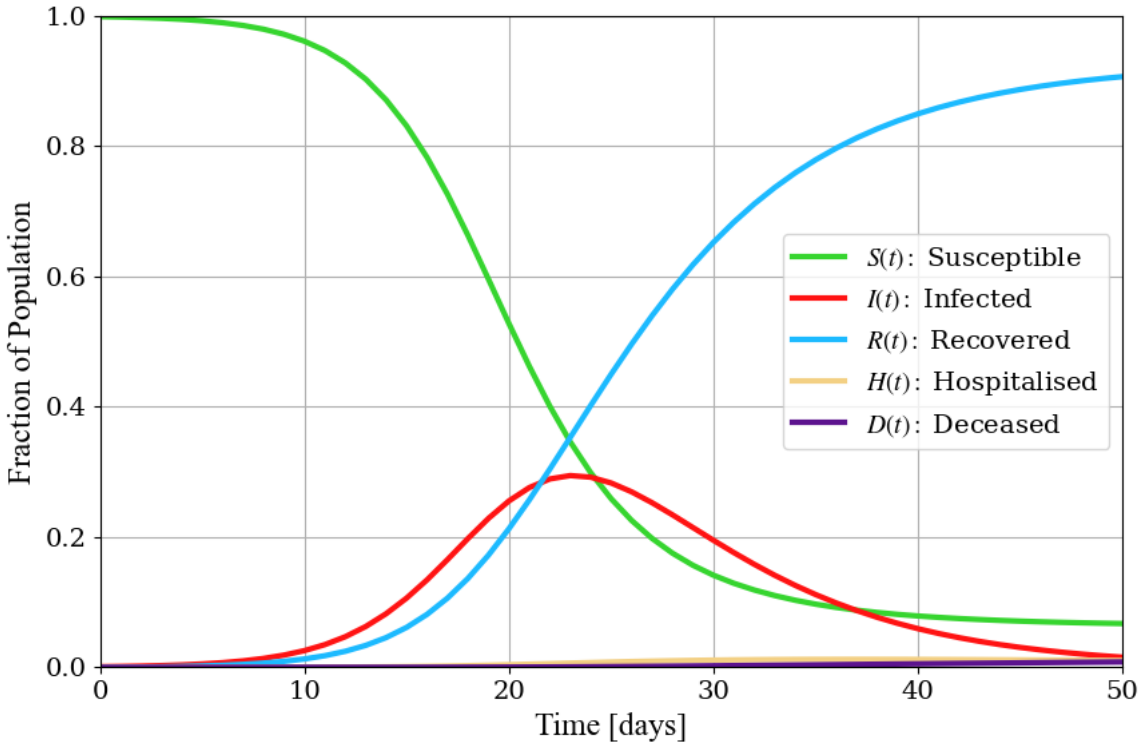

Figure 2.5: Solutions for the SIRHD model described by Equations 2.8 to 2.12, demonstrated by the time-evolution of compartments – $S(t)$ in green, $I(t)$ in red, $R(t)$ in blue, $H(t)$ in beige, and $D(t)$ in purple, with: $\beta = 0.5$, $\gamma = \frac{1}{6}$, $\eta = 0.00342865$, $\delta = 0.025$, $S(t = 0) = 999000$, $I(t = 0) = 1000$, $R(t = 0) = H(t = 0) = D(t = 0) = 0$. The values of the compartments are represented as a fraction of the population (y-axis) as population size chosen is arbitrary.

So far, we have built basic compartmental models and explored various advantages and disadvantages of more complex compartmental models. We have also selected and formulated a specific compartmental model – the SIRHD model – which captures the key aspects of a pandemic yet remains simple with very few model parameters, thus requiring few assumptions to be made. We can therefore feel relatively confident in using this model to infer parameters from COVID-19 data. The next section explores the relevant mathematical framework of *Bayesian Inference* and how it is used for this process.

Chapter 3

Bayesian Inference

While models are a great tool for representing a pandemic, they have little practical utility unless they use model parameters that reflect the true, real-world dynamics of a pandemic, which would allow us to make accurate predictions and devise mitigating strategies. Using data to learn what the 'true' model parameters are is called *inference* and we aim to use a specific kind, known as *Bayesian inference*, to obtain probability distributions for our model parameters, which offers us not only 'best estimates' for these parameters, but a measure of our uncertainty in them as well. This is key in developing mitigating strategies for a range of possible trajectories of the pandemic and assessing how likely each is.

Bayesian inference builds on the principle of 'updating' the probability distribution for an old hypothesis with new information (data) to obtain a new probability distribution. This is particularly useful in the case of *sequential analysis* where data is not complete and is assessed as it is obtained, which naturally suits the COVID-19 pandemic given its ever-changing nature and continually updated data. While the derivations of the relevant mathematical framework in this section are lengthy, they detail various assumptions made to allow for simplifications and have been tailored to the specific problems at hand, so it is important to analyse them explicitly.

More specifically, Bayesian inference uses Bayes' equation [24]:

$$\mathbb{P}(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{\mathbb{P}(\boldsymbol{D}|\boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta})}{\mathbb{P}(\boldsymbol{D})}, \tag{3.1}$$

which defines the probability of model parameters $\boldsymbol{\theta}$ given the data $\boldsymbol{D}$ we observe, known as the posterior probability $\mathbb{P}(\boldsymbol{\theta}|\boldsymbol{D})$, as the product of: the likelihood $\mathbb{P}(\boldsymbol{D}|\boldsymbol{\theta})$, which indicates the compatibility of the data with the model parameters considered; and the prior probability of the model parameters $\mathbb{P}(\boldsymbol{\theta})$. All this is normalised by the probability $\mathbb{P}(\boldsymbol{D})$ of the data which is constant for all model parameters considered but has an implicit condition on the model choice; a different model would result in a different data probability. Since we are not interested in how different models perform but are always using the SIRHD model, we can simplify Equation 3.1 to:

$$\mathbb{P}(\boldsymbol{\theta}|\boldsymbol{D}) \propto \mathbb{P}(\boldsymbol{D}|\boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta}). \tag{3.2}$$

We note that the proportionality arises from the fact that removing the normalisation constant means the values of the right-hand side are not probabilities anymore. However, this does not affect our distribution as relative proportional differences of the posterior probability for

different parameters remain the same. Moreover, since we are considering probability **distributions** rather than probabilities, it is convention to write f in place of $\mathbb{P}$, where f represents a probability density function.

In order to employ Equation 3.2 with actual data, we consider that the kind of data we typically have, such as the number of people who are in hospital on any given day, is related in some way to each compartment in a model. To simplify the problem, we consider data for each compartment to be independent from other compartments, allowing us to write:

$$f(\boldsymbol{\theta}|\boldsymbol{D}) = \prod_{c=1}^{C} f(\boldsymbol{\theta}|\boldsymbol{D}^c) \propto \prod_{c=1}^{C} f(\boldsymbol{D}^c|\boldsymbol{\theta}) f(\boldsymbol{\theta}), \tag{3.3}$$

with data $\boldsymbol{D}^c$ being from compartment c, for a model with C compartments. Next, we break down the likelihood for each compartment's data $f(\boldsymbol{D}^c|\boldsymbol{\theta})$, typically written as $L(\boldsymbol{\theta}|\boldsymbol{D}^c) \propto f(\boldsymbol{D}^c|\boldsymbol{\theta})$. To achieve this, we assume that each data-point is an independent and identically-distributed (i.i.d.) Gaussian variate with mean and standard deviation provided by the model with parameters $\boldsymbol{\theta}$:

$$L(\boldsymbol{\theta}|\boldsymbol{D}^c) \propto f(\boldsymbol{D}^c|\boldsymbol{\theta}) = \prod_{i=1}^{N_c} \frac{1}{\sqrt{2\pi}\sigma_i^c} \exp\left(-\frac{1}{2}\left(\frac{y_i^c - \mu_i^c(\boldsymbol{\theta})}{\sigma_i^c}\right)^2\right), \tag{3.4}$$

for N_c univariate Gaussian data-points $y_i^c \sim \mathcal{N}(\mu_i^c(\boldsymbol{\theta}), \sigma_i^{c2})$ with mean $\mu_i^c(\boldsymbol{\theta})$ and standard deviation σ_i^c, where $i \in \{1, \ldots, N_c\}$, all for compartment c. In order to get rid of the exponential and the products in Equation 3.3 we take logarithms:

$$\begin{aligned} \ln f(\boldsymbol{\theta}|\boldsymbol{D}) \propto \ln\left(\prod_{c=1}^{C} f(\boldsymbol{D}^c|\boldsymbol{\theta}) f(\boldsymbol{\theta})\right) &= \sum_{c=1}^{C} \ln\left(f(\boldsymbol{D}^c|\boldsymbol{\theta}) f(\boldsymbol{\theta})\right) \\ &= \sum_{c=1}^{C} \ln f(\boldsymbol{D}^c|\boldsymbol{\theta}) + \sum_{c=1}^{C} \ln f(\boldsymbol{\theta}). \end{aligned} \tag{3.5}$$

Substituting Equation 3.4 into Equation 3.5 we get:

$$\begin{aligned} \ln f(\boldsymbol{\theta}|\boldsymbol{D}) &\propto \sum_{c=1}^{C} \ln\left[\prod_{i=1}^{N_c} \frac{1}{\sqrt{2\pi}\sigma_i^c} \exp\left(-\frac{1}{2}\left(\frac{y_i^c - \mu_i^c(\boldsymbol{\theta})}{\sigma_i^c}\right)^2\right)\right] + \sum_{c=1}^{C} \ln f(\boldsymbol{\theta}) \\ &= \sum_{c=1}^{C} \sum_{i=1}^{N_c} \ln\left[\frac{1}{\sqrt{2\pi}\sigma_i^c} \exp\left(-\frac{1}{2}\left(\frac{y_i^c - \mu_i^c(\boldsymbol{\theta})}{\sigma_i^c}\right)^2\right)\right] + \sum_{c=1}^{C} \ln f(\boldsymbol{\theta}). \end{aligned} \tag{3.6}$$

Using the product rule for logarithms for the terms in the square brackets, we get:

$$\begin{aligned} \ln f(\boldsymbol{\theta}|\boldsymbol{D}) &\propto \sum_{c=1}^{C} \sum_{i=1}^{N_c} \left(-\frac{1}{2}\ln 2\pi - \ln \sigma_i^c - \frac{1}{2}\left(\frac{y_i^c - \mu_i^c(\boldsymbol{\theta})}{\sigma_i^c}\right)^2\right) + \sum_{c=1}^{C} \ln f(\boldsymbol{\theta}) \\ &= \sum_{c=1}^{C} \left(-\frac{N_c}{2}\ln 2\pi - \sum_{i=1}^{N_c} \ln \sigma_i^c - \frac{1}{2}\sum_{i=1}^{N_c}\left(\frac{y_i^c - \mu_i^c(\boldsymbol{\theta})}{\sigma_i^c}\right)^2\right) + \sum_{c=1}^{C} \ln f(\boldsymbol{\theta}) \\ &= \sum_{c=1}^{C} \left(-\frac{1}{2}\sum_{i=1}^{N_c}\left(\frac{y_i^c - \mu_i^c(\boldsymbol{\theta})}{\sigma_i^c}\right)^2 + K_c\right) + \sum_{c=1}^{C} \ln f(\boldsymbol{\theta}), \end{aligned} \tag{3.7}$$

where $K_c = -\frac{N_c}{2}\ln 2\pi - \sum_{i=1}^{N_c}\ln\sigma_i^c$. We notice that the sum over all data-points in the 3rd row of Equation 3.7 is in fact the chi-squared statistic for each compartment's data $\chi^2_{N_c}$, giving the mean squared weighted deviations between the observed and the modelled data. This is a crucial result, stemming from the assumptions of independent and identically distributed Gaussian variate data, allowing us to relate the logarithm of the posterior probability distribution to a very easily measurable quantity. Therefore, we can re-write Equation 3.7 as:

$$\ln f(\boldsymbol{\theta}|\boldsymbol{D}) \propto -\frac{1}{2}\sum_{c=1}^{C}\chi^2_{N_c} + K_1 + \sum_{c=1}^{C}\ln f(\boldsymbol{\theta}), \tag{3.8}$$

where $\chi^2_{N_c} = \sum_{i=1}^{N_c}\left(\frac{y_i^c - \mu_i^c(\boldsymbol{\theta})}{\sigma_i^c}\right)^2$ and $K_1 = \sum_{c=1}^{C} K_c$ is a constant.

Finally, the prior probability distribution $f(\boldsymbol{\theta})$ is simply the distribution of the parameters before we observe any data (and subsequently update our distribution), and may arise from independent research. If such research exists we may be able to use specific probability distributions for each parameter. However, in a real pandemic there is rarely enough time to conduct such research and it is thus safer to assume as little about these probability distributions as possible. For that reason, we either assume no prior knowledge about the parameters and hence completely disregard the priors, or, if we have some indication about the values of the parameters, we assume independent Gaussian distributions for each, yielding:

$$\begin{aligned}
\ln f(\boldsymbol{\theta}|\boldsymbol{D}) &\propto -\frac{1}{2}\sum_{c=1}^{C}\chi^2_{N_c} + K_1 + \sum_{c=1}^{C}\ln\left(\prod_{p=1}^{P}\frac{1}{\sqrt{2\pi}\sigma_p}\exp\left(-\frac{1}{2}\left(\frac{\theta_p-\mu_p}{\sigma_p}\right)^2\right)\right)\\
&= -\frac{1}{2}\sum_{c=1}^{C}\chi^2_{N_c} + K_1 + \sum_{c=1}^{C}\sum_{p=1}^{P}\ln\left(\frac{1}{\sqrt{2\pi}\sigma_p}\exp\left(-\frac{1}{2}\left(\frac{\theta_p-\mu_p}{\sigma_p}\right)^2\right)\right)\\
&= -\frac{1}{2}\sum_{c=1}^{C}\chi^2_{N_c} + K_1 + C\sum_{p=1}^{P}\left(-\frac{1}{2}\ln 2\pi - \ln\sigma_p - \frac{1}{2}\left(\frac{\theta_p-\mu_p}{\sigma_p}\right)^2\right)\\
&= -\frac{1}{2}\sum_{c=1}^{C}\chi^2_{N_c} + K_1 + CK_2 - \frac{C}{2}\chi^2_P\\
&= -\frac{1}{2}\sum_{c=1}^{C}\chi^2_{N_c} - \frac{C}{2}\chi^2_P + \text{const.},
\end{aligned} \tag{3.9}$$

for P model parameters $\theta_p \in \boldsymbol{\theta}$, $\theta_p \sim \mathcal{N}(\mu_p, \sigma_p^2)$ with mean μ_p and standard deviation σ_p, where: $p \in \{1, \ldots, P\}$, $\chi^2_P = \sum_{p=1}^{P}\left(\frac{\theta_p-\mu_p}{\sigma_p}\right)^2$, and $K_2 = -\frac{P}{2}\ln 2\pi - \sum_{p=1}^{P}\ln\sigma_p$ is a constant.

Now that we have a compact way of representing the log of the posterior probability distribution – using the chi-squared statistics of the compartment data and the parameter priors – we could exponentiate both sides, to get a functional form of the posterior probability distribution in terms of $\boldsymbol{\theta}$. However, although that is, in principle, possible, it would not provide a closed-form solution of the posterior as a function of the parameters, as the likelihood is non-analytic in $\boldsymbol{\theta}$, as shown in Equation 3.4. As a result, we need a numerical method to sample the posterior probability distribution, to obtain an approximation of it. The development of a bespoke sampling method for this purpose is covered in Chapter 4. The best-estimate for the

vector of parameter values $\boldsymbol{\theta}^*$ is the one that maximises the posterior probability distribution – and hence the log of it – given by:

$$\begin{aligned}\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}}\ \ln f(\boldsymbol{\theta}|\boldsymbol{D}) &= \underset{\boldsymbol{\theta}}{\text{argmax}} \left(-\frac{1}{2}\sum_{c=1}^{C} \chi^2_{N_c} - \frac{C}{2}\chi^2_P + \text{const.} \right) \\ &= \underset{\boldsymbol{\theta}}{\text{argmin}} \left(\frac{1}{2}\sum_{c=1}^{C} \chi^2_{N_c} + \frac{C}{2}\chi^2_P \right), \end{aligned} \tag{3.10}$$

where removing the negative signs recasts this a minimisation problem, and the constant term can be disregarded as it is irrelevant to the optimisation.

Finally, it is important to note that the posterior probability distribution considered so far is a multivariate distribution – i.e. a distribution in all model parameters $\boldsymbol{\theta}$. However, to extract the uncertainty of a single parameter θ_p, we need its univariate probability distribution. For any individual parameter θ_p, we obtain this by calculating its *marginal* probability distribution, integrating out all other parameters [25]:

$$f(\theta_p|\boldsymbol{D}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f(\boldsymbol{\theta}|\boldsymbol{D}) d\theta_1 d\theta_2 \ldots d\theta_{p-1} d\theta_{p+1} \ldots d\theta_P, \tag{3.11}$$

where $\theta_p \in \boldsymbol{\theta}$, and $p \in \{1, \ldots, P\}$ for a model with P parameters. Given that there is no closed-form solution for $f(\boldsymbol{\theta}|\boldsymbol{D})$ and hence it is sampled, we consider the discrete case of Equation 3.11 which is given by:

$$f(\theta_p|\boldsymbol{D}) = \sum_{\theta_1}\sum_{\theta_2}\cdots\sum_{\theta_{p-1}}\sum_{\theta_{p+1}}\cdots\sum_{\theta_P} f(\boldsymbol{\theta}|\boldsymbol{D}). \tag{3.12}$$

At this point, we have derived a (non-analytic) expression for the posterior probability distribution in terms of the mean squared weighted deviations between the observed and the modelled data for some set of parameters $\boldsymbol{\theta}$, shown in Equation 3.9. We have also derived an (again, non-analytic) expression for the marginal posterior probability distribution for each parameter, and the best-estimate parameters, given by Equations 3.12 and 3.10, respectively. In Chapter 4 we develop numerical methods to approximate the posterior probability distribution, from which we can obtain best-estimate parameter values, and also approximate the marginal posterior probability distributions.

Chapter 4

Sampling Algorithms

The posterior probability distributions obtained using Bayesian inference are intractable for all but the most trivial probabilistic models – that is, analytical solutions do not exist. Therefore, most often, numerical methods are used to approximate them. One of the most common such methods is Markov Chain Monte Carlo (MCMC), which constructs Markov Chains through Monte Carlo experiments.

Monte Carlo experiments are a type of algorithmic method that uses repeated random sampling in order to obtain numerical solutions. This method was first invented by Jonn von Neumann and Stanislaw Ulam to run simulations necessary for the Manhattan project [26]. The Monte Carlo method can be used to 'explore' the posterior probability distribution numerically. However, it suffers from the *curse of dimensionality*; as the number of parameters increases, the parameter-space volume of the posterior probability distribution increases exponentially, hence exploring it with independent random sampling becomes very slow.

To eliminate this problem it is often combined with a *Markov Chain* – a model that describes a sequence of events known as a **chain** $(\mathcal{Y}_t)_{t\geq 0}$ of $\boldsymbol{\theta}$, taken from a state-space $\mathcal{S}$, where the probability of an event depends solely on the state of the previous event. This is the defining property of Markov Chains – known as the *Markov Property* (or sometimes *memorylessness*) summarised by [27]:

$$\mathbb{P}(\mathcal{Y}_{t+1} = \boldsymbol{\theta}_{t+1} | \mathcal{Y}_t = \boldsymbol{\theta}_t, \ldots, \mathcal{Y}_0 = \boldsymbol{\theta}_0) = \mathbb{P}(\mathcal{Y}_{t+1} = \boldsymbol{\theta}_{t+1} | \mathcal{Y}_t = \boldsymbol{\theta}_t). \tag{4.1}$$

Combining these two methods allows us to randomly sample the posterior probability distribution – but rather than drawing each sample independently, it is drawn solely based on the previous sample. This allows us to close in on areas of interest within the posterior probability distribution being approximated and draw most of the samples from those areas. The detailed workings of and steps taken to develop a bespoke MCMC algorithm are summarised in Section 4.1.

4.1 Markov Chain Monte Carlo (MCMC)

The Markov Chain Monte Carlo (MCMC) method works by creating a sequence of points in such a way that the distribution of these points tends towards the distribution they are sampled from – in our case, the posterior probability distribution – as more and more points are sampled. It begins by drawing a sample from a uniform probability distribution:

$$\boldsymbol{\theta}_0 \sim \mathcal{U}(\boldsymbol{\theta}_{\min}, \boldsymbol{\theta}_{\max}), \tag{4.2}$$

whose bounds $\boldsymbol{\theta}_{\min}$ and $\boldsymbol{\theta}_{\max}$ are determined using the parameter priors, such that it is extremely unlikely that the true parameters lie near these bounds. Each subsequent sample is drawn iteratively. At each iteration, the algorithm considers a candidate sample for the next value in the chain, which is drawn from a *proposal distribution* – in this case Gaussian – centred at the current sample:

$$\boldsymbol{\theta}_{t+1} \sim \mathcal{N}(\mu = \boldsymbol{\theta}_t, \sigma^2 = \alpha\boldsymbol{\theta}_t) \quad t \geq 0, \tag{4.3}$$

This satisfies the key property of a Markov process: the next sample $\boldsymbol{\theta}_{t+1}$ is dependent only on the current sample $\boldsymbol{\theta}_t$, where $\alpha \in (0, 1)$. The new sample is either accepted or rejected using an *acceptance function*, defined as:

$$A(\boldsymbol{\theta}_{t+1}, \boldsymbol{\theta}_t) = \min\left(1, \exp\left(-(\chi_{t+1} - \chi_t)/T\right)\right), \tag{4.4}$$

where $T \in \mathbb{R}_{\geq 0}$ is a parameter of the algorithm known as *temperature*, and $\chi_t \coloneqq \frac{1}{2}\sum_{c=1}^{C} \chi^2_{N_c} + \frac{C}{2}\chi^2_P$, for $\boldsymbol{\theta}_t$, defined in Equation 3.10, represents the negative logarithm of the posterior probability distribution. Finally, what we do is draw a random variable from a uniform distribution $u \in [0, 1]$.

1. If $u \leq A(\boldsymbol{\theta}_{t+1}, \boldsymbol{\theta}_t)$, then $\boldsymbol{\theta}_{t+1}$ is accepted.
2. If $u > A(\boldsymbol{\theta}_{t+1}, \boldsymbol{\theta}_t)$, then $\boldsymbol{\theta}_{t+1}$ is rejected.

If the new sample is accepted, it is added to the chain and the process repeats based on this new sample. If it is rejected, the process repeats using the current sample again. The key aspect of this algorithm is the choice of acceptance function. With detailed proofs, it can be shown that using the kind of acceptance function shown in Equation 4.4, the algorithm produces a chain $(\mathcal{Y}_t)_{t\geq 0}$ that approximates the posterior probability distribution [28]. An intuitive explanation for why this occurs is that if a candidate sample is in a higher probability density region of the posterior probability distribution than the current sample, then $A(\boldsymbol{\theta}_{t+1}, \boldsymbol{\theta}_t) = 1 \geq u$, hence the candidate sample is always accepted. If, on the other hand, the candidate sample is in a lower probability density region than the current sample, it will sometimes be accepted and sometimes rejected; while the greater the disparity in probability density, the less likely the candidate sample is to be accepted. As a result, most often points are sampled from regions of high probability density, and, occasionally, from regions of lower probability density. This method of sampling generates a chain that approximates the posterior probability distribution.

4.1.1 Algorithm Implementation Details

The *temperature* T in the acceptance function in Equation 4.4 is a hyper-parameter commonly used to control how likely the algorithm is to accept candidate samples in lower probability density regions (and, by extension, to control the algorithm's speed); as $T \to 0$ we accept fewer and fewer lower probability density points (eventually performing gradient ascent), while when $T \gg 1$ we accept almost all candidate samples. A technique commonly employed is *simulated annealing*, where T is progressively decreased across iterations, so that we explore a large volume of the parameter space, by accepting more candidate samples, before focusing on any specific region of the posterior probability distribution, thereby avoiding getting stuck in local maxima. However, optimising T for a specific sampling problem is a very complex field currently under research and hence beyond the scope of this report. Instead, we simply proceed with $T = 1$, which is tantamount to not having introduced T to begin with and avoids making any assumptions about its optimal value.

Furthermore, we note that the MCMC algorithm is especially sensitive to the first sample. The algorithm needs to iterate through a few initial samples before the chain can arrive at a region of high probability density in the parameter space and begin sampling around it. Therefore, it is customary to discard a fraction of the chain – typically the first 5% to 10% of the samples – so that the posterior probability distribution approximation is not affected by the initial randomness of the chain, which introduces several points in an arbitrary region around the first sample, artificially increasing the probability density therein and thus skewing the approximation. This process is known as *burn-in* [28].

Taking into consideration the above, the complete MCMC algorithm is summarised below in Algorithm 1, which returns the generated chain $(\mathcal{Y}_t)_{t\geq 0}$ that approximates the posterior probability distribution.

Algorithm 1: Markov Chain Monte Carlo (MCMC)

$\boldsymbol{\theta}_0 \leftarrow \mathcal{U}(\boldsymbol{\theta}_{\min}, \boldsymbol{\theta}_{\max})$;
$\chi_0 \leftarrow \frac{1}{2}\sum_{c=1}^{C} \chi^2_{N_c} + \frac{C}{2}\chi^2_P, \quad$ for $\boldsymbol{\theta}_0$;
$\mathcal{Y}_0 \leftarrow \boldsymbol{\theta}_0$;
$t \leftarrow 0$;
while *True* **do**
  $\boldsymbol{\theta}_{t+1} \leftarrow \mathcal{N}(\mu = \boldsymbol{\theta}_t, \sigma^2 = \alpha\boldsymbol{\theta}_t)$; /* $\alpha \in (0,1)$ */
  $\chi_{t+1} \leftarrow \frac{1}{2}\sum_{c=1}^{C} \chi^2_{N_c} + \frac{C}{2}\chi^2_P, \quad$ for $\boldsymbol{\theta}_{t+1}$;
  $A(\boldsymbol{\theta}_{t+1}, \boldsymbol{\theta}_t) \leftarrow \min\left(1, \exp\left(-(\chi_{t+1} - \chi_t)/T\right)\right)$; /* $T \sim 1$ */
  $u \leftarrow \mathcal{U}(0,1)$;
  if $u \leq A(\boldsymbol{\theta}_{t+1}, \boldsymbol{\theta}_t)$ **then**
    $\mathcal{Y}_{t+1} \leftarrow \boldsymbol{\theta}_{t+1}$;
  else
    $\mathcal{Y}_{t+1} \leftarrow \boldsymbol{\theta}_t$;
  end
  $t \leftarrow t + 1$;
end
return $(\mathcal{Y}_t)_{t\geq 0}$

As mentioned above, we discard 5% to 10% of initial chain samples, to minimise the effect of the initial sample's locality. An additional method employed to make the MCMC algorithm even more robust by further reducing its sensitivity to the starting point, is to initiate multiple chains and run them in parallel. Each chain is initialised with a randomly selected sample that is sufficiently distant (in parameter space) from the initial sample of all other chains. Multiple chains running in parallel – totally independent from each other – map a larger volume of the parameter space. Once all chains have been generated independently, they are aggregated together into one chain, which is a better approximation of the posterior probability distribution than any of the individual chains. In a single chain, as each sample is conditioned on the previous one, there is a level of correlation, referred to as *auto-correlation*. By allowing multiple chains to run in parallel and then aggregating them into a single chain, the effect of the auto-correlations from each chain is reduced, increasing the effective number of samples and providing a more accurate approximation of the posterior probability distribution [29].

An aspect of the MCMC algorithm not yet discussed is its termination. In Algorithm 1, the iterative process continues indefinitely as the loop-condition always remains True, approximating the posterior probability distribution more and more closely. However, in practice, we want to terminate the algorithm to obtain the chain of samples – which begs the question of when is the right time to terminate the process? The answer is: when we have a chain with a 'stable' set of samples – representing a 'stationary' probability distribution. In order to determine when this has occurred, we use a convergence criterion put forward by Gelman and Rubin [30] that seeks to minimise both the variance of the model parameters within each chain and between chains. The between-chain and within-chain variances are defined as follows.

For a given model parameter $\theta_p \in \boldsymbol{\theta}$ let $\left(\theta_p^m\right)_{t=1}^N$ be the m'th chain with N samples, where $m = 1, \ldots, M$. We then consider $\hat{\theta}_p^m$ to be the sample mean, and $\hat{\sigma}_p^{m2}$ to be the sample variance of the m'th chain for parameter θ_p. We then consider the sample mean from all chains to be $\hat{\theta_p} = \frac{1}{M}\sum_{m=1}^M \hat{\theta}_p^m$. Then, the between-chain variance is defined as:

$$B = \frac{N}{M-1}\sum_{m=1}^{M}\left(\hat{\theta}_p^m - \hat{\theta_p}\right)^2, \tag{4.5}$$

and the within-chain variance as:

$$W = \frac{1}{M}\sum_{m=1}^{M}\hat{\sigma}_p^{m2}. \tag{4.6}$$

Using these two, Gelman and Rubin defined the *pooled variance* as:

$$\hat{V} \coloneqq \frac{N-1}{N}W + \frac{M+1}{MN}B. \tag{4.7}$$

Finally, the Gelman-Rubin convergence criterion is defined as [30]:

$$R_c \coloneqq \frac{\hat{V}}{W}. \tag{4.8}$$

In essence, this criterion assesses whether an ensemble of chains running in parallel, initialised with different samples, converge to the same distribution. For a finite number of samples, the pooled variance $\hat{V}$ overestimates the variance of the posterior probability distribution, while

the within-chain variance W underestimates it [31]. Therefore, as the number of points N sampled increases – and $B \to 0$ – R_c gets closer to 1, demonstrating that the means of parallel chains get closer and closer in value. In the book 'Bayesian Data Analysis', Gelman, Rubin, and others suggest that with $R_c \leq 1.1$, the chains have converged sufficiently [32]. Therefore, we use $R_c = 1.05$ to determine that our chains have converged and terminate the algorithm. It is important to note that in order to calculate this convergence criterion we need a sufficiently large number of chains M, so that the between-chain variance can be meaningful. We therefore decide to use a minimum of 6 chains when running the MCMC algorithm. In effect, Algorithm 1 is run in parallel for M chains, and the Gelman-Rubin criterion is calculated at each iteration; once it drops below 1.05, the algorithm terminates.

Using this parallel-chain MCMC algorithm, we are now able to sample the posterior probability distribution numerically, until all chains have sufficiently converged to the desired distribution – the posterior probability distribution. The final step is to obtain the marginal posterior probability distributions for each model parameter, so that we have our final inference results: the inferred mean and variance of each model parameter. To achieve this, we use an open-source Python package called *GetDist*, which utilises *Kernel Density Estimation* (KDE) [33]. KDE is a "wide class of non-parametric methods of estimating probability densities from samples, improving on histograms by making some weak assumptions about smoothness" [33]. In essence, this method splits the domain of a given model parameter into a number of bins, and creates a histogram by adding the number of samples in each bin over all other parameters i.e. dimensions of the parameter space. Then, specific KDE methods are used to 'smooth' the histograms, offering marginal posterior probability distributions. This package can also produce 2D marginal probability distributions, which create bins in the space of 2 model parameters, making a 2D histogram again by adding the number of samples over all remaining model parameters. Such 2D marginal distributions enable us to see how any two model parameters correlate with one another. With regards to our specific sampling problem, this offers great insight into the model used for the pandemic, highlighting where potential degeneracy of model parameters – which could significantly slow down the sampling of the posterior probability distribution – can be corrected. We therefore develop a bespoke MCMC algorithm, analysed in Section 4.2, adjusted to take into account correlations between model parameters.

4.2 Adaptive Markov Chain Monte Carlo (AMCMC)

Correlations between model parameters can slow down the MCMC algorithm significantly. The reason for this is not self-evident, but can be made clearer with the use of a graph. To demonstrate this, we consider bi-variate Gaussian data – initially with no correlation, and then with some correlation, shown in Figure 4.1. We can observe that, unlike with uncorrelated data – where both the data and Gaussian proposal distribution are symmetric in all dimensions – when data is correlated (Figure 4.1.b), a Gaussian proposal distribution would cover large low-density areas, reducing the probability of a candidate sample being accepted. This occurs because high probability-density regions occupy a smaller part of the proposal distribution as a result of the correlation. Therefore, we would like to use a proposal distribution similar to the red ellipse in Figure 4.1.b, which is in some way 'tailored' to the correlation of the data – i.e. to select candidate samples over correlated directions in order to increase the acceptance

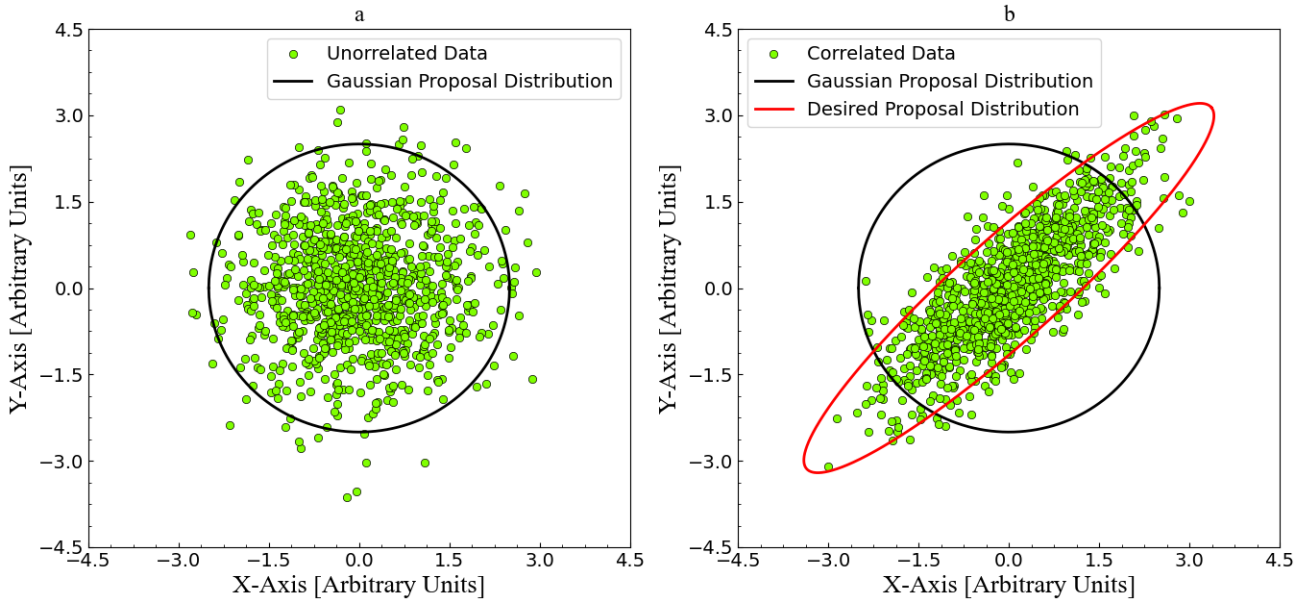

Figure 4.1: a) Uncorrelated bi-variate Gaussian data shown in green dots. b) Correlated bi-variate Gaussian data shown in green dots, with a correlation coefficient of 0.8. The black circles demonstrate what a Gaussian proposal distribution would look like, with the density of green dots corresponding to probability density in parameter space. The red ellipse in (b) depicts what would be a desirable proposal distribution for this correlated data.

rate efficiency. Of course, the example in Figure 4.1 is simplified to 2 dimensions (parameters) – but without loss of generality for more parameters.

To create a proposal distribution that best captures the correlation of the parameters, we use the **covariance matrix C**, which gives the covariance between all pairs of model parameters, $\mathbf{C} = \langle (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T \cdot (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \rangle$ [34]; this gives the expected value of the product of the deviations of any two model parameters from their respective means. To obtain the covariance matrix, we use NumPy's *numpy.cov* function [35], which requires us to pass as an argument the following matrix of all chain samples 'stacked' together:

$$\mathbf{C} = \text{np.cov}\left(\begin{bmatrix} \theta_1^{1,1} & \theta_2^{1,1} & & \theta_P^{1,1} \\ \vdots & \vdots & \cdots & \vdots \\ \theta_1^{1,N} & \theta_2^{1,N} & & \theta_P^{1,N} \\ \\ \vdots & \vdots & \ddots & \vdots \\ \\ \theta_1^{M,1} & \theta_2^{M,1} & & \theta_P^{M,1} \\ \vdots & \vdots & \cdots & \vdots \\ \theta_1^{M,N} & \theta_2^{M,N} & & \theta_P^{M,N} \end{bmatrix}\right)$$

where $\theta_p^{m,n}$ is the n'th sample for parameter θ_p from chain m, with $p \in \{1, \ldots, P\}$, $m \in \{1, \ldots, M\}$, and $n \in \{1, \ldots, N\}$, for M chains with N samples. Next, we eigen-decompose the covariance matrix, where $\mathbf{S}$ represents the matrix whose column-vectors are the eigenvectors of

$\mathbf{C}$, and $\mathbf{\Lambda} \coloneqq \text{diag}(\boldsymbol{\lambda})$, where $\boldsymbol{\lambda}$ are the eigenvalues of $\mathbf{C}$, such that $\mathbf{C} = \mathbf{S\Lambda S}^{-1}$. This is akin to performing *Principal Component Analysis*, where the directions of the eigenvectors create an orthogonal basis where parameters are linearly uncorrelated, while the eigenvalues demonstrate how much the data vary in each of these directions. This now allows us to take random steps in each parameter independently, without co-varying other parameters, where step magnitudes are scaled by their respective eigenvalues:

$$\boldsymbol{\zeta} = \begin{pmatrix} \zeta_1 \\ \vdots \\ \zeta_P \end{pmatrix}, \tag{4.9}$$

where $\zeta_p \sim \mathcal{N}(0, \sigma^2 = \sqrt{\lambda_p})$, for $\lambda_p \in \boldsymbol{\lambda}$, and $p \in \{1, \ldots, P\}$. Then, we transform these back into parameter-space, where these steps are now along the correlated directions:

$$\boldsymbol{\delta\theta} = \mathbf{S}^T \boldsymbol{\zeta}. \tag{4.10}$$

We can prove that these correlated parameter steps have the same covariance as the original parameters:

$$\begin{aligned} \langle \boldsymbol{\delta\theta}^T \cdot \boldsymbol{\delta\theta} \rangle &= \langle \boldsymbol{\zeta}^T \mathbf{S} \cdot \mathbf{S}^T \boldsymbol{\zeta} \rangle \\ &= \mathbf{S} \langle \boldsymbol{\zeta}^T \cdot \boldsymbol{\zeta} \rangle \mathbf{S}^T \\ &= \mathbf{S}\, \text{diag}(\boldsymbol{\lambda})\, \mathbf{S}^T \\ &= \mathbf{S\Lambda S}^T \\ &= \mathbf{S\Lambda S}^{-1} \\ &= \mathbf{C} \equiv \langle (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T \cdot (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \rangle, \end{aligned} \tag{4.11}$$

having used the fact that $\mathbf{S}$ is orthogonal, therefore $\mathbf{S}^T = \mathbf{S}^{-1}$. Thus, instead of using a Gaussian proposal distribution for MCMC, we will take correlated steps from the previous positions, as follows:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathbf{S}^T \boldsymbol{\zeta}. \tag{4.12}$$

This new proposal distribution requires the computation of the covariance matrix of the model parameters, which itself requires a sufficiently large number of accepted points in each chain in order to be accurate. We can compute the covariance matrix at each iteration of the MCMC algorithm, noting that each time it will be more accurate, as we accrue more points. Due to the continual improvement of the covariance matrix, which adapts to the correlations of the parameters better with time and allows for an improving proposal distribution, we call this bespoke method *Adaptive MCMC* (AMCMC). Moreover, using multiple parallel chains yields a more accurate covariance matrix than a single chain would, as it is computed with more independent points from a larger volume of the posterior probability distribution.

The AMCMC algorithm is summarised below in Algorithm 2. All algorithms and relevant code developed can be found on this public GitHub repository [36].

Algorithm 2: Adaptive Markov Chain Monte Carlo (AMCMC)

$\boldsymbol{\theta}_0 \leftarrow \mathcal{U}(\boldsymbol{\theta}_{\min}, \boldsymbol{\theta}_{\max})$;
$\chi_0 \leftarrow \frac{1}{2}\sum_{c=1}^{C} \chi^2_{N_c} + \frac{C}{2}\chi^2_P$, for $\boldsymbol{\theta}_0$;
$\mathcal{Y}_0 \leftarrow \boldsymbol{\theta}_0$;
$t \leftarrow 0$;
while *True* **do**
  $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \mathbf{S}^T\boldsymbol{\zeta}$; /* $\mathbf{C} = \mathbf{S}\boldsymbol{\Lambda}\mathbf{S}^{-1}$, $\zeta_p \sim \mathcal{N}(0, \sigma^2 = \sqrt{\lambda_p})$ */
  $\chi_{t+1} \leftarrow \frac{1}{2}\sum_{c=1}^{C} \chi^2_{N_c} + \frac{C}{2}\chi^2_P$, for $\boldsymbol{\theta}_{t+1}$;
  $A(\boldsymbol{\theta}_{t+1}, \boldsymbol{\theta}_t) \leftarrow \min\left(1, \exp\left(-(\chi_{t+1} - \chi_t)/T\right)\right)$; /* $T \sim 1$ */
  $u \leftarrow \mathcal{U}(0, 1)$;
  if $u \leq A(\boldsymbol{\theta}_{t+1}, \boldsymbol{\theta}_t)$ **then**
    $\mathcal{Y}_{t+1} \leftarrow \boldsymbol{\theta}_{t+1}$;
  else
    $\mathcal{Y}_{t+1} \leftarrow \boldsymbol{\theta}_t$;
  end
  $t \leftarrow t + 1$;
end
return $(\mathcal{Y}_t)_{t\geq 0}$

At this point we have: formulated a simple but descriptive compartmental model, the SIRHD model; developed a mathematical framework within Bayesian inference used to infer model parameters; and built a bespoke sampling algorithm, the AMCMC algorithm, to numerically sample the posterior probability distribution, and hence obtain marginal distributions for each model parameter. In Chapter 5, we test the AMCMC algorithm by inferring parameters from simulated data with known model parameters, to ensure our inference method works correctly, before applying it to real data to make predictions.

Chapter 5

Application to Simulated Data

While the end-goal of this report is to make a short-term prediction about the course of the COVID-19 pandemic, we first need to ensure the inference method and sampling algorithms developed, reliably infer model parameters. To achieve this, we simulate data using our models, with known parameters, and then apply our inference method to infer said parameters from the data. This allows us to assess whether our inference method has successfully recovered the correct simulation parameters, before applying it to real COVID-19 data.

However, we know that, in reality, data we get from the pandemic is not produced by a well-behaved model, but is instead the product of numerous, intractable and unstable variables, such as testing capacity and quality. This often makes real data biased and noisy. To make testing of our inference method more reliable and robust, we want our simulated data to resemble real data as much as possible – so we must add noise.

5.1 Generating Noisy Data

We first consider the sources of noise in real data. There are two main kinds: random variations in the pandemic parameters stemming from small differences in behavioural dynamics and the nature of the disease pathogen itself, e.g. lock-downs, new variants; and measurement error from day-to-day differences in testing. We introduce both of these to the model.

To incorporate the first type, we let the model parameters vary in each time-step in the integration of the ODEs, drawing each parameter $\theta_p \in \boldsymbol{\theta}$ from a Gaussian distribution:

$$\theta_p \sim \mathcal{N}(\mu = \hat{\theta}_p, \sigma^2 = v_1\hat{\theta}_p), \tag{5.1}$$

centred around some mean $\hat{\theta}_p$, where $v_1 \in (0, 1)$. This alteration makes the system of ODEs that describe the model *stochastic*, as all model parameters, previously constant, are random variables. We note that since all model parameters represent a 'rate', e.g. *the number of people who recover* ***per unit time***, they are positive, thus we restrict the sampling from the Gaussian in Equation 5.1 to positive values.

The second type of noise is added after the solutions of the (now stochastic) system of ODEs have been numerically obtained, as described in Chapter 2 by drawing the values of the compartments $C(t)$ at each time-point from a Gaussian distribution, centred at the solutions

of the ODEs:

$$C(t) \sim \mathcal{N}(\mu = S_c(t), \sigma^2 = v_2 S_c(t)), \tag{5.2}$$

where $S_c(t)$ is the computed solution from the stochastic system of ODEs, for compartment c at time $t \in t_{eval}$, and $v_2 \in (0, 1)$. An example of noisy simulated data, obtained by combining these two sources of noise for a simple model – the SIR model – is shown in Figure 5.1. The

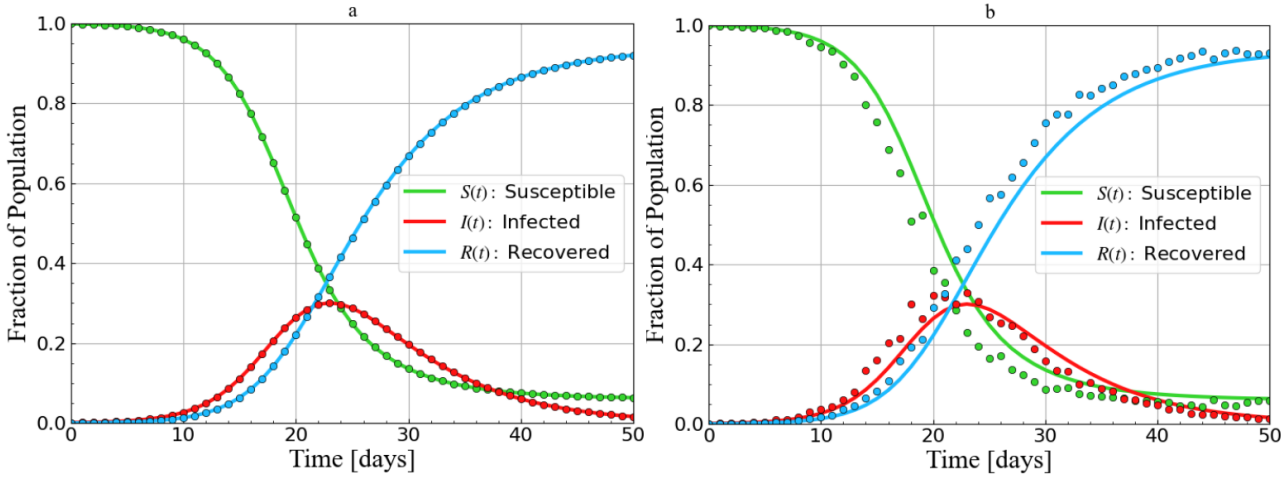

Figure 5.1: The continuous lines depict the solutions for the SIR model described by Equations 2.3, 2.4, and 2.5, with: $\beta = 0.5$, $\gamma = \frac{1}{6}$, $S(t = 0) = 999000$, $I(t = 0) = 1000$, and $R(t = 0) = 0$. a) Dots represent simulated data taken from the SIR model without noise. b) Dots represent simulated data taken from the same SIR model, with noise, where: $v_1 = 0.05$, and $v_2 = 0.07$.

deviations of the noisy data (dots) from the model solutions (continuous lines) represent the errors σ_i^c in Equation 3.8 for the process of inference.

Using this noise-generation technique, we can now produce noisy, simulated data, to test our inference method and sampling algorithms.

5.2 Testing of Inference Method & Sampling Algorithms

For the generation of the simulated data, we used the SIRHD model, summarised in Section 2.2.1, with the same initial conditions and model parameters used for the model depicted in Figure 2.5: $\beta = 0.5$, $\gamma = \frac{1}{6}$, $\eta = 0.00342865$, $\delta = 0.025$, $S(t = 0) = 999000$, $I(t = 0) = 1000$, $R(t = 0) = H(t = 0) = D(t = 0) = 0$ – parameters consistent with early COVID-19 data [22, 23]. Noise was added with the aforementioned techniques, using $v_1 = 0.05$ and $v_2 = 0.07$.

We first tested the MCMC algorithm using: 12 chains; a value of $\alpha = 0.2$ for the variance of the proposal distribution in Equation 4.3; a temperature of $T = 1$; a burn-in fraction of 10% of the initial samples of the chains to be discarded; and a Gelman-Rubin convergence criterion of $R_c = 1.05$. Furthermore, to test the inference method for the hardest case – 'maximal ignorance' about the model parameters, upon accurate results of which we can be confident the inference method works well, generally – we did not include any prior distributions. Hence,

the initial sample of each chain was drawn from a uniform distribution $\mathcal{U}(0, 1000)$, i.e. up to several orders of magnitude larger than parameters, hence effectively arbitrarily large [22, 23].

Next, we tested the AMCMC algorithm with the exact same hyper-parameters, using correlated steps obtained from the covariance matrix once there were at least 50 accepted points in every chain. The AMCMC algorithm was found to be 87.4% faster than the MCMC algorithm – that is, it accepted an average of 87.4% more samples in the same time unit than the MCMC algorithm, demonstrating the increased efficiency of sampling using a proposal distribution that takes the co-variances of model parameters into consideration. In addition, to minimise computing time and maximise collection efficiency, the process of generating the chains was separated from that of analysing them with the Python package *GetDist* to obtain marginal posterior probability distributions. Hence, the analysis of the chains could be repeated multiple times and very quickly, once the sampling algorithm had been run.

The AMCMC algorithm terminated after 8243 iterations, when the Gelman-Rubin convergence criterion dropped below $R_c = 1.05$ for all model parameters, as shown in Figure 5.2. Figure 5.2 demonstrates that there is a big discrepancy in the inferred model parameters be-

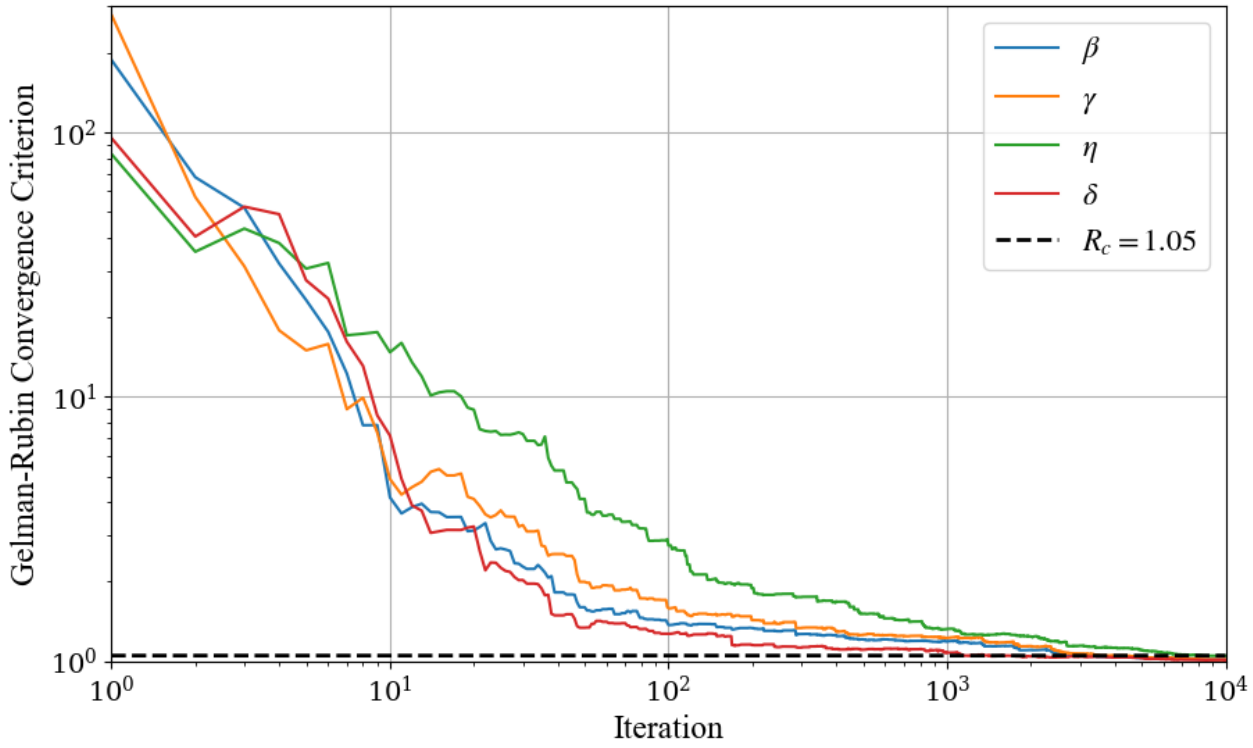

Figure 5.2: Gelman-Rubin convergence criterion, as defined in Equation 4.8, for all model parameters of the SIRHD model – β, γ, η, δ – plotted in a double-logarithmic scale for each iteration of the AMCMC algorithm. The dotted line represents the convergence criterion limit R_c, indicating a sufficient convergence of the chains.

tween chains at the beginning of the AMCMC algorithm, but, they converge at a similar rate, arriving at a 'stationary' distribution. The inferred marginal posterior probability distributions obtained with *GetDist* by analysing the chains returned by the AMCMC algorithm are depicted below in Figure 5.3.

The contours in the 2D distributions shown in Figure 5.3 represent the 1σ, 2σ, and 3σ

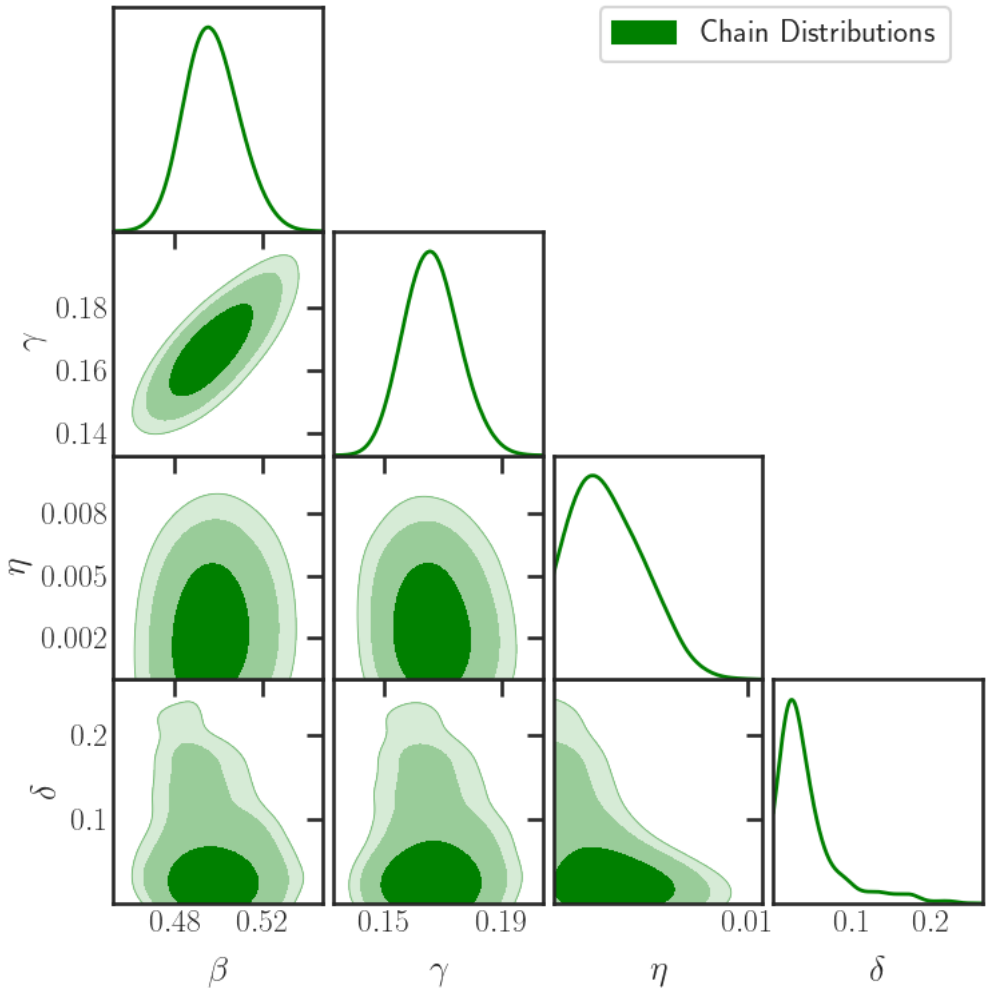

Figure 5.3: 'Triangle' plot of the marginal posterior probability distributions obtained using Python package *GetDist* by analysing the chains generated by the AMCMC algorithm. 1D marginal posterior probability distributions are shown on the diagonals, and 2D marginal posterior probability distributions are shown on the off-diagonals. The y-axis scales on the 1D distributions are omitted by default, as they are arbitrary.

limits. Furthermore, we notice that the 1D distributions of parameters η and δ are one-tailed with a minimum at 0. As a result, 2D distributions involving these two parameters are 'cut-off' at 0, demonstrating that the sampling algorithm has not explored the small-value tails of these distributions sufficiently. This indicates that chains can converge to a 'stationary' distribution for all parameters without necessarily exploring their high-σ tails. Moreover, the shape of the 2D distributions provides an indication for the correlation between model parameters; the more symmetric a 2D distribution is the less correlated parameters are, and vice versa. For example, parameters β and γ seem to have some level of positive correlation, suggesting that an increase in both parameters will have little effect on the output of the model. This could be explained intuitively; making a pathogen more transmissible (higher β) can be somewhat compensated by a disease with a shorter lifetime (higher γ), thus each effect 'cancels' each other out, having little effect on the model output.

From the 1D marginal posterior probability distributions we can obtain the marginal means, which represent the inferred values for each model parameter. Integrating the marginal distributions around their respective means, we can obtain their 1σ, and 2σ limits, indicating the ranges within which 68.2% and 95.4% of the samples exist. These provide an indication of the uncertainty in each inferred model parameter, i.e. we can say with 95.4% confidence that some parameter lies within the 2σ range. The final results for the inferred model parameters, including their uncertainty ranges, are summarised in Table 5.1.

Inferred Model Parameters from Simulated Data					
Parameter	Mean	1σ Lower Bound	2σ Lower Bound	1σ Upper Bound	2σ Upper Bound
β	0.497	0.484	0.474	0.509	0.523
γ	0.166	0.156	0.149	0.175	0.186
η	0.00299	0.0005	0	0.00416	0.00628
δ	0.0259	0.00218	0	0.0499	0.142

Table 5.1: Marginal posterior probability distribution means, 1σ and 2σ ranges of model parameters β, γ, η, and δ, inferred from simulated data using the AMCMC algorithm.

Using the marginal means, 1σ and 2σ limits, we can state that at the 2σ (or 95.4%) confidence interval, our inferred model parameters are:

$$\beta = 0.497^{+0.026}_{-0.023},\ \gamma = 0.166^{+0.020}_{-0.017},\ \eta = 0.003^{+0.003}_{-0.003},\ \text{and}\ \delta = 0.03^{+0.09}_{-0.03}.$$

Comparing the inferred model parameters with the parameters used to simulate the data, we notice that the marginal means are quite accurate, with an average deviation from the true parameters of 4.62%. However, parameters η and δ have very large uncertainties, while β and γ – which were observed to most affect output trajectory – have small uncertainties. This suggests that without the use of priors to focus in the exploration of the parameter space around specific regions, the parameters with the strongest effect on the trajectory tend to be inferred more precisely.

An additional way of assessing the success of the inference method is to select the sample in the chains that resulted in the highest posterior probability density – i.e. the *best-estimate* for the vector of parameters $\boldsymbol{\theta}^*$ given by Equation 3.10 – and run the SIRHD model with those parameters. In Figure 5.4, we compare the results from this, for the infected compartment, in particular, with the simulated data. We notice that the best-estimate parameters $\boldsymbol{\theta}^*$ yield a curve for the infected compartment that matches the noisy data very closely. This can be evaluated quantitatively with the *p-value* test, by calculating the mean squared weighted deviations between the noisy data and the best-estimate parameters' data, and then dividing them by the number of degrees of freedom [37]. This yields a p-value of $p = 0.9988$, meaning that we are 99.88% confident the best-estimate parameters describe the model underlying the noisy data.

So far, we have used our inference method and sampling algorithms to infer model parameters from noisy simulated data. We showed that: the AMCMC algorithm is indeed considerably more efficient than the MCMC algorithm; the parallel chains converge to a 'stationary' probability distribution; and the inferred model parameters are relatively accurate, even without the use of prior distributions to 'guide' our search, while best-estimate parameters can be used

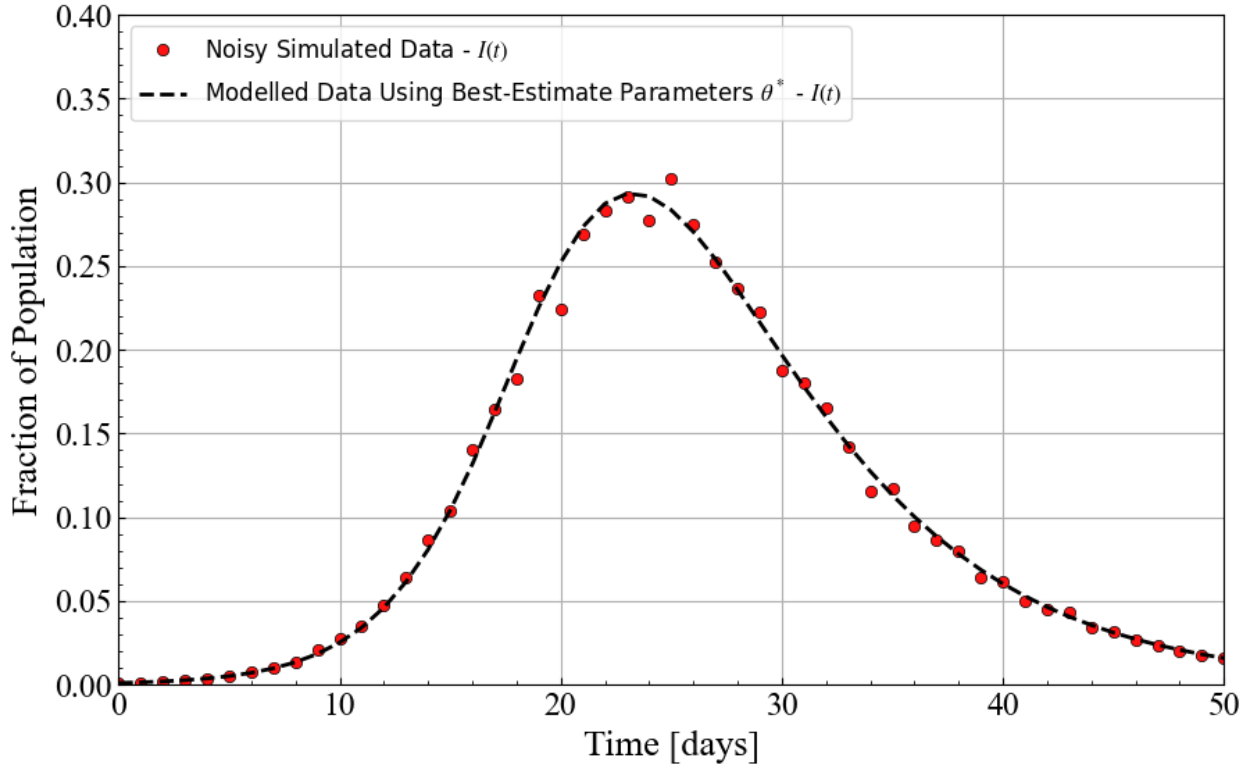

Figure 5.4: The noisy, simulated data generated for the infected compartment $I(t)$ of the SIRHD model is shown with red dots. The black, dashed line represents the infected compartment of the SIRHD model using the best-estimate parameters $\boldsymbol{\theta}^*$ obtained by the AMCMC algorithm.

to reproduce data that very closely matches simulated data. We can now proceed to infer parameters of the COVID-19 pandemic, so that we can make a short-term prediction about its progression.

Chapter 6

Application to Real Data

Having verified that our inference method works, we can now use it to infer parameters of the pandemic for a given time-period, and thus make a short-term prediction about its trajectory.

6.1 Inference of Parameters from Real Data

We infer parameters for September 20^{th} 2020 to December 29^{th} 2020 – a period of time demonstrating simple but interesting dynamics: no new variants, no immunity effects from vaccination, and a strict enforced lockdown before Christmas of 2020. We used modelled data for the prevalence of the virus from the *Coronavirus (COVID-19) Infection Survey for England* by the Observatory for National Statistics (O.N.S.) [16], extrapolated from large-scale population-sample studies. Therefore, this data is significantly more accurate than the daily number of cases, which suffers from testing bias, and also includes quantified uncertainties for each datapoint, used as the errors σ_i^c in Equation 3.8 during inference – wherein we only use the infected compartment of the SIRHD model, to which this data directly corresponds.

We start by initialising the SIRHD model with real data for the English population for the 20^{th} of September 2020, taken from the UK government's coronavirus database [38]:

- $N = 56000000$,
- $S(t = 0) = 50216776$,
- $I(t = 0) = 140000$,
- $R(t = 0) = 5600000$,
- $H(t = 0) = 1399$,
- $D(t = 0) = 41825$.

Next, given that model parameters γ, η, and δ are unaffected by exogenous factors and that our selected time period features no endogenous changes – such as a new variant, which would for example alter γ – we can use well-documented priors from literature [22, 23], to increase sampling around these values and hence reduce uncertainty in their posterior probability distributions. More specifically [22, 23]:

- $\gamma \sim \mathcal{N}(0.1666, 0.008)$,
- $\eta \sim \mathcal{N}(0.003429, 0.00069)$,
- $\delta \sim \mathcal{N}(0.025, 0.005)$.

Parameter β, on the other hand, depends heavily on fluctuating exogenous factors, so we do not use a prior and instead make it time-dependent $\beta := \beta(t)$, by allowing it to take different values for each week in the inference time-period. In practice, this creates a series of additional parameters in the model – a value of β for each week – which are inferred independently.

Finally, we run the AMCMC algorithm, using: 12 chains; a value of $\alpha = 0.1$ for the variance of the proposal distribution in Equation 4.3; a temperature of $T = 1$; a burn-in fraction of 10% of the initial samples of the chains to be discarded; and a Gelman-Rubin convergence criterion of $R_c = 1.05$. Initial samples in the chains were drawn from the prior probability distributions. The AMCMC algorithm terminated after 39999 iterations, yielding the inferred model parameters summarised in Table A.1 in Appendix A.

6.2 Short-Term Prediction of Pandemic Trajectory

We can run the SIRHD model using the marginal means of the inferred model parameters in Table A.1, plotting a model 'fit' of the data. However, to make a short-term prediction, we let the model evolve beyond the 28$^{\text{th}}$ of December 2020 for 25 days, using the marginal means of the inferred model parameters (in the case of β, from the last week of the inference period, which was nearest our extrapolation), summarised in Table 6.1. This results in a short-

Inferred Model Parameters from Real Data					
Parameter	Mean	1σ Lower Bound	2σ Lower Bound	1σ Upper Bound	2σ Upper Bound
β	0.2361	0.1711	0.0952	0.3043	0.4723
γ	0.1687	0.1624	0.1540	0.1761	0.1827
η	0.0035	0.0031	0.0028	0.0039	0.0043
δ	0.0244	0.0021	0.0127	0.0293	0.0342

Table 6.1: Marginal posterior probability distribution means, 1σ and 2σ ranges of model parameters β, γ, η, and δ, inferred from O.N.S. data, for the 29$^{\text{th}}$ of December 2020 using the AMCMC algorithm.

term prediction for the time-period 29$^{\text{th}}$ of December 2020 to 23$^{\text{rd}}$ of January 2021, assuming constant model parameters in that time. We also run Monte Carlo simulations of the SIRHD model with parameters randomly sampled between their lower and upper 2σ uncertainty, for the prediction time, representing the 95.4% confidence interval in our prediction for the trajectory of the pandemic. The model 'fit' and the short-term prediction are depicted in Figure 6.1. The real data for the prediction time – *not* used for inference – have also been plotted, to evaluate against. We notice in Table 6.1 that the inferred parameters have considerably smaller uncertainty than those in Table 5.1, where priors were not used, demonstrating how priors can help obtain more precise marginal posterior probability distributions. In Figure 6.1, we observe

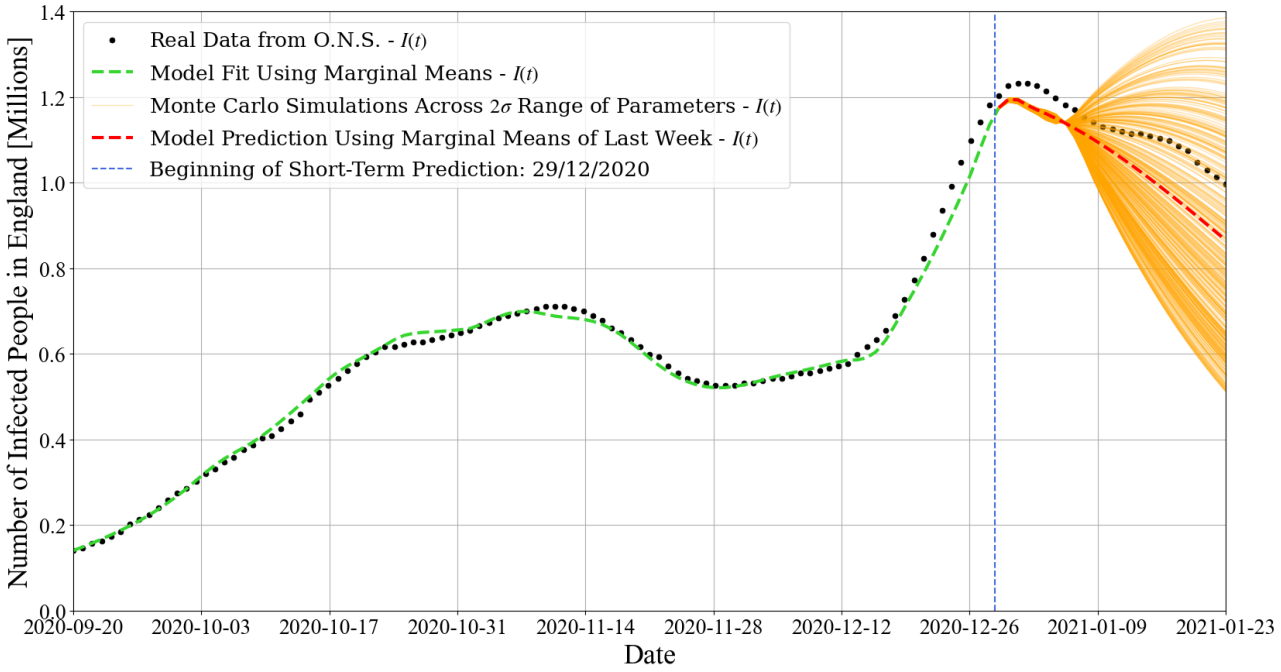

Figure 6.1: O.N.S. data for the prevalence of COVID-19 in England is shown by the black dots. The dashed green line is the model 'fit' of the data using the marginal means of the model parameters inferred using the AMCMC algorithm. The dashed red line represents the short-term prediction for the pandemic using the marginal means of the model parameters from 28/12/2020, while the orange lines represent Monte Carlo simulations of the model with parameters randomly sampled across their 2σ uncertainty.

a successful model 'fit', following the data very closely. Furthermore, regarding the predicted trajectory, the inferred parameters were able to predict the turning-point, with a decrease in the prevalence of the virus from the 30^{th} of December 2020, exhibited by the data – further demonstrating robust inferred parameters. The prediction is not as close to the real data as the model 'fit' – largely attributable to β being kept constant and not capturing unknown 'future' exogenous factors, which also explains the divergence between prediction and data as the error propagates – but the general trend is correct. Finally, we observe that the 2σ uncertainty is significant enough for some curves of the Monte Carlo simulation to increase (while the real data decreases), although the real data falls well within the 2σ uncertainty range near the end of the prediction time-frame. Thus, predictions can only be reliably made for the near future.

Chapter 7

Conclusions & Future Considerations

In this report, we began by building basic compartmental models like the SIR and the SIRHD, detailing how they can be programmed and run. The limitations of their simplicity were then discussed, motivating the development of more complex compartmental models – which are, however, more sensitive to errors in the assumptions they are built on. The mathematical framework of Bayesian inference – with relevant assumptions tailored to the particular models – was then used to derive equations to represent probability distributions for inferred model parameters. Next, we developed bespoke Markov Chain Monte Carlo (MCMC) sampling algorithms that take correlations of model parameters into consideration, to numerically approximate their probability distributions. We then tested these algorithms on simulated data with known model parameters, accurately inferring them with a mean deviation of 4.62%, describing the model underlying the simulated data with a p-value of $p = 0.9988$. Finally, we applied our inference method to real COVID-19 data from September 20^{th} 2020 to December 28^{th} 2020, using the inferred model parameters to predict the trajectory of the pandemic for the subsequent 25-day period. Compared with subsequent data, the model successfully predicted a turning-point and a decrease in cases. Monte Carlo simulations were used to illustrate the uncertainty in the predicted trajectory, given the 2σ parameter uncertainties, yielding model behaviours that included divergence from the real data.

The simplicity of the compartmental models developed renders them robust to errors in assumptions, but offers limited insight into the various dynamics of a pandemic. Hence, future work could consider effects such as waning immunity, vaccination, and age-stratification. Furthermore, the ability of different models, like agent-based models, to infer parameters could be investigated. Finally, MCMC algorithms can become computationally expensive and slow when sampling higher-dimensional parameter spaces, so further research into making them more efficient, e.g. by optimising *temperature*, could be especially beneficial to epidemiological modelling of infectious diseases like COVID-19.

Appendix A

Appendices

A Inferred Model Parameters from Real Data

Inferred Model Parameters from Real Data					
Parameter	Mean	1σ Lower Bound	2σ Lower Bound	1σ Upper Bound	2σ Upper Bound
β_1	0.2570	0.2447	0.2321	0.2702	0.2818
β_2	0.2624	0.2393	0.2179	0.2844	0.3063
β_3	0.2347	0.2165	0.1976	0.2546	0.2710
β_4	0.2435	0.2189	0.2056	0.2617	0.2899
β_5	0.2260	0.2076	0.1852	0.2479	0.2640
β_6	0.2086	0.1875	0.1623	0.2336	0.2515
β_7	0.2252	0.2001	0.1843	0.2442	0.2715
β_8	0.2054	0.1805	0.1646	0.2230	0.2555
β_9	0.1799	0.1621	0.1306	0.2044	0.2226
β_{10}	0.2030	0.1749	0.1611	0.2246	0.2513
β_{11}	0.2344	0.2119	0.1887	0.2583	0.2807
β_{12}	0.2379	0.2120	0.1883	0.2655	0.2873
β_{13}	0.3033	0.2738	0.2481	0.3219	0.3608
β_{14}	0.3062	0.2767	0.2537	0.3322	0.3651
β_{15}	0.2361	0.2137	0.1858	0.2627	0.2822
β_{16}	0.2361	0.2175	0.1884	0.2790	0.3067
β_{17}	0.2361	0.2214	0.1995	0.2776	0.2972
β_{18}	0.2361	0.1711	0.0951	0.3043	0.4723
γ	0.1687	0.1624	0.1540	0.1761	0.1827
η	0.0035	0.0031	0.0028	0.0039	0.0043
δ	0.0244	0.0021	0.0127	0.0293	0.0342

Table A.1: Marginal posterior probability distribution means, 1σ and 2σ ranges of model parameters β_w, γ, η, and δ, where $w \in \{1, \ldots, 18\}$ for 18 weeks, inferred from O.N.S. data, for the 20^{th} of September 2020 to the 4^{th} of January 2021, using the AMCMC algorithm.

Bibliography

[1] Neil M. Ferguson et al. *Imperial College COVID-19 Response Team 2020-2021 Report (22-03-2021)*. Imperial College London, 2021. DOI: 10.25561/87192.

[2] World Health Organisation. *WHO Coronavirus (COVID-19) Dashboard*. 2022. URL: https://covid19.who.int/ (visited on 04/13/2022).

[3] Imperial College London. *MRC Centre for Global Infectious Disease Analysis*. 2022. URL: https://www.imperial.ac.uk/mrc-global-infectious-disease-analysis/ (visited on 04/13/2022).

[4] Ronald Ross. "An Application of the Theory of Probabilities to the Study of A Priori Pathometry. - Part I". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 92.638 (Feb. 1916), pp. 204–230. DOI: 10.1098/rspa.1916.00074.

[5] Ronald Ross and Hilda Hudson. "An Application of the Theory of Probabilities to the Study of A Priori Pathometry. - Part II". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 93.650 (May 1917), pp. 212–224. DOI: 10.1098/rspa.1917.0014.

[6] Ronald Ross and Hilda Hudson. "An Application of the Theory of Probabilities to the Study of A Priori Pathometry. - Part III". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 93.650 (May 1917), pp. 225–240. DOI: 10.1098/rspa.1917.0015.

[7] William Ogilvy Kermack and Anderson Gray McKendrick. "A Contribution to the Mathematical Theory of Epidemics". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 115.772 (Aug. 1927), pp. 700–721. DOI: 10.1098/rspa.1927.0118.

[8] Ross Beckley et al. "Modelling epidemics with differential equations". In: *Journal of Physics A: Mathematical and Theoretical* 54.50 (June 2013).

[9] Tiberiu Harko, Francisco S. N. Lobo, and M. K. Mark. "Exact analytical solutions of the Susceptible-Infected-Recovered (SIR) epidemic model and of the SIR model with equal death and birth rates". In: *Applied Mathematics and Computation* 236 (May 2014), pp. 184–194. DOI: 10.1016/j.amc.2014.03.030.

[10] Joel C. Miller. "A note on the derivation of epidemic final sizes". In: *Bulletin of Mathematical Biology* 74 (July 2012), pp. 2125–2141. DOI: 10.1007/s11538-012-9749-6.

[11] Joel C. Miller. "Mathematical models of SIR disease spread with combined non-sexual and sexual transmission routes". In: *Infectious Disease Modelling* 74 (July 2017). DOI: 10.1016/j.idm.2016.12.003.

[12] Martin Kroger and Reinhard Schlickeiser. "Analytical solution of the SIR-model for the temporal evolution of epidemics. Part A: Time-independent reproduction factor". In: *Journal of Physics A: Mathematical and Theoretical* 53.505601 (Nov. 2020). DOI: 10.1088/1751-8121/abc65d.

[13] Martin Kroger and Reinhard Schlickeiser. "Analytical solution of the SIR-model for the temporal evolution of epidemics. Part B: Semi-time case". In: *Journal of Physics A: Mathematical and Theoretical* 54.174601 (Nov. 2020). DOI: 10.1088/1751-8121/abed66.

[14] The SciPy Community. *scipy.integrate.solve_ivp*. 2022. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html (visited on 04/15/2022).

[15] J. R. Dormand and P. J. Prince. "A family of embedded Runge-Kutta formulae". In: *Journal of Computational and Applied Mathematics* 6.1 (Mar. 1980), pp. 19–26. DOI: 10.1016/0771-050X(80)90013-3.

[16] Office for National Statistics. *Coronavirus (COVID-19) Infection Survey: England*. 2020. URL: https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/conditionsanddiseases/datasets/coronaviruscovid19infectionsurveydata/2020 (visited on 04/15/2022).

[17] Ruian Ke et al. "Estimating the reproductive number R0 of SARS-CoV-2 in the United States and eight European countries and implications for vaccination". In: *Journal of Theoretical Biology* 517 (May 2021), p. 110621. ISSN: 0022-5193. DOI: https://doi.org/10.1016/j.jtbi.2021.110621.

[18] Stephen A. Lauer et al. "The Incubation Period of Coronavirus Disease 2019 (COVID-19) From Publicly Reported Confirmed Cases: Estimation and Application". In: *Annals of Internal Medicine* (2020). DOI: https://doi.org/10.7326/M20-0504.

[19] Athalia Christie et al. "Decreases in COVID-19 Cases, Emergency Department Visits, Hospital Admissions, and Deaths Among Older Adults Following the Introduction of COVID-19 Vaccine — United States, September 6, 2020–May 1, 2021". In: *CDC: Morbidity and Mortality Weekly Report (MMWR)* 70.23 (June 2021), pp. 858–864. DOI: 10.15585/mmwr.mm7023e2.

[20] Albin Andreasson. "Different Compartmental Models' Sensitivities to Uncertain Parameters". KTH Royal Institutre of Technology School of Electrical Engineering and Computer Science, 2021. URL: https://www.diva-portal.org/smash/get/diva2:1605633/FULLTEXT01.pdf (visited on 04/16/2022).

[21] Weston C Roda et al. "Why is it difficult to accurately predict the COVID-19 epidemic?" In: *Infectious Disease Modelling* 5 (Mar. 2020), pp. 271–281. DOI: 10.1016/j.idm.2020.03.001.

[22] Nir Menachemi et al. "How Many SARS-CoV-2-Infected People Require Hospitalization? Using Random Sample Testing to Better Inform Preparedness Efforts". In: *Journal of Public Health Management Practice* 27.3 (May 2021), pp. 246–250. DOI: 10.1097/PHH.0000000000001331.

[23] Centers for Disease Control and Prevention. *In-hospital Mortality Among Hospital Confirmed COVID-19 Encounters by Week From Selected Hospitals*. 2022. URL: https://www.cdc.gov/nchs/covid19/nhcs/hospital-mortality-by-week.htm (visited on 04/17/2022).

[24] Thomas Bayes. "An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S". In: *Journal of Philosophical Transactions of the Royal Society of London* 53 (Jan. 1763), pp. 370–418. DOI: 10.1098/rstl.1763.0053.

[25] Michael Dekking. *A modern introduction to probability and statistics: understanding why and how*. Springer, 2005. ISBN: 9781852338961.

[26] Roger Eckhardt. "Stan Ulam, John von Neumann, and the Monte Carlo method". In: *Los Alamos Science* 15 (Jan. 1987), pp. 131–137. URL: http://library.lanl.gov/cgi-bin/getfile?15-13.pdf (visited on 04/20/2022).

[27] Pierre Bremaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer Science & Business Media, 2013. ISBN: 9781475731248.

[28] Radford M. Neal. *Probabilistic Inference Using Markov Chain Monte Carlo Methods*. University of Toronto, 1993, pp. 58–65. URL: http://www.cs.toronto.edu/~radford/review.abstract.html (visited on 04/21/2022).

[29] M. E. J. Newman and G. T. Barkema. *Monte Carlo Methods in Statistical Physics*. Oxford University Press, 1999. ISBN: 9780198517979.

[30] Andrew Gelman and Donald B. Rubin. "Inference from Iterative Simulation Using Multiple Sequences". In: *Journal of Statistical Sciences* 7.4 (Nov. 1992), pp. 457–511. DOI: 10.1214/ss/1177011136.

[31] Dootika Vats and Christina Knudson. "Revisiting the Gelman-Rubin Diagnostic". In: *Statistics: Computation* (Dec. 2018), pp. 1–2. DOI: 10.48550/arXiv.1812.09384.

[32] Andrew Gelman et al. *Bayesian Data Analysis. 3rd Edition*. CBC Press, 2013. ISBN: 9780429113079. DOI: 10.1201/b16018.

[33] Antony Lewis. "GetDist: a Python package for analysing Monte Carlo samples". In: *Instrumentation and Methods for Astrophysics* (2019). DOI: 10.48550/ARXIV.1910.13970.

[34] Kun Il Park. *Fundamentals of Probability and Stochastic Processes with Applications to Communications*. Springer, 2018. ISBN: 9783319680743. DOI: 10.1007/978-3-319-68075-0.

[35] NumPy Developers. *numpy.cov*. 2022. URL: https://numpy.org/doc/stable/reference/generated/numpy.cov.html (visited on 04/25/2022).

[36] Georgios Alevras. *MSciProjectReport*. 2022. URL: https://github.com/GeorgeAlevras/MSciProjectReport (visited on 05/02/2022).

[37] David E. Bock, Paul F. Velleman, and Richard D. De Veaux. *Stats. Modelling the World*. Pearson Addison Wesley, 2009, pp. 606–627. ISBN: 9780321570444.

[38] UK Health Security Agency. *UK Summary — Coronavirus (COVID-19) in the UK*. 2022. URL: `https://coronavirus.data.gov.uk/` (visited on 04/30/2022).