Inference of Stochastic Disease Transmission Models Using Particle-MCMC and a Gradient Based Proposal

Conor Rosato  
Dept. of Electrical Engineering and Electronics  
University of Liverpool, United Kingdom  
c.m.rosato@liverpool.ac.uk

John Harris  
Public Health England  
United Kingdom  
john.p.harris@phe.gov.uk

Jasmina Panovska-Griffiths  
The Big Data Institute  
The Queen’s College, University of Oxford, United Kingdom  
jasmina.panovska-griffiths@queens.ox.ac.uk

Simon Maskell  
Dept. of Electrical Engineering and Electronics  
University of Liverpool, United Kingdom  
s.maskell@liverpool.ac.uk

Abstract—State-space models have been widely used to model the dynamics of communicable diseases in populations of interest by fitting to time-series data. Particle filters have enabled these models to incorporate stochasticity and so can better reflect the true nature of population behaviours. Relevant parameters such as the spread of the disease, $R_t$, and recovery rates can be inferred using Particle MCMC. The standard method uses a Metropolis-Hastings random-walk proposal which can struggle to reach the stationary distribution in a reasonable time when there are multiple parameters.

In this paper we obtain full Bayesian parameter estimates using gradient information and the No U-Turn Sampler (NUTS) when proposing new parameters of stochastic non-linear Susceptible-Exposed-Infected-Recovered (SEIR) and SIR models. Although NUTS makes more than one target evaluation per iteration, we show that it can provide more accurate estimates in a shorter run time than Metropolis-Hastings.

Index Terms—Differentiable particle filter, Particle-MCMC, gradients, NUTS, epidemics

I. INTRODUCTION

Calibration of epidemiological models using disease specific data is important when striving to understand how a disease evolves with time. Inferring parameters such as the effective contact and recovery rates, especially at the beginning of an outbreak, is vital for public health officials when assessing the deadliness of a disease.

Two methods for modelling diseases include: Agent-based models (ABMs) which simulate interactions with individuals within the population based on a set of rules; and the compartmental approach of splitting the population into unobservable compartments, for example Susceptible, Infected and Recovered (SIR) and allowing a fraction at every timestep $t$, to progress to the next compartment. State space models (SSMs) are a favoured method for representing the dependence of latent states in non-linear dynamical systems in a wide range of research fields. A SSM consists of a state equation,

\[ x_t \mid x_{t-1} \sim f(x_t \mid x_{t-1}, \theta), \]

which is parameterised by $\theta$ and models how the dynamical system moves from the previous state to the current state, $x_t$, at time $t$. They also include an observation equation

\[ y_t \mid x_t \sim f(y_t \mid x_t, \theta), \]

which describes how the observation, $y_t$, is linked to the state equation. Particle filters are a popular method for inferring the time-dependent hidden states of SSMs and have been previously applied to compartmental disease transmission models.

The two overarching methods used for estimating parameters in the context of epidemiological models, as outlined in a recent systematic review, are optimisation and sampling algorithms. Optimisation algorithms include grid search and iterative, descent-guided optimisation while sampling methods include Markov Chain Monte Carlo (MCMC) techniques, particle-MCMC (p-MCMC), Approximate Bayesian Computation (ABC) and history matching. One difference between these two groups of techniques is that optimisation algorithms find point estimates of the parameters while sampling methods can produce full Bayesian parameter estimates which in-turn will capture parameter uncertainty.
MCMC algorithms work by drawing a sample, which is based on the previous sample, from a target distribution, \( \pi(\theta) \). An accept/reject step is included which determines whether to keep or discard the drawn sample. Metropolis-Hastings (MH) randomly samples from \( \pi(\theta) \) without reference to its structure. It can therefore take the Markov Chain a long time to reach its stationary distribution if \( \theta \) has multiple dimensions. Hamiltonian Monte Carlo (HMC) \cite{11} and the No-U-Turn Sampler (NUTS) \cite{12} use gradient information about the log-posterior of \( \theta \) so can efficiently explore \( \pi(\theta) \). Probabilistic programming languages (ppls), such as Stan \cite{13}, have made MCMC an accessible tool for calibrating epidemiological models \cite{14,15} because these ppls use NUTS.

Particle-MCMC (p-MCMC) \cite{16} combines the particle filter and MCMC to make state and parameter estimates by performing Bayesian inference on non-linear non-Gaussian scenarios where standard MCMC methods can fail. A detailed explanation is given in section \[12\]. Particle-MCMC has been use to infer parameters of simulated genealogies and time series data \cite{17}, a dengue outbreak \cite{18}, non-communicable diseases in Egypt \cite{19}, cholera in Bangladesh \cite{20}, COVID-19 in the United Kingdom (UK) \cite{21}, the 2009 A/H1N1 pandemic in England \cite{22} and an ABM in \cite{23}. It has been suggested in \cite{24,25} that a potential reason why p-MCMC has yet to become a mainstream method for modelling epidemics is due to the complexity of the mathematics behind the algorithm: \cite{25} make the methodology more accessible by providing a step-by-step tutorial. The standard proposal used in calibration \cite{24,25} make the methodology more accessible by providing a step-by-step tutorial. The standard proposal used in calibration of epidemiological models when using p-MCMC is the MH random walk \cite{17,19,21,23,25} which inherits the same drawbacks as explained previously for MCMC.

In this piece of work we focus on estimating the parameters of stochastic disease compartmental models using the recent development of a variant of p-MCMC that uses NUTS as the proposal \cite{26}.

The paper is organised as follows: section \[I\] explains how to combine p-MCMC and NUTS in more detail, before section \[II\] describes the models we will consider and the results obtained. Section \[IV\] then concludes the paper.

II. PARTICLE-MCMC

In this section we outline how to construct the p-MCMC algorithm and show how an unbiased estimate of the log-likelihood, given by the particle filter, can be used in the MH proposal within the MCMC algorithm. We also show how to estimate the gradient of the log-likelihood which is used when proposing a new vector of \( \theta' \) using NUTS.

Bayesian calibration of SSMs seen in \[1\] and \[2\] is undertaken with the goal of estimating the parameter posterior distribution \( p(\theta'|y) \), which involves finding a set of \( \theta \) that best represents the data, \( y \). This can be done by using Bayes theorem:

\[
p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)},
\]

where \( p(\theta) \) is the prior.

A. Metropolis Hastings

In the well understood MH proposal a set of parameters \( \theta' \) is drawn from a proposal distribution \( q(\theta'|\theta) \), which is commonly chosen to be a Gaussian centered around the current position, \( \theta \), and a variance parameter, also known as step-size, which is chosen by the user. An accept or reject step is introduced that determines whether to accept \( \theta' \) as part of the Markov chain or reject and revert back to \( \theta \). If \( \theta' \) is accepted the inputs for the next iteration are \( \theta = \theta' \) and \( p(y_1:T|\theta) = p(y_1:T|\theta') \). If \( \theta' \) is rejected the inputs are \( \theta = \theta \) and \( p(y_1:T|\theta) = p(y_1:T|\theta') \). This process occurs for, \( M \) MCMC iterations and at each iteration two values are stored: a current value of \( \theta \) and a particle filter estimate of \( p(y_1:T|\theta) \). The acceptance probability is

\[
\alpha(\theta, \theta') = \min \left\{ 1, \frac{p(\theta') q(\theta'|\theta)}{p(\theta) q(\theta'|\theta)} \right\}.
\]

Note \( p(\theta) = p(\theta|y) \). When estimating \( p(\theta) \) a tractable likelihood is needed which in many cases is not available. The particle filter (explained in section \[13\]) provides a method to approximate the marginal likelihood, \( p(y_1:T|\theta) \) which allows us to formulate the acceptance step as

\[
\alpha(\theta, \theta') = \min \left\{ 1, \frac{p(\theta') p(y_1:T|\theta') q(\theta'|\theta)}{p(\theta) p(y_1:T|\theta) q(\theta'|\theta)} \right\}.
\]

This is the particle marginal Metropolis-Hastings (PMMH) algorithm described in \cite{16} and is described in Algorithm \[1\].

B. No-U-Turn Sampler

NUTS \cite{12} is a gradient based method and an extension of HMC \cite{27,28}. Both methods use the leapfrog numerical integrator to generate new samples by discretising Hamilton’s equations which are defined to be

\[
\frac{d\theta}{dt} = \frac{\partial H}{\partial m},
\]

\[
\frac{dm}{dt} = -\frac{\partial H}{\partial \theta},
\]

where \( m \) is the momentum. The Hamiltonian \( H \) is defined to be

\[
H(\theta, m) = U(\theta) + K(m),
\]

where \( K(m) \) is the kinetic energy and \( U(\theta) \) is the potential energy. Hamilton’s equations are a pair of differential equations that describe a system in terms of its position and momentum where the potential of the system is defined by \( U = -\log(p(\theta)) \). Sampling a random initial \( m \), from a Gaussian, a new state \( \theta' \) is proposed using the gradient of \( \pi(\theta) \). The leapfrog method is symplectic, meaning it preserves the geometry of the system which leads to high acceptance rates, and reversible, meaning detailed balance is maintained. The generated samples from HMC are governed by two user defined parameters: the number of steps \( L \) and the stepsize used by leapfrog \( \Delta \). NUTS adaptively calibrates \( L \) by building trajectories that take steps both forward and backwards.
C. Particle Filter

The benefit of using a particle filter is its ability to model SSMs that are nonlinear and non-Gaussian. It does this by representing the probability density function over the hidden states, \( x_t \), with a set of \( N \) random samples (particles), each being an independent hypothesis of \( x_t \).

The particle filter propagates particles forward in time using the model function (often referred to as the dynamics) in (1) which encodes some prior information on how the state evolves with time. Due to \( \theta \) being static, some noise needs to be introduced which will improve the diversity among the drawn samples. An observation model (often referred to as the likelihood), seen in (2), is used to associate a measurement with the state, \( x_t \), at every timestep. The dynamics and likelihood can be parameterised by \( \theta \) such that

\[
p(y_{1:t}, x_{1:t}|\theta) = p(y_1|x_1, \theta)p(x_1|\theta) \prod_{\tau=2}^{t} p(y_\tau|x_\tau, \theta) p(x_\tau|x_{\tau-1}, \theta)
\]

where \( p(x_{\tau-1}, y_\tau) \) is the proposal. Different options exist for formulating a proposal which are described in [26]. However, in this paper we choose the proposal to be equal the dynamic model, \( q(x_t|x_{t-1}, y_t) = p(x_t|x_{t-1}, \theta) \), which simplifies (9) to

\[
w_{1:t}^{(\theta,i)} = p(y_1|x_{1:t}^{(\theta,i)})w_{1:t-1}^{(\theta,i)},
\]

where \( q(x_{t-1}|x_{t-1}^{(\theta,i)}, y_t) \) is the proposal. Different options exist for formulating a proposal which are described in [26]. However, in this paper we choose the proposal to be equal the dynamic model, \( q(x_t|x_{t-1}^{(\theta,i)}, y_t) = p(x_t|x_{t-1}^{(\theta,i)}, \theta) \), which simplifies (9) to

\[
w_{1:t} = w_{1:t-1}p(y_1|x_{1:t})\frac{\prod p(x_{\tau-1}^{(\theta,i)})}{q(x_{\tau-1}^{(\theta,i)})},
\]

where \( q(x_{\tau-1}^{(\theta,i)}) \) is the proposal. Different options exist for formulating a proposal which are described in [26]. However, in this paper we choose the proposal to be equal the dynamic model, \( q(x_t|x_{t-1}^{(\theta,i)}, y_t) = p(x_t|x_{t-1}^{(\theta,i)}, \theta) \), which simplifies (9) to

\[
w_{1:t}^{(\theta,i)} = p(y_1|x_{1:t}^{(\theta,i)})w_{1:t-1}^{(\theta,i)}.
\]

The joint distribution, \( p(y_{1:t}, x_{1:t}|\theta) \), can be estimated using the unnormalised weights from (10) such that

\[
\int p(y_{1:t}, x_{1:t}|\theta) f(x_{1:t}) dx_{1:t} \approx \frac{1}{N} \sum_{i=1}^{N} w_{1:t}^{(\theta,i)} f(x_{1:t}^{(i)}),
\]

which is unbiased. The posterior distribution, \( p(x_{1:t}|y_{1:t}, \theta) \), can then be estimated by

\[
\int p(x_{1:t}|y_{1:t}, \theta) f(x_{1:t}) dx_{1:t} = \int \frac{p(y_{1:t}, x_{1:t}|\theta)}{p(y_{1:t}|\theta)} f(x_{1:t}) dx_{1:t}.
\]

If \( f(x_{1:t}) = 1 \) then

\[
p(y_{1:t}|\theta) = \int p(y_{1:t}, x_{1:t}|\theta) dx_{1:t} \approx \frac{1}{N} \sum_{i=1}^{N} \tilde{w}_{1:t}^{(\theta,i)} f(x_{1:t}^{(i)}),
\]

which is similar to (11). The normalised weights (which differ from those in (10)) can be calculated using

\[
\tilde{w}_{1:t}^{(\theta,i)} = \frac{w_{1:t}^{(\theta,i)}}{\sum_{j=1}^{N} w_{1:t}^{(\theta,j)}},
\]

which are used when estimating the integral

\[
\int p(x_{1:t}|y_{1:t}, \theta) f(x_{1:t}) dx_{1:t} = \sum_{i=1}^{N} \tilde{w}_{1:t}^{(\theta,i)} f(x_{1:t}^{(i)}).
\]

We note that while (11) is an unbiased estimate, (15) is biased (due to it being a ratio of estimates).

1) Resampling: The methods described in section II-C until now results in the sequential importance sampling (SIS) algorithm. One limitation of SIS is that the normalised weights in (14) becoming increasingly skewed at each increment of \( t \) with one becoming close to 1 and the others 0.

A widely used method for overcoming this is to calculate the number of effective samples at each iteration of \( t \) as

\[
N_{eff} = \frac{1}{\sum_{i=1}^{N} \tilde{w}_{1:t}^{(\theta,i)}},
\]

and comparing \( N_{eff} \) with a user-defined threshold. If \( N_{eff} \) is less than the threshold, resampling is implemented. Multinomial resampling is a popular method which involves drawing from the current particle set \( N \) times, with probabilities proportional to the corresponding normalised weights. This allows for particles that have a better guess of \( x_t \) and higher weights to get replicated while particles with lower weights die off. After resampling the normalised and unnormalised weights are set to \( \frac{1}{N} \sum_{i=1}^{N} w_{1:t}^{(\theta,i)} \) and \( 1/N \), respectively.

D. Calculating Likelihood and Gradients

In this section we outline how to calculate the likelihood and the gradient of the likelihood w.r.t \( \theta \).
Therefore in this paper we will only describe how to calculate \( \nabla \). We assume \( \nabla \) where

\[
\log \pi \theta \text{ mined by a user-defined step-size. If } \theta \text{ rarely gets accepted while having too small a step-size may result in } \pi(\theta) \text{ being poorly explored.}
\]

This problem is accentuated if \( \theta \) is high dimensional. The work done in [26] shows how to differentiate the particle filter so gradients of the log-posterior of \( \theta \) can be used to efficiently propose \( \theta' \).

We present the gradient of the log-posterior of \( \theta \) as

\[
\nabla \log p(\theta|y_{1:t}) = \nabla \log p(\theta) + \nabla \log p(y_{1:t}|\theta),
\]

where \( \nabla \log p(\theta) \) is the gradient of the log-prior and \( \nabla \log p(y_{1:t}|\theta) \) is the gradient of the log-likelihood\(^1\). Note that we assume \( \nabla \log p(\theta) \) can be calculated explicitly.

Differentiating the weights gives an approximation to the gradient of the likelihood:

\[
\frac{d}{d\theta} \log p(y_{1:t}|\theta) = \frac{1}{N} \sum_{i=1}^{N} \frac{d}{d\theta} w_{1:t}^{(\theta,i)}.
\]

Applying the Chain Rule to (13) and (18) gives

\[
\frac{d}{d\theta} \log p(y_{1:t}|\theta) = \frac{1}{p(y_{1:t}|\theta)} \sum_{i=1}^{N} w_{1:t}^{(\theta,i)} \frac{d}{d\theta} \log w_{1:t}^{(\theta,i)}
\]

\[
\approx \sum_{i=1}^{N} w_{1:t}^{(\theta,i)} \frac{d}{d\theta} \log w_{1:t}^{(\theta,i)}
\]

In the general case we need to calculate the derivatives of the dynamics, \( \frac{d}{d\theta} p(x_{t}^{(\theta,i)}|x_{t-1}^{(\theta,i)}, \theta) \), likelihood, \( \frac{d}{d\theta} p(y_{t}|x_{t}^{(\theta,i)}) \), and proposal, \( \frac{d}{d\theta} q(x_{t}^{(\theta,i)}|x_{t-1}^{(\theta,i)}, y_{t}) \) which are described in detail in [26]. However, similarly to [10], because we set the proposal to equal the dynamic model, the derivatives of the weights simplifies to

\[
\frac{d}{d\theta} w_{1:t}^{(\theta,i)} = \frac{d}{d\theta} p(y_{t}|x_{t}^{(\theta,i)}) w_{1:t-1}^{(\theta,i)} + p(y_{t}|x_{t}^{(\theta,i)}) \frac{d}{d\theta} w_{1:t-1}^{(\theta,i)}.
\]

Therefore in this paper we will only describe how to calculate (21).

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]
simulate (29) and (30) for \( T \) prior choices as described in [8] for the SIR model III-A. We p-MCMC by running the same simulation with parameter and the infected to 10 which results in the fractions of 4990/5000 people that are in the susceptible compartment to 4990 and \( t = 5000 \). Results where \( b \). 

### A. SIR Model

\[
\begin{align*}
    s_{t+1} &= s_t - \beta i_t s_t + \epsilon_\beta, \\
    e_{t+1} &= e_t + \beta i_t s_t - \delta e_t - \epsilon_\beta + \epsilon_\delta, \\
    i_{t+1} &= i_t + \delta e_t - \gamma i_t + \epsilon_\gamma - \epsilon_i, \\
    r_{t+1} &= r_t - \gamma i_t + \epsilon_\gamma.
\end{align*}
\]

(29) (30) (31)

\[
\begin{tikzpicture}
  
  \node (S) at (0,0) {$S$};
  \node (I) at (1,0) {$I$};
  \node (R) at (2,0) {$R$};
  \path[->] (S) edge node {$1 \cdot \frac{S}{N} \cdot \beta I$} (I);
  \path[->] (I) edge node {$\gamma \cdot 1 \cdot I$} (R);
  \end{tikzpicture}
\]

Fig. 1. A graphical representation of the SIR transmission model.

### B. SEIR Model

\[
\begin{align*}
    s_{t+1} &= s_t - \beta i_t s_t + \epsilon_\beta, \\
    e_{t+1} &= e_t + \beta i_t s_t - \delta e_t - \epsilon_\beta + \epsilon_\delta, \\
    i_{t+1} &= i_t + \delta e_t - \gamma i_t + \epsilon_\gamma - \epsilon_i, \\
    r_{t+1} &= r_t - \gamma i_t + \epsilon_\gamma.
\end{align*}
\]

(32) (33) (34) (35)

\[
\begin{tikzpicture}
  
  \node (S) at (0,0) {$S$};
  \node (E) at (1,0) {$E$};
  \node (I) at (2,0) {$I$};
  \node (R) at (3,0) {$R$};
  \path[->] (S) edge node {$1 \cdot \frac{S}{N} \cdot \beta I$} (E);
  \path[->] (E) edge node {$\delta \cdot 1 \cdot E$} (I);
  \path[->] (I) edge node {$\gamma \cdot 1 \cdot I$} (R);
  \end{tikzpicture}
\]

Fig. 2. (a) Simulated epidemic curves for the SIR model in section III-A with \( \theta = [\beta, \gamma, \phi] \) set to 0.254, 0.111 and 1.246, respectively. (b) Syndromic data simulated from the Infected curve. The Log-likelihood and the gradient of the log-likelihood across a range of 100 values of \( \beta \) equally spaced from 0.248 and 0.260 can be seen in (c) and (d), respectively. The black dashed vertical lines indicate the true value of beta and the horizontal line indicates a gradient of 0.

### C. Observation Equation

We model the log of the observations by

\[
\log y_t \sim \mathcal{N} \left( b_i \phi^t, \sigma_i^2 \right),
\]

where \( b_i \), \( \phi_i \) and \( \sigma_i \) are all non-negative constants. For the purpose of this analysis \( b_i \), \( \phi_i \) and \( \sigma_i \) are known which allows us to formulate the likelihood as

\[
p \left( y_t | x_t^{(\theta, i)} \right) = \log \mathcal{N} \left( y_t; b_i \phi^t, \sigma_i^2 \right).
\]

(36) (37)

\[
\begin{tikzpicture}
  
  \node (S) at (0,0) {$S$};
  \node (E) at (1,0) {$E$};
  \node (I) at (2,0) {$I$};
  \node (R) at (3,0) {$R$};
  \path[->] (S) edge node {$1 \cdot \frac{S}{N} \cdot \beta I$} (E);
  \path[->] (E) edge node {$\delta \cdot 1 \cdot E$} (I);
  \path[->] (I) edge node {$\gamma \cdot 1 \cdot I$} (R);
  \end{tikzpicture}
\]

Fig. 3. A graphical representation of the SEIR transmission model.

### D. Results

1) SIR: Firstly we compared a NUTS and MH proposal in p-MCMC by running the same simulation with parameter and prior choices as described in [8] for the SIR model III-A. We simulate (29) and (30) for \( T = 125 \) days with a population of \( P = 5000 \). The true values of \( \theta = [\beta, \gamma, \phi] \) are set to 0.254, 0.111 and 1.246, respectively. At \( t_0 \) we set the number of people that are in the susceptible compartment to 4990 and the infected to 10 which results in the fractions of 4990/5000 and 10/5000, respectively. Figure 2(a) shows the percentage of the population in each compartment throughout the duration of the epidemic. Figure 2(b) shows the simulated syndromic observations, \( y_{1:T} \), which are a fraction of the infected compartment. They are drawn from (36) with parameters \( b_i \), \( \phi_i \) and \( \sigma_i \) set to 0.25, 1.07 and 0.0012, respectively.

We run the particle filter to obtain estimates of the log-likelihood and the gradient of the log-likelihood w.r.t \( \beta \) across a range of 100 values of \( \beta \) equally spaced from 0.248 and 0.260. The aim is to maximise the log-likelihood and have a gradient of 0 when \( \beta = 0.254 \). The results can be seen in Figure 2(c) and (d). The Mean Squared Error (MSE) between

### Table I

| Parameter     | Description                                      | Prior Information               |
|---------------|--------------------------------------------------|---------------------------------|
| \( P \)       | The total population                             | -                               |
| \( S(t) \)    | The proportion of people in the susceptible compartment | \( 1 - (E_0 + I_0) \)           |
| \( E(t) \)    | The proportion of people in the exposed compartment | \( \text{Uniform}(0.00016, 0.000024) \) |
| \( I(t) \)    | The proportion of people in the infectious compartment | \( \text{Uniform}(0.00016, 0.000024) \) |
| \( R(t) \)    | The proportion of people in the recovered compartment | -                               |
| \( \beta \)   | Mean rate of people an infected person infects per day | HalfNormal \((0.0, 0.5)\)        |
| \( \gamma \)  | The proportion of infected recovering per day     | Normal \((4.0, 5.0)\)           |
| \( \delta \)  | Length of incubation period                      | Normal \((4.0, 5.0)\)           |
| \( R_0 \)     | The total number of people an infected person infects | -                               |
the true and inferred values of $\theta$ for both MH and NUTS as well as the time taken in seconds when using different numbers of particles, $N$, can be seen in Table III. The MSE and computational time are averaged over 10 runs, which use different random number seeds for 50 MCMC iterations. Using 10 different random number seeds will produce different realisations of when resampling occurs (as explained in Section II-D), so in turn will converge to the true values of $\theta$ at different rates. Averaging over the 10 runs gives a more accurate representation of the typical convergence accuracy and the time taken. This is exemplified in Figure 4 when using M-H (subplots: (a) and (b)) and NUTS (subplots: (c) and (d)) for the SIR model. Each line is an independent traceplot of an MCMC chain with the same initial starting values of 0.15 and 0.21 for $\beta$ and $\gamma$, respectively. Note for the MH proposal we use the same step-sizes for $\beta$ and $\gamma$ as are stated in [8] of 0.005 and 0.001, respectively. It is evident when looking at Table III that using NUTS can minimise the MSE in less computation time when compared with MH.

2) SEIR: Secondly we show how well the Markov-chain converges to the true parameters when running with different numbers of particles, $N$. Table III outlines the results for the SEIR model in section III-A when using NUTS with $\Delta$ set to 0.0055. We run the simulation for 2000 samples with the first 1000 discarded as burn-in. The true values of $\beta$, $\gamma$ and $\delta$ set to 0.254, 0.111 and 0.4, respectively. Figure 5(a) and (b) shows the trace plots for $\beta$ when $N = 16$ and $N = 256$, respectively. By eye it is evident that using $N = 256$ results in a Markov-chain that has better mixing which in turn explores $\pi(\theta)$ more efficiently than having $N = 16$. This is backed up by looking at the acceptance rate in Table III with $N = 256$ resulting in 0.53 of samples being accepted compared with 0.20 for $N = 16$. The auto-correlation function (ACF) plots when having $N = 16$ and $N = 256$ can be seen in Figures 5(c) and (d), respectively. ACF plots show the auto-correlation between samples in the Markov-chain as a function of a user-specified lag which we select to be 100. A plot that trends towards 0 in fewer lags is seen to be better, which is the case when $N = 256$. Table III outlines the integrated auto-correlation time (IACT) for each parameter. This is a numerical measure of the area under the ACF plot and a lower value represents less correlation between consecutive samples.

The effective sample size (ESS) for different numbers of particles is also shown in Table III as well as the time taken in seconds the p-MCMC takes to run. This gives an indication of the number of independent samples it would take to have the same estimation power as a set of auto-correlated samples. The ESS time taken in seconds shows how efficient the algorithm is by calculating how many independent samples are drawn per second.

The predicted marginal distributions, when using $N = 256$ and NUTS, for the parameters $\beta$, $\gamma$ and $\delta$ can be seen in Figures 6(a) and (d). This is calculated by $R_0 = \beta/\gamma$. The predicted distribution around the the basic reproduction number, $R_0$, in Figure 6(d). This is calculated by $R_0 = \beta/\gamma$.

IV. CONCLUSIONS AND FUTURE WORK

We have shown that using NUTS as the proposal can improve accuracy when inferring the parameters of a simple SIR model over the standard MH proposal. We have also outlined multiple methods to assess convergence of MCMC chains. These are a necessity to ascertain if the results obtained from the p-MCMC algorithm are useful.

Another popular method for modelling epidemics described by Reed-Frost assumes that people becoming infected with a disease follows a Binomial distribution. We do not include

| $N$ | $\beta$ | $\gamma$ | Time (secs) |
|-----|---------|---------|-------------|
| 50  | M-H: 0.0055 0.0095  2.3020 | NUTS: 0.0010 0.0006  25.4688 |
| 100 | M-H: 0.0051 0.0095  3.9031 | NUTS: 0.0010 0.0006  31.9671 |
| 200 | M-H: 0.0054 0.0094  7.6374 | NUTS: 0.0009 0.0097  51.1340 |
| 400 | M-H: 0.0049 0.0097  17.4208 | NUTS: 0.0012 0.0008  101.5174 |
| 800 | M-H: 0.0046 0.0094  39.4373 | NUTS: - - - |
this model in our analysis due to (25) being non-differentiable when the particle state equations follow a discrete distribution. A fruitful direction for future work would involve including the ideas described in [29], which leverages optimal transport ideas to mimic the multinomial distribution, and mimic the Binomial distribution.

In this piece of work we have only considered simulated data to provide a proof of concept. A next step would be to apply these methods to more sophisticated epidemiological models, as seen in [15], and to use real data pertinent to disease outbreaks.

One last direction for future work could involve ingesting multiple data streams with different reporting latency. It has been shown in [30] that certain combinations of datasets can improve accuracy when predicting COVID-19 related deaths.

**ACKNOWLEDGMENT**

The authors would like to acknowledge Lee Devlin and the UK Health Security Agency (UKHSA) for their support.

**REFERENCES**

[1] C. C. Kerr, R. M. Stuart, D. Mistry, R. G. Abeysuriya, K. Rosenfeld, G. R. Hart, R. C. Núñez, J. A. Cohen, P. Selvaraj, B. Hagedorn, et al., “Covasim: an agent-based model of covid-19 dynamics and interventions,” PLOS Computational Biology, vol. 17, no. 7, p. e1009149, 2021.

[2] W. O. Kermack and A. G. McKendrick, “A contribution to the mathematical theory of epidemics,” Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character, vol. 115, no. 772, pp. 700–721, 1927.

[3] A. Doucet, N. De Freitas, N. J. Gordon, et al., Sequential Monte Carlo methods in practice, vol. 1. Springer, 2001.

[4] N. J. Gordon, D. J. Salmond, and A. F. Smith, “Novel approach to nonlinear/non-gaussian bayesian state estimation,” in IEE Proceedings F-radar and signal processing, vol. 140, pp. 107–113, IET, 1993.

[5] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” IEEE Transactions on signal processing, vol. 50, no. 2, pp. 174–188, 2002.

[6] V. Dukic, H. F. Lopes, and N. G. Polson, “Tracking epidemics with google flu trends data and a state-space seir model,” Journal of the American Statistical Association, vol. 107, no. 500, pp. 1410–1426, 2012.

[7] P. Dawson, R. Gailis, and A. Meehan, “Detecting disease outbreaks using a combined bayesian network and particle filter approach,” Journal of theoretical biology, vol. 370, pp. 171–183, 2015.

[8] D. M. Sheinson, J. Niemi, and W. Meiring, “Comparison of the performance of particle filter algorithms applied to tracking of a disease epidemic,” Mathematical biosciences, vol. 255, pp. 21–32, 2014.

[9] N. M. Shahtori, C. Scoglio, A. Pourhabib, and F. D. Sahneh, “Sequential monte carlo filtering estimation of ebola progression in west africa,” in 2016 American Control Conference (ACC), pp. 1277–1282, IEEE, 2016.

[10] C. M. Hazelbag, J. Dushoff, E. M. Dominic, Z. E. Mthombathi, and W. Delva, “Calibration of individual-based models to epidemiological data: A systematic review,” PLoS computational biology, vol. 16, no. 5, p. e1007893, 2020.

[11] R. M. Neal et al., “Mcmc using hamiltonian dynamics,” Handbook of markov chain monte carlo, vol. 2, no. 11, p. 2, 2011.
If

\[ N(x; \mu, C) = \exp \left( -\frac{1}{2} (x - \mu)^T C^{-1} (x - \mu) \right) \sqrt{\frac{1}{2\pi C}} \]  

(38)

then

\[ \frac{\partial}{\partial x} \log N = -C^{-1}(x - \mu) \]  

(39)

\[ \frac{\partial}{\partial \mu} \log N = C^{-1}(x - \mu) \]  

(40)

\[ \frac{\partial}{\partial C} \log N = -\frac{1}{2} (C^{-1} - C^{-1}(x - \mu)(x - \mu)^T C^{-1}) \]  

(41)

Algorithm 1 Pseudocode for p-MCMC using a MH Proposal for model [11-A] when inferring \( \beta \) and \( \gamma \)

Require: \( \theta_0 \) \{Inputs\}
1: Run Algorithm 2 with \( \theta = [\beta, \gamma] \) to obtain estimate of \( \log p(y_{1:T}|\theta) \)
2: for \( i = 1, \ldots, M \) do
3: Sample \( \theta' = [\beta', \gamma'] \) from a proposal distribution \( q(\theta'|\theta) \)
4: Run Algorithm 2 with \( \theta' \) to obtain estimates of \( \log p(y_{1:T}|\theta') \)
5: Compute acceptance probability, \( \alpha(\theta, \theta') \) using (5)
6: Sample \( u \sim U[0,1] \)
7: if \( u < \alpha(\theta, \theta') \) then
8: accept \( \theta' \), \( \log p(y_{1:T}|\theta') \)
9: else
10: accept \( \theta \), \( \log p(y_{1:T}|\theta) \)
11: end if
12: end for

Algorithm 2 Pseudocode for the Particle Filter

Require: \( \theta \), \( Y_{1:T} \), \( N \) \{Inputs\}
1: Initialise \( x_0^i, dx_0^i/d\theta, \log(w_0^i), d\log(w_0^i)/d\theta \)
2: for \( i = 1, \ldots, T \) do
3: If \( N_{\text{eff}} \leq \text{threshold} \):
4: Resample \( x_{i-1}^i, \log(w_{i-1}^i), dx_{i-1}^i/d\theta, d\log(w_{i-1}^i)/d\theta \)
5: Calculate the particle weights and derivatives of particle weights using (9) and (21), respectively.
6: end for

Algorithm 3 Pseudocode for p-MCMC using NUTS as the proposal.

Require: \( \ell(\theta), \nabla L(\theta) \) = Run Algorithm 2
1: Initialise \( x_0^i, dx_0^i/d\theta, \log(w_0^i), d\log(w_0^i)/d\theta \)
2: for \( i = 1, \ldots, M \) do
3: NUTS = Algorithm 3 in [12]
4: end for

APPENDIX A

DERIVATIVES OF MULTIVARIATE LOG NORMAL