




Adaptive Semiparametric Bayesian Differential Equations Via Sequential Monte Carlo

Shijia Wang, Shufei Ge, Renny Doig & Liangliang Wang


To cite this article: Shijia Wang, Shufei Ge, Renny Doig & Liangliang Wang (2021): Adaptive Semiparametric Bayesian Differential Equations Via Sequential Monte Carlo, Journal of Computational and Graphical Statistics, DOI: [10.1080/10618600.2021.1987252](https://doi.org/10.1080/10618600.2021.1987252)


To link to this article: <https://doi.org/10.1080/10618600.2021.1987252>

 [View supplementary material](#) 

 Published online: 18 Nov 2021.

 [Submit your article to this journal](#) 



 Article views: 116

 [View related articles](#) 

 [View Crossmark data](#) 



Adaptive Semiparametric Bayesian Differential Equations Via Sequential Monte Carlo

Shijia Wang^a , Shufei Ge^b, Renny Doig^c, and Liangliang Wang^c 

^aSchool of Statistics and Data Science, LPMC & KLMDASR, Nankai University, Tianjin, China; ^bInstitute of Mathematical Sciences, ShanghaiTech University, Shanghai, China; ^cDepartment of Statistics and Actuarial Science, Simon Fraser University, Burnaby, BC, Canada

ABSTRACT

Nonlinear differential equations (DEs) are used in a wide range of scientific problems to model complex dynamic systems. The differential equations often contain unknown parameters that are of scientific interest, which have to be estimated from noisy measurements of the dynamic system. Generally, there is no closed-form solution for nonlinear DEs, and the likelihood surface for the parameter of interest is multi-modal and very sensitive to different parameter values. We propose a Bayesian framework for nonlinear DE systems. A flexible nonparametric function is used to represent the dynamic process such that expensive numerical solvers can be avoided. A sequential Monte Carlo algorithm in the annealing framework is proposed to conduct Bayesian inference for parameters in DEs. In our numerical experiments, we use examples of ordinary differential equations and delay differential equations to demonstrate the effectiveness of the proposed algorithm. We developed an R package that is available at <https://github.com/shijaw/smcDE>. Supplementary files for this article are available online.

ARTICLE HISTORY

Received December 2019
Revised September 2021

KEYWORDS

Bayesian smoothing;
B-spline; Conditional
effective sample size; Delay
differential equation;
Ordinary differential
equation

1. Introduction

Nonlinear differential equations (e.g., nonlinear ordinary or delay differential equations (DDEs)) are commonly used in modeling dynamic systems in ecology, physics, and engineering. DDEs are described by equations $\frac{dx(t)}{dt} = g(x(t), x(t - \tau)|\theta)$, where θ is the vector of unknown parameters and τ is the time delay parameter. These are continuous-time models for interactions between variables $x(t)$ and a time delay τ . Ordinary differential equations (ODEs) are often presented by $\frac{dx(t)}{dt} = g(x(t)|\theta)$, which can be regarded as a special case of DDEs with $\tau = 0$. The form of $g(\cdot)$ is generally proposed by specialists with scientific intuition. For example, ecologists proposed the simple Lotka–Volterra model (Rosenzweig and MacArthur 1963) to understand and predict the populations of predators and prey in an ecosystem. Given a concrete form of the function $g(x(t), x(t - \tau)|\theta)$, the parameters θ and τ are unknown and need to be estimated using observations at some data points. Differential equations (DEs) are often observed with measurement error. We assume that the observed $y(t)$ is linked to $x(t)$ through an additive error model such that $y(t) = x(t) + \epsilon$, where ϵ is measurement error. The estimation of parameters in DEs is of great interest and usually requires us to solve the DEs $\frac{dx(t)}{dt} = g(x(t), x(t - \tau)|\theta)$.

Many DE systems do not admit an analytic solution. One alternative approach is to solve the DEs numerically (Butcher 2016), for example by using the Euler method (Jain 1979; Bulirsch and Stoer 1966), the exponential integrators (Hochbruck, Lubich, and Selhofer 1998; Hochbruck and Ostermann 2010), or the Runge–Kutta method (Jameson, Schmidt, and Turkel 1981; Ascher, Ruuth, and Spiteri 1997).

However, numerical DE solvers are computationally expensive, especially for DDEs. Various methods have been proposed to solve DEs more efficiently in recent decades. The idea of using smoothing splines to fit dynamic data was first proposed by Varah (1982). Ramsay and Silverman (2007), Poyton et al. (2006), and Chen and Wu (2008) extended the idea of smoothing to a two-stage approach. In the first stage, spline coefficients are optimized by minimizing the sum of the squared distances between the data and the spline functions at the observation times. In the second stage, using the estimated spline coefficients, DE parameters are optimized by minimizing the residuals of DE models. The two-stage approach may lead to inconsistent estimates (Ramsay et al. 2007). Ramsay et al. (2007) proposed a generalized smoothing approach, called “parameter cascading”, based on data smoothing methods and a generalization of profiled estimation. In the proposed approach, the spline coefficients are treated as nuisance parameters. Their method iterates between optimizing the objective function with respect to the spline coefficients given current DE parameter estimates, and optimizing the objective function with respect to the DE parameters given the estimated spline coefficients. The iteration is repeated until convergence is achieved. The parameter estimates are consistent and asymptotically normally distributed under mild conditions (Qi et al. 2010; Pang, Yan, and Zhou 2017). There are several variations of the parameter cascading approach. Cao, Wang, and Xu (2011) proposed a robust algorithm to estimate parameters using measurements with outliers based on smoothing splines. Cao, Huang, and Wu (2012) proposed a method to estimate time-varying parameters in ODEs, in which the ODE parameters are also modeled by smoothing splines. Wang and Cao (2012) developed a

semiparametric method with smoothing spline to estimate DDE parameters.

Using smoothing splines to model DEs is computationally efficient since we do not need to numerically solve DEs. Most methods based on data smoothing to estimate parameters of DEs are derived from a frequentist perspective. Bayesian methods are of interest since they quantify the uncertainty of parameters. Campbell and Steele (2012) proposed a smooth functional tempering algorithm to conduct posterior inference for ODE parameters. This idea originates from parallel tempering and model-based smoothing. Zhang et al. (2017) proposed a high-dimensional linear ODE model to accommodate the directional interaction between areas of the brain. Parallelized schemes for Markov chain Monte Carlo (MCMC) have been proposed to estimate this model. Bhaumik et al. (2015) investigated a two-stage procedure to estimate parameters by minimizing the penalized ODEs.

There are several lines of work involved in estimating DE parameters from a Bayesian perspective based on numerical DE solvers. Dass et al. (2017) proposed a two-step approach to approximate posterior distributions of parameters of interest. They first applied a numerical algorithm to solve ODEs, then integrated out nuisance parameters using Laplace approximations. Bhaumik et al. (2017) proposed a modification of Bhaumik et al. (2015) by directly considering the distance between the function in the nonparametric model and that obtained from a four-stage Runge–Kutta (RK4) method. Calderhead, Girolami, and Lawrence (2009) presented a novel Bayesian sampler to infer parameters in nonlinear DDEs; the derivatives and time delay parameters were estimated via Gaussian processes. To make the DE estimation more consistent, Dondelinger et al. (2013) proposed an adaptive gradient matching approach to jointly infer the hyperparameters of a Gaussian process as well as ODE parameters. Barber and Wang (2014) simplified previous approaches by proposing a more natural generative model via Gaussian process. The proposed approach directly links state derivative information with system observations.

Standard sequential Monte Carlo (SMC) methods (Liu and Chen 1998; Doucet, Godsill, and Andrieu 2000; Doucet, De Freitas, and Gordon 2001) are popular approaches for estimating dynamic models (e.g., state-space models). SMC methods combine importance sampling and resampling algorithms. Under mild conditions, consistency properties and asymptotic normality hold (Chopin et al. 2004). Del Moral, Doucet, and Jasra (2006) proposed a general SMC framework, to sample sequentially from a sequence of intermediate probability distributions that are defined on a common space. This general framework has promoted popularity of SMC methods in areas besides state space models. For example, Wang, Wang, and Bouchard-Côté (2020) proposed an annealed SMC algorithm for phylogenetics by designing an artificial sequence of intermediate distributions. Several SMC methods have been proposed to estimate parameters in ODE models. Zhou, Johansen, and Aston (2016) presented an adaptive SMC sampling strategy to estimate parameters and conduct model selection. They used a simple example of ODEs to demonstrate the performance of model selection using their algorithm. Lee, Lee, and Dass (2018) introduced additive Gaussian errors into the ODE trajectory provided by numerical solvers, and they proposed a particle filter to infer

ODE parameters. In addition, Gaussian processes have been used to avoid numerical integration. These works are based on numerically solving ODE models.

In this article, we propose a semiparametric Bayesian model for nonlinear DEs and design an annealed SMC algorithm to conduct inference efficiently for parameters. The DE trajectories are represented using a linear combination of basis functions. Consequently, our method avoids expensive numerical solvers, especially those for DDEs. It instead needs to estimate the basis coefficients together with other parameters in the DEs. In other words, the parameters of interest include the DE parameters, basis coefficients of smoothing spline functions, and parameters in the observation model. In addition, a tuning parameter is used to balance the fit to data and fidelity to the DEs. We estimate the tuning parameter using the Bayesian approach to avoid tuning it through expensive cross-validation. Inspired by the reference distribution of Fan et al. (2011) in the context of model selection, we design an artificial sequence of intermediate distributions that starts from a reference distribution, which is easier to sample from, and gradually approaches the target distribution through a sequence of annealing parameters. The proposed annealed SMC can effectively sample parameters with multiple isolated posterior modes and basis function coefficients of high dimensionality. It adopts the adaptive scheme in Zhou, Johansen, and Aston (2016) and Wang, Wang, and Bouchard-Côté (2020) to choose the sequence of annealing parameters that determines the intermediate target distributions of SMC. Our numerical experiments demonstrate the effectiveness of our algorithm in estimating parameters and DE trajectories for both ODEs and DDEs.

The rest of article is organized as follows. In Section 2, we construct a fully Bayesian framework for nonlinear DEs. In Section 3, we introduce our new algorithm for Bayesian inference for nonlinear DEs. In Sections 4 and 5, we use a real data analysis and numerical experiments to show the effectiveness of our method. We conclude in Section 6.

2. Hierarchical Bayesian Differential Equations

In this section, we introduce a hierarchical Bayesian structure for DE models. In Section 2.1, we introduce the likelihood function for DEs. In Section 2.2, we construct a Bayesian model for the DE model. In Section 2.3, we introduce selection of the tuning parameter λ . In Section 2.4, we introduce the posterior distribution of the DE model.

2.1. DE Models

We use $\mathbf{x}(t) = (x_1(t), \dots, x_I(t))'$ to denote the DE variables (i.e., the solution of a DE system), where $x_i(t)$ denotes the i th DE variable and I denotes the total number of DE variables. Each DE variable $x_i(t)$, $i = 1, \dots, I$, is a dynamic process modeled with one differential equation

$$\begin{aligned} \frac{dx_i(t)}{dt} &= g_i(\mathbf{x}(t), \mathbf{x}(t - \tau) | \boldsymbol{\theta}), \\ x_i(0) &= x_{i0}, \end{aligned} \quad (1)$$

where $t \in [t_1, t_{\max}]$, $t_1 = 0$ unless it is specified otherwise, $\boldsymbol{\theta}$ denotes the vector of unknown parameters in the DE model, τ is

the delay parameter in DDE model ($\tau = 0$ in ODE model), and $x_i(0)$ is the initial condition for the i th DE variable, which is also unknown and needs to be estimated. DDEs are time-delayed systems. The time delay τ in DDEs considers the dependence of the present state of the DE variable based on its past state. We refer readers to Section 4 for a more detailed description of DDE models.

We do not observe the DEs directly, instead we observe them with measurement errors. In addition, we often only observe a subset of the I DE variables, $\mathcal{I}_0 \subseteq \{1, \dots, I\}$. We let $\mathbf{y}_i = (y_{i1}, \dots, y_{ij_i})'$ denote the observations for the i th DE trajectory. The j th observation of \mathbf{y}_i is assumed to be normally distributed with mean $x_i(t_{ij}|\boldsymbol{\theta}, \tau, x_{i0})$ and variance σ_i^2 ,

$$y_{ij} \sim N(x_i(t_{ij}|\boldsymbol{\theta}, \tau, x_{i0}), \sigma_i^2), j = 1, \dots, J_i,$$

where $x_i(t_{ij}|\boldsymbol{\theta}, \tau, x_{i0})$ denotes the DE solution given $\boldsymbol{\theta}$, τ , and initial condition x_{i0} ; t_{ij} denotes the time we observe the j th observation of \mathbf{y}_i .

The joint likelihood function of $\boldsymbol{\theta}$, τ , $\mathbf{x}(0)$, and σ_i^2 admits the following form:

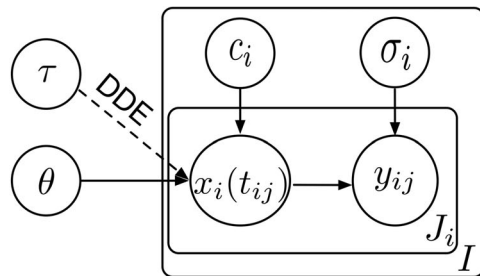
$$L(\boldsymbol{\theta}, \tau, \mathbf{x}(0), \sigma_i^2) = \prod_{i \in \mathcal{I}_0} \prod_{j=1}^{J_i} (\sigma_i^2)^{-1/2} \exp \left\{ - \frac{(y_{ij} - x_i(t_{ij}|\boldsymbol{\theta}, \tau, \mathbf{x}(0)))^2}{2\sigma_i^2} \right\}. \quad (2)$$

We use a figure (see Figure 1 (b)) to show an example of the log-likelihood surface over the DE parameters $\boldsymbol{\theta}$, and for the setup of this model we refer to Section 5.1. The log-likelihood surface for $\boldsymbol{\theta}$ has multiple isolated modes, and it is very sensitive to different parameter values.

2.2. A Bayesian Structure for DE Model

Numerically solving DEs can be computationally extremely intensive, especially for DDE models. We propose to solve differential equations by penalized smoothing. More specifically, we represent the i th DE function $x_i(t)$ as a linear combination of L_i B-spline basis functions $\boldsymbol{\Phi}_i(t) = (\phi_{i1}(t), \phi_{i2}(t), \dots, \phi_{iL_i}(t))'$,

$$x_i(t) = \boldsymbol{\Phi}_i(t)' \mathbf{c}_i,$$



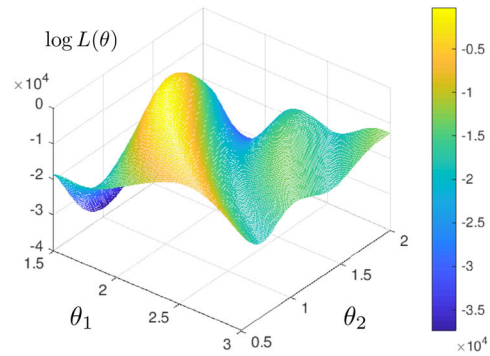
(a)

where \mathbf{c}_i denotes the vector of basis coefficients. See Figure 1 in the supplementary material for an example of cubic B-spline functions (Ramsay 2006; De Boor 1972). Here, we do not distinguish the true $x_i(t)$ from its approximation using splines, which is a common practice in nonparametric smoothing (Berry, Carroll, and Ruppert 2002; Ramsay 2006; Wood 2017). The reason is that the number of basis functions is chosen to be sufficiently large such that we expect the basis function approximation can avoid bias from model over-simplification. The error in approximating $x_i(t)$ by its B-spline approximation typically is negligible compared to the estimation error, so we can assume that the two are equivalent (see Berry, Carroll, and Ruppert 2002, p. 161). The initial condition for the i th DE function is $x_i(0) = \boldsymbol{\Phi}_i(0)' \mathbf{c}_i$. One advantage of using smoothing spline functions to model DE trajectories is that we can avoid explicitly estimating the initial condition $\mathbf{x}(0)$; instead, it is estimated using $\hat{x}_i(0) = \boldsymbol{\Phi}_i(0)' \hat{\mathbf{c}}_i$, where $\hat{\mathbf{c}}_i$ is the vector of estimated basis coefficients. Figure 1(a) represents the graphical structure for the proposed DE model. The unknown parameters in our DE model include spline coefficients \mathbf{c}_i , the delay time parameter τ (which is known in an ODE with $\tau = 0$), the DE parameter $\boldsymbol{\theta}$, and variance parameter σ_i^2 .

With the basis function representation, finding the DE solution becomes a problem of estimating the basis function coefficients, $\mathbf{c} = (\mathbf{c}'_1, \mathbf{c}'_2, \dots, \mathbf{c}'_I)'$. In the Bayesian framework, we specify a prior distribution for \mathbf{c} conditional on the DE parameters $\boldsymbol{\theta}$, τ , and a smoothing parameter λ , as follows:

$$\begin{aligned} \tilde{\pi}_0(\mathbf{c}|\boldsymbol{\theta}, \tau, \lambda) & \propto \exp \left\{ - \frac{\lambda}{2} \sum_{i=1}^I \int_{t_1+\tau}^{t_{\max}} \left[\frac{dx_i(s)}{ds} - g_i(\mathbf{x}(s), \mathbf{x}(s-\tau)|\boldsymbol{\theta}) \right]^2 ds \right\}, \\ & = \exp \left\{ - \frac{\lambda}{2} \sum_{i=1}^I \int_{t_1+\tau}^{t_{\max}} \left[\frac{d\boldsymbol{\Phi}_i(s)'}{ds} \mathbf{c}_i - g_i(\boldsymbol{\Phi}(s)' \mathbf{c}, \boldsymbol{\Phi}(s-\tau)' \mathbf{c}|\boldsymbol{\theta}) \right]^2 ds \right\}, \end{aligned} \quad (3)$$

where $\boldsymbol{\Phi}(s) = \text{Diag}(\boldsymbol{\Phi}_1(s), \boldsymbol{\Phi}_2(s), \dots, \boldsymbol{\Phi}_I(s))$ is a $\sum_i L_i \times I$ -dimensional block diagonal matrix such that the i -th column contains $\boldsymbol{\Phi}_i(x)$ in the appropriate diagonal block and 0 elsewhere. This prior distribution measures how well the estimated DE variables $\mathbf{x}(t)$ satisfy the DE system defined on $[t_1, t_{\max}]$. It is based on treating a penalty term proposed in Ramsay et al. (2007) as a prior similar to how Berry, Carroll, and Ruppert



(b)

Figure 1. (a) Graphical representation of DEs; (b) log-likelihood surface for a DE model.

(2002) incorporated a penalty as a prior. The smoothing parameter λ controls the tradeoff between fit to the data and fidelity to the DE model. Details on selecting a proper λ will be discussed in Section 2.3.

In the Bayesian framework, we need to assign appropriate priors for model parameters θ , τ , σ_i^2 , $i \in \mathcal{I}_0$. The following priors are specified:

$$\theta \sim \text{MVN}(\mathbf{0}_D, \sigma_\theta^2 \mathbf{I}_D), \quad (4)$$

$$\tau \sim \text{Unif}(t_1, t_{\max}), \quad (5)$$

$$\sigma_i^2 \sim \text{IG}(g_0, h_0), \quad i \in \mathcal{I}_0, \quad (6)$$

where σ_θ^2 , g_0 , and h_0 are the hyperparameters in prior distributions, and D is the dimension of the vector θ . The vector of all zeros is represented by $\mathbf{0}$, and \mathbf{I} is an identity matrix. Their subscripts denote the vector/matrix dimension.

2.3. The Choice of λ

The tuning parameter λ is important in balancing between fit to the data and fidelity to the DE model. A small value of λ does not impose much information about the DE fitting. If $\lambda \rightarrow 0$, then we end up fitting least squares for spline coefficients with the data. If we choose a large value of λ , then the prior information of the DE system is too strong and not much information about the data is taken into consideration. Hence, it is crucial to choose a proper value of λ to balance the DE fitting and data information.

One approach to choose λ is through cross-validation (Wang and Cao 2012; Reiss and Todd Ogden 2009) from a range of reasonable choices of λ . However, this approach is infeasible in a Bayesian framework as it significantly increases the computational cost. We propose to treat λ as an unknown parameter by specifying a prior distribution on λ and estimating its posterior distribution through a Bayesian method. This idea is adapted from Berry, Carroll, and Ruppert (2002), in which they automatically select a smoothing parameter for splines. We choose the prior distribution for the smoothing parameter to be $\text{Gamma}(a_\lambda, b_\lambda)$.

2.4. Posterior Distribution of DE Model

The likelihood function is

$$p(\mathbf{y}|\tau, \theta, \mathbf{c}, \sigma) \propto \left(\prod_{i \in \mathcal{I}_0} \prod_{j=1}^{J_i} \sigma_i^2 \right)^{-1/2} \exp \left\{ - \sum_{i \in \mathcal{I}_0} \left(\sum_{j=1}^{J_i} \frac{(y_{ij} - \Phi_i(t_{ij})' \mathbf{c}_i)^2}{2\sigma_i^2} \right) \right\}.$$

We introduce a new notation $\beta = (\theta', \tau, \mathbf{c}', \sigma', \lambda)'$ to denote all the parameters of interest. Let $\tilde{\pi}_0(\beta)$ denote the prior distribution, which is specified in Equations (3)–(6), and $\text{Gamma}(a_\lambda, b_\lambda)$ for λ . We are interested in the posterior distribution for β

$$\pi(\beta) \propto \gamma(\beta) = p(\mathbf{y}|\tau, \theta, \mathbf{c}, \sigma) \tilde{\pi}_0(\mathbf{c}|\theta, \tau, \lambda) \tilde{\pi}_0(\theta) \tilde{\pi}_0(\tau) \tilde{\pi}_0(\sigma) \tilde{\pi}_0(\lambda).$$

Here $\gamma(\beta) = \tilde{\pi}_0(\beta) p(\mathbf{y}|\beta)$ is the unnormalized posterior distribution of β and can be written as follows:

$$\begin{aligned} \gamma(\beta) &\propto \left(\prod_{i \in \mathcal{I}_0} \prod_{j=1}^{J_i} \sigma_i^2 \right)^{-1/2} \\ &\exp \left\{ - \sum_{i \in \mathcal{I}_0} \sum_{j=1}^{J_i} \frac{(y_{ij} - \Phi_i(t_{ij})' \mathbf{c}_i)^2}{2\sigma_i^2} \right. \\ &\left. - \frac{\lambda}{2} \sum_{i=1}^I \int_{t_1+\tau}^{t_{\max}} \left[\frac{d\Phi_i(s)'}{ds} \mathbf{c}_i - g_i(\Phi(s)' \mathbf{c}, \Phi(s-\tau)' \mathbf{c}|\theta) \right]^2 ds \right\} \\ &\cdot \left(\prod_{i \in \mathcal{I}_0} \sigma_i^2 \right)^{-g_0-1} \exp \left\{ - \sum_{i \in \mathcal{I}_0} \frac{h_0}{\sigma_i^2} \right\} \\ &\cdot \exp \left\{ - \frac{\theta' \theta}{\sigma_\theta^2} \right\} \cdot \lambda^{a_\lambda-1} \exp(-b_\lambda \lambda). \end{aligned}$$

The integral

$$\mathbf{R}_i = \int_{t_1+\tau}^{t_{\max}} \left[\frac{d\Phi_i(s)'}{ds} \mathbf{c}_i - g_i(\Phi(s)' \mathbf{c}, \Phi(s-\tau)' \mathbf{c}|\theta) \right]^2 ds$$

usually does not have a closed-form expression. However, it can be evaluated by numerical quadrature approximation. Let $\eta_{i0} = t_1 + \tau < \eta_{i1} < \eta_{i2} < \dots < \eta_{i\zeta_i} < t_{\max} = \eta_{i(\zeta_i+1)}$ denote the knots placed within $[t_1 + \tau, t_{\max}]$ for i -th DE function, we approximate the integral by using the composite Simpson's rule (Burden, Faires, and Reynolds 2011)

$$\mathbf{R}_i \approx \sum_{l_i=0}^{\zeta_i} \sum_{m=1}^M v_{l_i m} \cdot \left(\left[\frac{d\Phi_i(s)'}{ds} \mathbf{c}_i - g_i(\Phi(s)' \mathbf{c}, \Phi(s-\tau)' \mathbf{c}|\theta) \right]^2 \Big|_{s=\xi_{i,l_i m}} \right),$$

where M is the number of quadrature points, ζ_i is the number of knots used for the i th DE function, $\xi_{i,l_i m}$ is the m th quadrature point in $[\eta_{i,l_i}, \eta_{i(l_i+1)}]$, and $v_{l_i m}$ is the corresponding quadrature weight.

3. Methodology

One classical methodology for Bayesian inference of nonlinear DE parameters is MCMC. In MCMC, we construct an ergodic Markov chain which admits the normalized posterior as its stationary distribution. If we run the chain long enough, convergence to the posterior is guaranteed. We show the details of this method in Supplementary Section 2.1.

However, MCMC (more specifically, the Metropolis–Hastings (MH) algorithm) is inefficient for estimating parameters of nonlinear DEs for several reasons. First, the posterior surface is extremely sensitive to DE parameters θ . There may exist isolated modes in the posterior distribution. The posterior may change quite a bit even with a tiny change in the parameter value. Second, the computation of the likelihood function involves numerically solving nonlinear DEs, which is computationally expensive. Third, the convergence of MCMC is generally difficult to assess.

3.1. An Annealed Sequential Monte Carlo for Bayesian DE Inference

To better cope with the inadequateness of MCMC, we propose a sequential Monte Carlo (SMC) algorithm in the SMC framework of Del Moral, Doucet, and Jasra (2006) for the *static* setting for Bayesian DEs. This special case of SMC is a generic method to approximate a sequence of intermediate probability distributions $\{\pi_r(\boldsymbol{\beta})\}_{0 \leq r \leq R}$ defined on a common measurable space (E, \mathcal{E}) . This method is different from the standard SMC algorithm (Doucet, Godsill, and Andrieu 2000; Doucet, De Freitas, and Gordon 2001), as the sequence of intermediate probability distributions $\{\pi_r(\boldsymbol{\beta})\}_{0 \leq r \leq R}$ in standard SMC methods are generally defined on measurable spaces with increasing dimension.

The SMC algorithm in the static setting approximates the target distribution $\pi(\boldsymbol{\beta})$ in R steps. We are interested in sequentially sampling from the distributions $\{\pi_r(\boldsymbol{\beta})\}_{0 \leq r \leq R}$. For example, we first approximate $\pi_0(\boldsymbol{\beta})$, then approximate $\pi_1(\boldsymbol{\beta})$ and so on. The subscript r denotes the index of intermediate probability distributions. The last intermediate target distribution $\pi_R(\boldsymbol{\beta})$ is $\pi(\boldsymbol{\beta})$. We let $\pi_r(\boldsymbol{\beta}) \propto \gamma_r(\boldsymbol{\beta})$, where γ_r can be evaluated pointwise. There are several reasons to use multiple distributions in SMC. First, in online problems, data arrive sequentially and we aim to do inference sequentially in time. The distribution $\pi_r(\boldsymbol{\beta})$ is then the unnormalized posterior distribution conditioning on the first r batches of data. Second, as in this work, a sequence of intermediate target distributions is introduced to facilitate the exploration of the state space. At each step r , we use a collection of K samples to represent $\pi_r(\boldsymbol{\beta})$, denoted by $\{\boldsymbol{\beta}_r^{(k)}\}_{k=1}^K$. Each of these K samples is called a particle. There is a positive weight associated with each particle $\boldsymbol{\beta}_r^{(k)}$. We use $w_r^{(k)}$ to denote the unnormalized weight of $\boldsymbol{\beta}_r^{(k)}$ and use $W_r^{(k)}$ to be the corresponding normalized weight. From iteration r to $r+1$, we move particles from $\{\boldsymbol{\beta}_r^{(k)}\}_{k=1}^K$ to $\{\boldsymbol{\beta}_{r+1}^{(k)}\}_{k=1}^K$ using a Markov kernel denoted $T_{r+1}(\boldsymbol{\beta}_r^{(k)}, \boldsymbol{\beta}_{r+1}^{(k)})$. One typical approach in the SMC framework for the static setting is to select $T_{r+1}(\boldsymbol{\beta}_r^{(k)}, \boldsymbol{\beta}_{r+1}^{(k)})$ to be a π_{r+1} -invariant MCMC kernel; this will be detailed later in the article. Then we compensate for the difference between the particles $\boldsymbol{\beta}_{r+1}^{(k)}$ proposed from $\{T_{r+1}(\boldsymbol{\beta}_r^{(k)}, \boldsymbol{\beta}_{r+1}^{(k)})\}_{k=1}^K$ and $\pi_r(\boldsymbol{\beta})$ by the updated weights $W_{r+1}^{(k)}$. To get $W_{r+1}^{(k)}$, we first compute the incremental importance weight

$$\tilde{w}_{r+1}^{(k)} = \frac{\gamma_{r+1}(\boldsymbol{\beta}_{r+1}^{(k)}) L_r(\boldsymbol{\beta}_{r+1}^{(k)}, \boldsymbol{\beta}_r^{(k)})}{\gamma_r(\boldsymbol{\beta}_r^{(k)}) T_{r+1}(\boldsymbol{\beta}_r^{(k)}, \boldsymbol{\beta}_{r+1}^{(k)})},$$

where $L_r(\boldsymbol{\beta}_{r+1}^{(k)}, \boldsymbol{\beta}_r^{(k)})$ is an artificial backward kernel (Del Moral, Doucet, and Jasra 2006, 2012), denoting the probability of moving from $\boldsymbol{\beta}_{r+1}^{(k)}$ to $\boldsymbol{\beta}_r^{(k)}$. Then we calculate the unnormalized weight by using the previous unnormalized weight and the incremental importance weight as follows:

$$w_{r+1}^{(k)} = w_r^{(k)} \cdot \tilde{w}_{r+1}^{(k)}.$$

The normalized weights $W_{r+1}^{(k)}$ are obtained by $W_{r+1}^{(k)} = w_{r+1}^{(k)} / (\sum_{k=1}^K w_{r+1}^{(k)})$.

The selection of the backward kernel $L_r(\boldsymbol{\beta}_{r+1}^{(k)}, \boldsymbol{\beta}_r^{(k)})$ is important as it will impact the variance of $\{W_{r+1}^{(k)}\}_{k=1}^K$. We refer readers

to Del Moral, Doucet, and Jasra (2006) for a more detailed discussion of this artificial backward kernel. A convenient backward Markov kernel that allows an easy evaluation of the importance weight is

$$L_r(\boldsymbol{\beta}_{r+1}^{(k)}, \boldsymbol{\beta}_r^{(k)}) = \frac{\pi_{r+1}(\boldsymbol{\beta}_r^{(k)}) T_{r+1}(\boldsymbol{\beta}_r^{(k)}, \boldsymbol{\beta}_{r+1}^{(k)})}{\pi_{r+1}(\boldsymbol{\beta}_{r+1}^{(k)})}.$$

With this backward kernel, the weight update function $\tilde{w}_{r+1}^{(k)}$ becomes

$$\begin{aligned} \tilde{w}_{r+1}^{(k)} &= \frac{\gamma_{r+1}(\boldsymbol{\beta}_{r+1}^{(k)}) L_r(\boldsymbol{\beta}_{r+1}^{(k)}, \boldsymbol{\beta}_r^{(k)})}{\gamma_r(\boldsymbol{\beta}_r^{(k)}) T_{r+1}(\boldsymbol{\beta}_r^{(k)}, \boldsymbol{\beta}_{r+1}^{(k)})} \\ &= \frac{\gamma_{r+1}(\boldsymbol{\beta}_{r+1}^{(k)})}{\gamma_r(\boldsymbol{\beta}_r^{(k)})} \cdot \frac{\pi_{r+1}(\boldsymbol{\beta}_r^{(k)}) T_{r+1}(\boldsymbol{\beta}_r^{(k)}, \boldsymbol{\beta}_{r+1}^{(k)})}{\pi_{r+1}(\boldsymbol{\beta}_{r+1}^{(k)})} \\ &= \frac{1}{T_{r+1}(\boldsymbol{\beta}_r^{(k)}, \boldsymbol{\beta}_{r+1}^{(k)})} \\ &= \frac{\gamma_{r+1}(\boldsymbol{\beta}_r^{(k)})}{\gamma_r(\boldsymbol{\beta}_r^{(k)})}. \end{aligned}$$

Thus, we do not require pointwise evaluation of the forward kernel $T_{r+1}(\boldsymbol{\beta}_r^{(k)}, \boldsymbol{\beta}_{r+1}^{(k)})$ and the backward kernel $L_r(\boldsymbol{\beta}_{r+1}^{(k)}, \boldsymbol{\beta}_r^{(k)})$ to compute the weight update function.

In this article, we propose a sequence of annealing intermediate target distributions (Neal 2001; Wang, Wang, and Bouchard-Côté 2020) $\{\pi_r(\boldsymbol{\beta})\}_{0 \leq r \leq R}$ to facilitate the exploration of posterior space, such that

$$\pi_r(\boldsymbol{\beta}) \propto \gamma_r(\boldsymbol{\beta}) = [p(\mathbf{y}|\boldsymbol{\beta}) \tilde{\pi}_0(\boldsymbol{\beta})]^{\alpha_r} \rho(\boldsymbol{\beta})^{1-\alpha_r},$$

where $\rho(\boldsymbol{\beta})$ is a reference distribution (Fan et al. 2011), and $0 = \alpha_0 < \alpha_1 < \dots < \alpha_{R-1} < \alpha_R = 1$ is the sequence of annealing parameters. When $\alpha_0 = 0$, the first distribution is the reference distribution $\rho(\boldsymbol{\beta})$; when $\alpha_R = 1$, the last distribution is our target distribution, the posterior distribution of $\boldsymbol{\beta}$.

The reference distributions should be easy to sample from and ideally they are close to the modes of the target distribution. Since we can easily sample from all the prior distributions except for the prior of \mathbf{c} , we use the same reference distribution as the prior distribution for all parameters except for \mathbf{c} .

In this case,

$$\pi_r(\boldsymbol{\beta}) \propto \gamma_r(\boldsymbol{\beta}) = [p(\mathbf{y}|\boldsymbol{\beta}) \tilde{\pi}_0(\mathbf{c}|\boldsymbol{\theta}, \tau, \lambda)]^{\alpha_r} \rho(\mathbf{c})^{1-\alpha_r} \tilde{\pi}_0(\boldsymbol{\theta}) \tilde{\pi}_0(\tau) \tilde{\pi}_0(\lambda).$$

We specify the following reference distribution for \mathbf{c} based on its MLE:

$$\mathbf{c}_i \sim \text{MVN}(\hat{\mathbf{c}}_i, \sigma_c^2 \mathbf{I}_{L_i}), \quad i = 1, \dots, I, \quad (7)$$

where $\hat{\mathbf{c}}_i$ is the MLE of \mathbf{c}_i by maximizing $p(\mathbf{y}|\boldsymbol{\beta}) \tilde{\pi}_0(\mathbf{c}|\boldsymbol{\theta}, \tau, \lambda)$ with respect to $\tau, \boldsymbol{\theta}, \mathbf{c}$, and σ_c^2 is the hyper-parameter in the reference distribution.

If there are isolated modes in $\pi(\boldsymbol{\beta})$, then MCMC may get stuck in one of the modes which is close to the initial value. A sequence of intermediate distributions is introduced to avoid this. With a small annealing parameter α_r , the intermediate distribution surface is flat, which makes MCMC samples move easily between modes. The intermediate distribution with a

higher value of annealing parameter is closer to the true posterior. The samples move closer to the target posterior distribution if we increase α_r . One simple choice of annealing parameters is to equally put parameters across $[0, 1]$, such that $\alpha_0 = 0$, $\alpha_1 = 1/R$, $\alpha_2 = 2/R$, \dots , $\alpha_{R-1} = (R-1)/R$, $\alpha_R = 1$.

We now introduce an SMC algorithm with a defined sequence of intermediate targets. First, we initialize particles $\{\beta_0^{(k)}\}_{k=1}^K$. At each step $r-1$, we keep a list of K particles $\{\beta_{r-1}^{(k)}\}_{k=1}^K$ in memory. We let $\{\tilde{\beta}_{r-1}^{(k)}\}_{k=1}^K$ denote particles after the resampling step (see Step 3). We iterate between the following three steps to obtain the approximated intermediate distributions

$$\hat{\pi}_r(\beta) = \sum_{k=1}^K W_r^{(k)} \cdot \delta_{\beta_r^{(k)}}(\beta), \quad (r = 1, \dots, R).$$

Step 1. We compute the weight function for particles at iteration r with

$$\begin{aligned} W_r^{(k)} \propto w_r^{(k)} &= w_{r-1}^{(k)} \cdot \frac{\gamma_r(\tilde{\beta}_{r-1}^{(k)})}{\gamma_{r-1}(\tilde{\beta}_{r-1}^{(k)})} \\ &= w_{r-1}^{(k)} \left(\frac{p(\mathbf{y}|\tilde{\beta}_{r-1}^{(k)})\tilde{\pi}_0(\tilde{\beta}_{r-1}^{(k)})}{\rho(\tilde{\beta}_{r-1}^{(k)})} \right)^{\alpha_r - \alpha_{r-1}}. \end{aligned} \quad (8)$$

Note that the weight update function for particles at the r th iteration only depends on particles at the $(r-1)$ th iteration, which is different from the standard SMC algorithm (Doucet, Godsill, and Andrieu 2000; Doucet, De Freitas, and Gordon 2001).

Step 2. We propagate new samples $\{\beta_r^{(k)}\}_{k=1}^K$ via π_r -invariant MCMC moves, $\{\beta_r^{(k)} \sim T_r(\tilde{\beta}_{r-1}^{(k)}, \cdot)\}_{k=1}^K$. The full conditional posterior distributions, $\pi_r(\sigma_i^2|\mathbf{c}_i)$, $\pi_r(\tau|\mathbf{c}, \theta, \lambda)$, $\pi_r(\theta|\mathbf{c}, \tau, \lambda)$ and $\pi_r(\mathbf{c}_i|\tau, \theta, \sigma, \mathbf{c}_{-i}, \lambda)$ are described in Supplementary Section 1.

Step 3. We conduct a resampling step to prune particles with small weights. The particles after the resampling step are denoted by $\{\tilde{\beta}_r^{(k)}\}_{k=1}^K$, and all particles are equally weighted. The simplest resampling method is multinomial resampling based on the normalized particle weights. However, advanced resampling schemes such as stratified resampling (Kitagawa 1996; Hol, Schon, and Gustafsson 2006) or residual resampling (Doucet and Cappé 2005) are preferable to multinomial resampling, since multinomial resampling will create more variance for the SMC estimator when compared with advanced resampling algorithms. In our numerical experiments, we use systematic resampling (Carpenter, Clifford, and Fearnhead 1999).

It is not recommended to conduct resampling at every iteration as resampling will create additional variation in the estimator (Chopin et al. 2004). Our resampling scheme is typically triggered when the relative effective sample size (rESS) falls below a given thresholds ζ . The effective sample size (ESS) at iteration r can be computed by

$$\text{ESS}_r^{(K)} = \frac{1}{\sum_{k=1}^K (W_r^{(k)})^2}.$$

$\text{ESS}_r^{(K)}$ denotes the number of ‘‘perfect’’ samples used to approximate the intermediate distribution π_r . Effective sample size

takes value between 1 and K . It takes value K if all particles are equally weighted, and it takes value close to 1 if one of the particles has a much larger weight than the others. The rESS normalizes the ESS to be between zero and one. The rESS at iteration r can be computed by $\text{rESS}_r^{(K)} = \text{ESS}_r^{(K)}/K$. If we never conduct resampling, the annealed SMC algorithm degenerates to the annealed importance sampling (Neal 2001).

After conducting the annealed SMC algorithm, we obtain a list of weighted samples to empirically represent the posterior distribution $\pi(\beta)$,

$$\hat{\pi}(\beta) = \sum_{k=1}^K W_R^{(k)} \cdot \delta_{\beta_R^{(k)}}(\beta).$$

3.2. Properties of the Annealed SMC Algorithm

We discuss some properties of our annealed SMC method. Note that the general SMC algorithm proposed by Del Moral, Doucet, and Jasra (2006) included our annealed SMC as a special case. Hence, *consistency* and *asymptotic normality* properties can be generated from Del Moral, Doucet, and Jasra (2006). We summarize these properties in following propositions. First, our annealed SMC method can provide a consistent representation of the intermediate target posterior distributions.

Proposition 1. The annealed SMC method provides asymptotically consistent estimates. We have

$$\sum_{k=1}^K W_r^{(k)} \psi(\beta_r^{(k)}) \rightarrow \int \pi_r(\beta) \psi(\beta) d\beta \quad \text{as } K \rightarrow \infty,$$

where the convergence is in L^2 norm, and ψ is a target function that satisfies regularity conditions, for example ψ is bounded. Del Moral (2004) and Chopin et al. (2004) discussed more general conditions which include the case of our annealed SMC algorithm.

The central limit theorem shown below can be used to assess the total variance of Monte Carlo estimators.

Proposition 2. Under the integrability conditions given in (Chopin et al. 2004, theor. 1), or (Del Moral 2004, pp. 300–306 in sec. 9.4), when multinomial resampling is performed at each iteration,

$$\begin{aligned} &K^{1/2} \left[\sum_{k=1}^K W_r^{(k)} \psi(\beta_r^{(k)}) - \int \pi_r(\beta) \psi(\beta) d\beta \right] \\ &\rightarrow N(0, \sigma_r^2(\psi)) \quad \text{as } K \rightarrow \infty, \end{aligned}$$

where the convergence is in distribution. The form of asymptotic variance $\sigma_r^2(\psi)$ depends on the Markov kernel T_r , and the artificial backward kernel L_r . We refer readers to Del Moral, Doucet, and Jasra (2006) for details of this asymptotic variance.

Note that Propositions 1 and 2 hold if the integral in \mathbf{R}_i can be computed exactly. These propositions do not hold exactly if we numerically approximate the integral due to the error being introduced.

In addition, the annealed SMC algorithm can be easily parallelized, by allocating particles across different cores (Del Moral, Doucet, and Jasra 2006; Wang, Wang, and Bouchard-Côté 2020).

Algorithm 1 An SMC algorithm of Bayesian inference for parameters in DEs

- 1: **Inputs:** (a) The prior distribution $\tilde{\pi}_0(\cdot)$ and the reference distribution $\rho(\cdot)$ over model parameters β , where $\beta = (\theta', \tau, \mathbf{c}', \sigma', \lambda)'$; (b) relative CESS threshold ϕ ; (c) resampling threshold ζ .
- 2: **Outputs:** Posterior approximation, $\hat{\pi}(\beta) = \sum_{k=1}^K W_R^{(k)} \cdot \delta_{\beta_R^{(k)}}(\beta)$.
- 3: Initialize the SMC iteration index and annealing parameter: $r \leftarrow 0, \alpha_0 \leftarrow 0$.
- 4: **for** $k \in \{1, 2, \dots, K\}$ **do**
- 5: Initialize particles with independent samples from the reference distribution:

$$\beta_0^{(k)} \leftarrow (\theta_0^{(k)'}, \tau_0^{(k)}, \mathbf{c}_0^{(k)'}, \sigma_0^{(k)'}, \lambda_0^{(k)'})'$$

- 6: Initialize weights to unity: $w_0^{(k)} \leftarrow 1, W_0^{(k)} \leftarrow 1/K$.
- 7: **for** $r \in \{1, 2, \dots\}$ **do**
- 8: Compute annealing parameter α_r using a bisection method with

$$f(\alpha) = \text{rCESS} \left(W_{r-1}^{(\cdot)}, \left(\frac{p(\mathbf{y}|\tilde{\beta}_{r-1}^{(\cdot)})\tilde{\pi}_0(\tilde{\beta}_{r-1}^{(\cdot)})}{\rho(\tilde{\beta}_{r-1}^{(\cdot)})} \right)^{\alpha_r - \alpha_{r-1}} \right) = \phi.$$

- 9: **for** $k \in \{1, \dots, K\}$ **do**
- 10: Compute unnormalized weights for $\beta_r^{(k)}$: $w_r^{(k)} = w_{r-1}^{(k)} \cdot \left(\frac{p(\mathbf{y}|\tilde{\beta}_{r-1}^{(k)})\tilde{\pi}_0(\tilde{\beta}_{r-1}^{(k)})}{\rho(\tilde{\beta}_{r-1}^{(k)})} \right)^{\alpha_r - \alpha_{r-1}}$.
- 11: Normalize weights: $W_r^{(k)} = w_r^{(k)} / (\sum_{k=1}^K w_r^{(k)})$.
- 12: Sample particles $\beta_r^{(k)}$ with one MCMC move admitting π_r as stationary/invariant distribution, using particles $\tilde{\beta}_{r-1}^{(k)}$ and the propagation step in *Supplementary Section 1*.
- 13: **if** $\alpha_r = 1$ **then**
- 14: return the current particle population $\{(\beta_r^{(k)}, W_r^{(k)})\}_{k=1}^K$.
- 15: **else**
- 16: **if** $\text{rESS} < \zeta$ **then**
- 17: Resample the particles.
- 18: **for** $k \in \{1, \dots, K\}$ **do**
- 19: Reset particle weights: $w_r^{(k)} = 1, W_r^{(k)} = 1/K$.
- 20: **else**
- 21: **for** $k \in \{1, \dots, K\}$ **do**
- 22: $\tilde{\beta}_r^{(k)} = \beta_r^{(k)}$.

3.3. Adaptive Annealing Parameter Scheme in SMC

In the annealed SMC algorithm, one challenge is to properly select the sequence of annealing parameters. If we choose $\alpha_0 = 0$ and $\alpha_1 = 1$, the annealed SMC sampler degenerates to importance sampling. A large number of annealing parameters improves the performance of algorithm, but it will be computationally more intensive. If we select an insufficient number of annealing parameters or an improper annealing scheme, the algorithm may collapse. We propose an adaptive annealing parameter scheme based on the seminal work of Del Moral, Doucet, and Jasra (2012), Zhou, Johansen, and Aston (2016), and Wang, Wang, and Bouchard-Côté (2020). Note that the weight function (Equation 8) for iteration r only depends on

particles of the $(r - 1)$ th iteration, and the difference between two successive annealing parameters $\alpha_r - \alpha_{r-1}$. This indicates that we can “manipulate” $\tilde{w}_r^{(k)}$ by changing the annealing parameter α_r . If α_r is close to α_{r-1} , then the incremental weight function $\tilde{w}_r^{(k)}$ is close to 1, and the variance of $\tilde{w}_r^{(k)}$ is smaller than it would be if we choose a larger value of α_r . This provides the intuition that we are able to control the discrepancy between two successive intermediate target distributions by manipulating α_r .

In this article, we use the relative conditional effective sample size (rCESS) (Zhou, Johansen, and Aston 2016) to measure the discrepancy between two successive intermediate targets. The rCESS normalizes the conditional effective sample size (CESS) to be between zero and one. The rCESS is defined as

$$\text{rCESS}_r(W_{r-1}^{(\cdot)}, \tilde{w}_r^{(\cdot)}) = \frac{(\sum_{k=1}^K W_{r-1}^{(k)} \tilde{w}_r^{(k)})^2}{\sum_{k=1}^K W_{r-1}^{(k)} (\tilde{w}_r^{(k)})^2},$$

which takes a value between $1/K$ and 1. The rCESS is equal to the relative ESS if we conduct resampling at every SMC iteration. Using the fact that $\tilde{w}_r^{(k)} = [p(\mathbf{y}|\beta_{r-1}^{(k)})\pi_0(\beta_{r-1}^{(k)})/\rho(\beta_{r-1}^{(k)})]^{\alpha_r - \alpha_{r-1}}$, rCESS_r is a decreasing function of α_r , where $\alpha_r \in (\alpha_{r-1}, 1]$. We control rCESS over iterations by selecting the annealing parameter α such that

$$f(\alpha) = \text{rCESS} \left(W_{r-1}^{(\cdot)}, \tilde{w}_r^{(\cdot)} \right) = \phi,$$

where ϕ is a value between 0 and 1. A small value of ϕ will lead to a high value of α_r , while a large value of ϕ will lead to a low value of α_r . It is impossible to obtain a closed-form solution of α^* by solving $f(\alpha) = \phi$, but we are able to use a bisection method to solve this one-dimensional search problem. The search interval of α is $(\alpha_{r-1}, 1]$. By using $f(\alpha_{r-1}) - \phi > 0, f(1) - \phi < 0$ (in case $f(1) \geq \phi$, we set $\alpha_r = 1$), and the continuous property of $f(\alpha) - \phi$, the solution α^* of $f(\alpha) = \phi$ is guaranteed. Algorithm 1 provides a detailed description for the SMC algorithm.

4. Real Data Analysis

In the dynamic system of the blowfly population, resource limitation acts with a time delay, roughly equal to the time for an egg to grow up to a pupa. One classic experiment on the resource competition in laboratory populations of Australian sheep blowflies (*Lucilia cuprina*) is studied by Nicholson (1954).

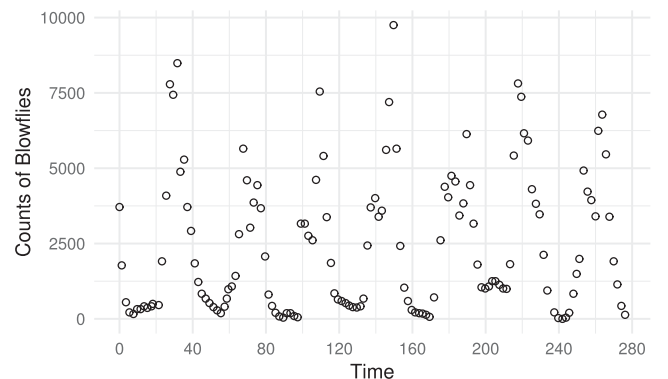


Figure 2. Blowfly population in one experiment published in Nicholson (1954); the time unit is one day.

The blowflies were cultivated in a room with temperature maintained at 25°C. The population of blowflies was measured every day for approximately one year. Figure 2 displays the counts of blowflies over time studied in Nicholson (1954). The time unit is one day. The oscillations displayed in the blowfly population are caused by the time lag between stimulus and reaction (Berezansky, Braverman, and Idels 2010). May (1976) proposed to model the counts of blowflies with the following DDE model

$$\frac{dx(t)}{dt} = \nu x(t)[1 - x(t - \tau)/(1000 \cdot P)], \quad (9)$$

where $x(t)$ is the blowfly population, ν is the rate of increase of the blowfly population, P is a resource limitation parameter set by the supply of food, and τ is the time delay, roughly equal to the time for an egg to grow up to a pupa. Our goal is to estimate the initial value, $x(0)$, and the three parameters, ν , P , and τ , from the noisy Nicholson's blowfly data $y(t)$. The observed counts of blowflies $y(t)$ is assumed to be lognormal distributed with mean $x(t)$ and variance σ^2 .

The counts of blowfly $x(t)$ is a positive function. Instead of modeling the constrained function $x(t)$ by a linear combination of cubic B-spline basis functions $W(t) = \Phi(t)'c$, we transform $x(t) = e^{W(t)}$ and use B-spline basis functions to model the unconstrained function $W(t) = \Phi(t)'c$. Equivalently, we solve the DDE

$$\frac{dW(t)}{dt} = \nu[1 - e^{W(t-\tau)}/(1000 \cdot P)], \quad (10)$$

with noisy observations $\log y(t) \sim N(W(t), \sigma^2)$.

We approximate the DDE solution using cubic B-splines with 34 equally spaced interior knots over the time span. The total number of knots is equal to 36. The total number of cubic B-spline functions is $L = 38$. Selection of the number of basis functions is explored in Section 5.1.2. Our prior/reference distributions for parameter of interest $(c, \nu, P, \tau, \sigma^2, \lambda)'$ are

$$\begin{aligned} \nu &\sim N(0, 5^2)I(\nu > 0), \quad P \sim N(0, 5^2)I(P > 0), \\ \tau &\sim \text{Unif}(0, 50), \quad c \sim \text{MVN}(\hat{c}, 100^2 I_L), \\ \sigma^2 &\sim \text{IG}(1, 1), \quad \lambda \sim \text{Gamma}(1, 1). \end{aligned}$$

In our adaptive SMC, we set the rCESS threshold $\phi = 0.9$ and resampling threshold $\zeta = 0.5$. The number of particles

Table 1. Posterior mean and corresponding 95% credible interval (CI) for parameters in population dynamics of blowflies.

	ν	P	τ
Mean	0.18	2.37	8.37
(2.5%, 97.5%)	(0.07, 0.28)	(1.31, 3.33)	(5.66, 9.92)
	$W(0)$	σ^2	λ
Mean	8.30	0.53	3.35
(2.5%, 97.5%)	(7.22, 9.36)	(0.46, 0.61)	(1.63, 5.91)

is $K = 500$. Under this setting, the number of SMC iteration was $R = 227$. Figure 2 in the supplementary material shows the annealing parameter sequence $\alpha_{1:R}$ under the adaptive scheme. In Section 5.1.2, we will compare the performance of our method using different values of ϕ and K . Figure 3 displays the estimated DDE trajectory. The left panel of Figure 3 shows the estimated $W(t)$ and the 95% pointwise credible intervals; the right panel of Figure 3 shows $X(t) = e^{W(t)}$, in which the blue points are observed data.

Table 1 displays the posterior means and the corresponding 95% pointwise credible intervals (CI) for DDE parameters in Equation (10). Note that our point estimates are similar to those obtained from Wang and Cao (2012), in which the same nonparametric function expressed using B-splines is estimated by maximizing the DDE-defined penalized likelihood function. However, the uncertainty of these parameters is significantly underestimated using their frequentist approach. In contrast, our Bayesian approach can provide more reasonable estimates for the parameter uncertainty. More concretely, we compare the estimates for the main parameter of interest, the delay parameter τ , which can be interpreted as the time for an egg to grow up to a pupa. From Table 1, our posterior mean of τ and its 95% CI is 8.368 (5.656, 9.916) while the maximum likelihood estimate for τ is 8.781 and the standard error is 0.039 in Wang and Cao (2012).

Figure 4 displays the pairwise scatterplots of the posterior samples of ν , τ , and P . We also calculate the correlation between posterior samples: $\text{corr}(\nu, \tau) = 0.139$, $\text{corr}(\nu, P) = 0.598$, $\text{corr}(P, \tau) = 0.008$. Recall that ν is the rate of increase of the blowfly population and τ is the time delay, roughly equal to the time for an egg to grow up to a pupa. The small positive value of the correlation between τ and ν indicates that the blowfly

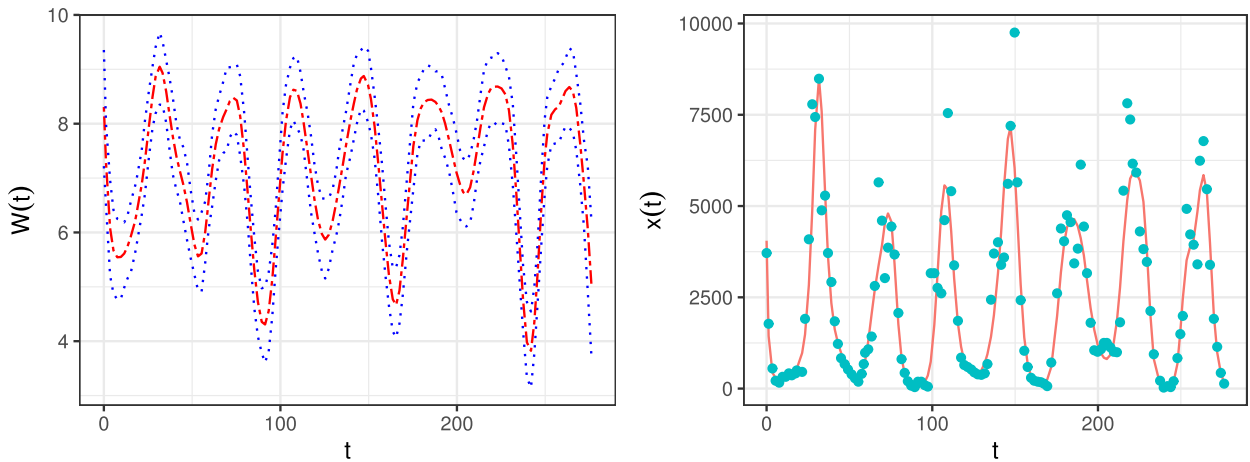


Figure 3. Estimated posterior mean trajectory and 95% pointwise credible intervals for the DDE modeling the population dynamics of blowflies.

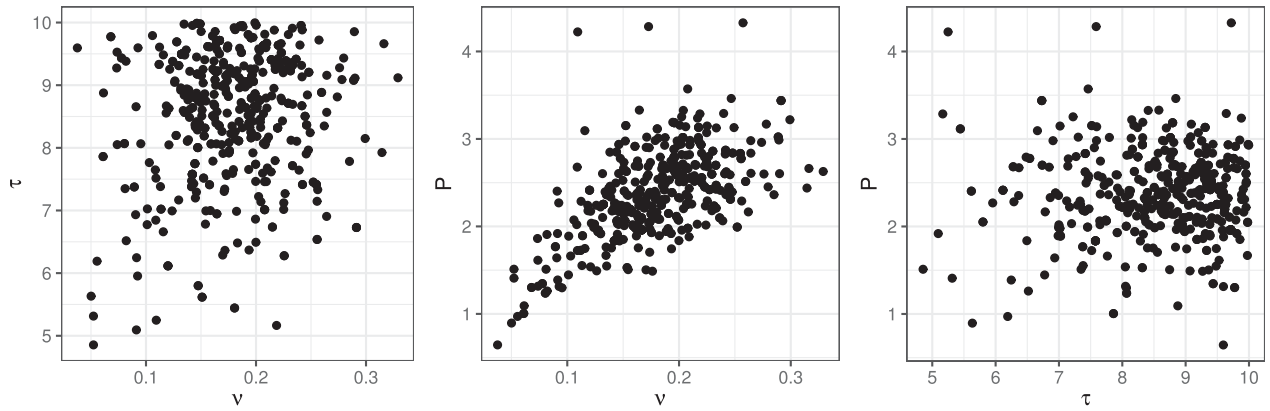


Figure 4. Posterior samples of ν , τ , P for DDE in Equation (10) estimated via SMC. We resample the particles at the last SMC iteration such that they are equally weighted. Correlation: $\text{corr}(\nu, \tau) = 0.139$, $\text{corr}(\nu, P) = 0.598$, $\text{corr}(P, \tau) = 0.008$.

population will increase if eggs take their time to develop into pupae. The parameter P is related to a resource limitation. The relatively large positive correlation between ν and P can be easily understood: the blowfly population grows faster when there is a larger food supply. The tiny positive value of the correlation between τ and P implies that the amount of food supply has a small impact on the period of being a pupa.

5. Simulation Study

We use simulation studies to demonstrate the effectiveness of our proposed model and method. The experiments include both ODE and DDE examples. We use the R package *deSolve* (Soetaert, Petzoldt, and Setzer 2010) to simulate differential equations.

5.1. A Nonlinear Ordinary Differential Equation Example

In this section, we use a nonlinear ODE example to illustrate the numerical behavior of SMC algorithm. We generate ODE trajectories according to the following ODE system:

$$\begin{aligned} \frac{dx_1(t)}{dt} &= \frac{72}{36 + x_2(t)} - \theta_1, \\ \frac{dx_2(t)}{dt} &= \theta_2 x_1(t) - 1, \end{aligned} \quad (11)$$

where $\theta_1 = 2$ and $\theta_2 = 1$, and initial conditions $x_1(0) = 7$ and $x_2(0) = -10$. The observations y_i are simulated from a normal distribution with mean $x_i(t|\theta)$ and variance σ_i^2 , where $\sigma_1 = 1$ and $\sigma_2 = 3$. We generate 121 observations for each ODE function, equally spaced within $[0, 60]$ (see Figure 3 in the supplementary material). Under this setting, the posterior distribution of θ_1 and θ_2 will have multiple local modes (see Figure 1).

We use cubic B-spline functions (see Figure 1 in the supplementary material) to represent ODE trajectories. We put equally spaced knots on each of eight observations. The total number of knots is 16, including 14 interior knots. The total number of cubic B-spline functions is $L_1 = L_2 = 18$. We select weak prior/reference distributions of β for the SMC algorithm,

$$\begin{aligned} \theta_1 &\sim N(5, 5^2), \quad \theta_2 \sim N(5, 5^2), \\ \mathbf{c}_1 &\sim \text{MVN}(\hat{\mathbf{c}}_1, 100^2 \mathbf{I}_{L_1}), \quad \mathbf{c}_2 \sim \text{MVN}(\hat{\mathbf{c}}_2, 100^2 \mathbf{I}_{L_2}), \\ \sigma_1^2 &\sim \text{IG}(1, 1), \quad \sigma_2^2 \sim \text{IG}(1, 1), \quad \lambda \sim \text{Gamma}(1, 1). \end{aligned}$$

5.1.1. Comparison of SMCs and MCMCs

We first alter Equation (11) to produce a symmetric, bimodal posterior for θ_1 ,

$$\begin{aligned} \frac{dx_1(t)}{dt} &= \frac{72}{36 + x_2(t)} - |\theta_1|, \\ \frac{dx_2(t)}{dt} &= \theta_2 x_1(t) - 1. \end{aligned} \quad (12)$$

We compare the performance of annealed SMC targeting $\pi(\beta)$ (denoted SMC-spline) with the following three algorithms in terms of speed and estimation using Equation (12): annealed SMC targeting $\pi(\theta, \tau, \mathbf{x}(0), \sigma^2)$ (SMC-deSolve), MCMC targeting $\pi(\beta)$ (MCMC-spline), and MCMC targeting $\pi(\theta, \tau, \mathbf{x}(0), \sigma^2)$ (MCMC-deSolve). *Supplementary Section 2* details the three algorithms.

We simulate the ODE trajectories using the “*Isoda*” method in the R package *deSolve* (Soetaert, Petzoldt, and Setzer 2010). In SMC-spline, we set the rCESS threshold $\phi = 0.9$ and resampling threshold $\zeta = 0.5$. The total number of particles we use is $K = 500$. With given samples $\theta^{(n)}$, $\mathbf{x}(0)^{(n)}$ in the SMC-deSolve and MCMC-deSolve methods, we use the “*euler*” method in *deSolve* to solve ODEs to obtain $\mathbf{x}(t_{ij}|\theta^{(n)})$, $\mathbf{x}(0)^{(n)}$. We purposely use a different ODE solver to mimic the fact that no data generation information is available for real data. We select weak prior distributions for θ and $\mathbf{x}(0)$. In the SMC-deSolve algorithms, we use $K = 500$ and $\phi = 0.999$ and utilize the same prior distributions and MCMC moves as those in the MCMC-deSolve methods. We run both MCMC algorithms with 400,000 iterations, which is close to $K \cdot R$ in SMC-spline.

Figure 5 shows the comparison of four algorithms in terms of estimating θ . Panel (a) and (b) show samples of the intermediate posterior distributions for θ by running SMC-spline and SMC-deSolve, respectively. The points with colors from light gray to dark gray are samples for $r = 1, R/6, R/2, R$, where $R = 942$ in SMC-spline and $R = 1220$ in SMC-deSolve. Panel (c) and (d) display the trace plots for θ by running MCMC-spline and MCMC-deSolve, respectively. The black dots indicate the

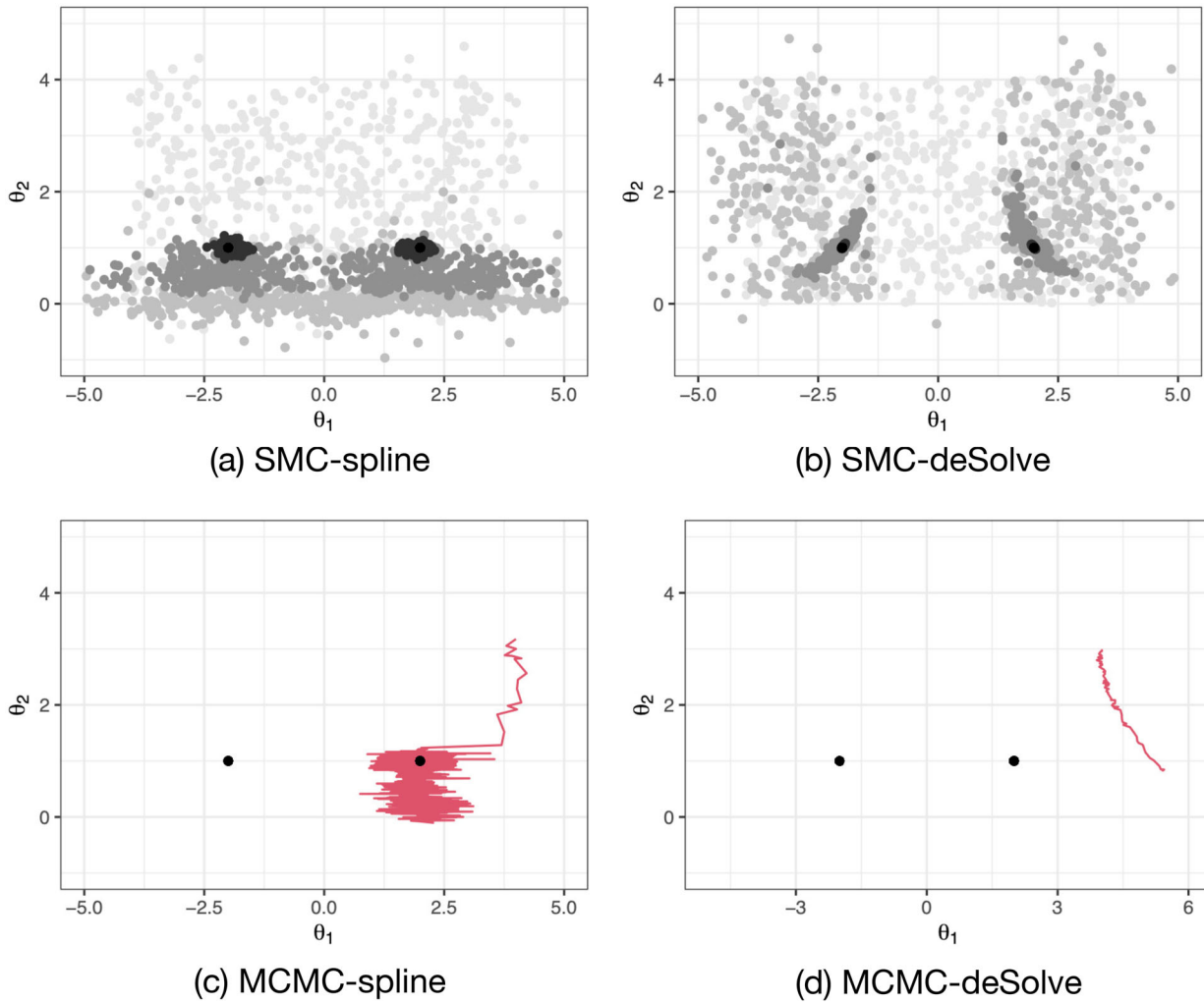


Figure 5. (a) and (b): Intermediate posterior distributions for θ by running SMC-spline and SMC-deSolve, respectively. The points with colors from light gray to dark gray are samples for $r = 1, R/6, R/2, R$. (c) and (d): Trace plots for θ by running MCMC-spline and MCMC-deSolve, respectively. The black dots indicate true parameter values for generating ODEs.

Table 2. Posterior mean of θ and $\mathbf{x}(0)$ (95% CI) from the four algorithms.

	True	SMC-spline	SMC-deSolve	MCMC-spline	MCMC-deSolve
$ \theta_1 $	2	1.93(1.68, 2.19)	1.84(1.81, 1.88)	1.92(1.48, 2.37)	5.37(5.37, 5.37)
θ_2	1	0.99(0.90, 1.09)	1.12(1.08, 1.17)	0.97(0.87, 1.08)	0.85(0.85, 0.85)
$x_1(0)$	7	6.43(2.56, 10.11)	3.94(3.71, 4.22)	6.48(4.01, 8.92)	4.55(4.55, 4.55)
$x_2(0)$	-10	-10.66(-17.86, -2.26)	-4.47(-5.63, -3.23)	-10.28(-14.18, -6.71)	0.68(0.65, 0.70)

true parameter values in the ODE. For SMC-spline and SMC-deSolve, the particles gradually move to the posterior distribution with increasing annealing parameters. For MCMC-spline, the acceptance ratio of MH algorithm is 21.4%. It can explore one mode rather than two modes created in Equation (12). For MCMC-deSolve, the acceptance rate of MH algorithm is 25.2%. It gets stuck in local modes close to the initial value, and cannot explore the two modes. Table 2 shows the posterior means of θ and $\mathbf{x}(0)$ as well as the corresponding 95% credible interval (CI) using SMC-spline, SMC-deSolve, MCMC-spline, and MCMC-deSolve. SMC-spline and MCMC-spline provide posterior means close to the true values, and the corresponding credible intervals cover the true values, while MCMC-spline can only explore one mode of θ_1 . Our experiment in Supplementary Section 4.1.1 reports the estimated ODE trajectories

and the 95% pointwise credible bands by SMC-spline. The estimated mean ODE trajectories are very close to the true ODE trajectories. The 95% pointwise credible bands cover the true ODE trajectories. The posterior mean of θ and $\mathbf{x}(0)$ provided by SMC-deSolve has a larger bias. The true value of θ and $\mathbf{x}(0)$ are not included in the 95% CIs, indicating that the estimated CI might be too narrow. MCMC-deSolve does not converge to the posterior distribution.

We also run SMC-deSolve and MCMC-deSolve using the “*Isoda*” method in *deSolve* with the same setting. Note that the “*Isoda*” is the same ODE solver that is used for simulating ODEs and therefore it will favor the algorithms SMC-deSolve and MCMC-deSolve. MCMC-deSolve does not converge to the posterior distribution. Table 3 displays the posterior mean of θ and $\mathbf{x}(0)$ together with the 95% CI from SMC-deSolve and

Table 3. Posterior mean of θ and $\mathbf{x}(0)$ together with the 95% CI from SMC-deSolve and MCMC-deSolve using “Isoda” for solving ODE.

	True value	SMC-deSolve	MCMC-deSolve
$ \theta_1 $	2	1.98(1.96, 2.01)	1.56(1.48, 1.65)
θ_2	1	1.02(1.00, 1.05)	4.58(3.70, 7.03)
$x_1(0)$	7	6.87(6.61, 7.13)	-0.09(-0.51, 0.57)
$x_2(0)$	-10	-10.52(-11.58, -9.34)	6.65(1.32, 12.45)

MCMC-deSolve. The posterior means from SMC-deSolve are close to the true values, and the credible intervals seem narrow. The true value of θ_2 is on the boundary of the 95% CI. Obviously, the methods with DE solvers heavily rely on the choice of numerical solvers and they tend to ignore the uncertainty from approximating DE solutions using these solvers.

We also compare two types of MCMC moves (MCMC-spline and MCMC-deSolve) in terms of computing time. Every 1000 MCMC-spline moves cost 0.71 sec on a 2.3 GHz Intel Core i9 processor, while every 1000 MCMC-deSolve moves cost 10.44 seconds on same machine. This indicates that representing DE trajectories using a linear combination of basis functions can significantly increase the speed compared to using numerical solvers.

5.1.2. Comparison of SMCs Using Different Tuning Parameters

We conduct experiments investigating the selection of tuning parameters ϕ , K , λ , and number of basis functions. 1. The parameter estimates get closer to the true values, and the RMSE of the ODE trajectories gets smaller, when we increase the rCESS threshold ϕ . A higher value of rCESS threshold is equivalent to more intermediate target distributions. 2. The proposed SMC method performs better when we use a large number of particles. 3. For a given amount of computation, a relatively small K and a large ϕ is optimal. However, a too small value of K is not recommended, as an extremely small K may lead to large Monte Carlo variance. 4. This experiment indicates a sufficient number of basis functions is important in ODE trajectory estimation. However, we do not recommend using an overly large number of basis functions because it gains little in improving the approximation of ODE trajectories and causes challenges in sampling high-dimensional basis function coefficients. 5. The Bayesian scheme for sampling λ performs satisfactorily in terms of parameter estimates and estimated ODE trajectories. The details of experiments are displayed in the supplementary material (Sections 4.1.2 and 4.1.3).

5.2. DDE Examples

5.2.1. Hutchinson’s Equation

Our first DDE example is the Hutchinson’s equation, which is used to model the blowfly data in Section 4,

$$\frac{dx(t)}{dt} = \nu x(t)[1 - x(t - \tau)/(1000 \cdot P)],$$

where τ , ν , and P are parameters of interest in the DDE. We set $x(0) = 3500$, $\tau = 3$, $\nu = 0.8$, and $P = 2$ to simulate the DDE trajectory. The DDE trajectory is observed with measurement error. The error is lognormally distributed with mean 0

Table 4. Estimated parameters and MSE of $W(t)$ for three simulated datasets.

J	ν	P	τ
True	0.8	2	3
101	0.64 (0.43, 0.82)	2.07 (1.56, 2.61)	3.05 (2.66, 3.56)
201	0.73 (0.60, 0.90)	1.98 (1.66, 2.32)	3.02 (2.73, 3.26)
401	0.75 (0.63, 0.86)	2.09 (1.80, 2.38)	3.00 (2.84, 3.16)

J	$W(0)$	σ^2	RMSE
True	8.16	0.16	—
101	7.93 (7.34, 8.47)	0.12 (0.08, 0.18)	0.26
201	7.90 (7.53, 8.33)	0.14 (0.11, 0.18)	0.21
401	7.98 (7.57, 8.40)	0.17 (0.15, 0.19)	0.14

Table 5. Coverage probability and averaged value of the 95% CI.

K	ϕ	ν (0.22)	P (2)	τ (8)	σ (0.2)
200	0.98	97.8% (0.20, 0.23)	97.8% (1.91, 2.18)	91.1% (7.18, 8.18)	95.5% (0.18, 0.22)

and standard deviation σ . We investigate the influence of the number of points per time step, and the influence of standard deviation of error σ . We simulate datasets in two scenarios. In the first scenario, we simulate 3 datasets, with 101, 201, and 401 observations respectively, equally spaced in $[0, 100]$. The standard deviation of error is $\sigma = 0.4$. In the second scenario, we simulate 36 datasets with 201 observations equally spaced in $[0, 100]$. The standard deviations of error for the 36 datasets are $\sigma = (0.1, 0.5, 1.5)$, 12 datasets for each level of σ .

We transform the positive constraint function $x(t) = e^{W(t)}$ and use B-spline basis functions to model the unconstrained function $W(t) = \Phi(t)'c$. This is equivalent to solving the DDE displayed in Equation (10). We put 51 knots equally spaced in $[0, 100]$, including 49 interior knots. The total number of cubic B-spline functions is $L = 53$. The hyperparameters in DDE parameter prior/reference distributions and sequential Monte Carlo setups are the same as Section 4.

Table 4 displays the estimated parameters ($\nu, P, \tau, W(0)$) and RMSE defined in Equation (12) in the Supplement for $W(t)$ using datasets simulated in the first scenario. For the same DDE function, a larger number of observations improves the performance of estimation. Figure 6 shows the estimated parameters ($\nu, P, \tau, W(0)$) and RMSE using datasets simulated in the second scenario. It indicates that a smaller value of σ improves the estimation.

We also evaluate the quality of the uncertainty estimation of the parameters through a simulation study using Hutchinson’s DDE model. The true parameters are set to $\nu = 0.22$, $P = 2$, $\tau = 8$, and $\sigma = 0.2$. We generate 50 datasets of 200 observations in the time interval $[0, 130]$ with different random seeds and run our SMC algorithm for each of them. We use 16 equally spaced knots in $[0, 130]$. Table 5 shows the percentage that the 95% CI’s cover the true parameter value and the averaged 95% CI. The coverage probability is close to the nominal 95%.

5.2.2. A Nonlinear DDE Example

In this section, we investigate a nonlinear DDE model proposed by Monk (2003) to model the feedback inhibition of gene expression. The nonlinear DDE is described as follows:

$$\frac{dx_1(t)}{dt} = \frac{1}{1 + (x_2(t - \tau)/p_0)^n} - \mu_m x_1(t),$$

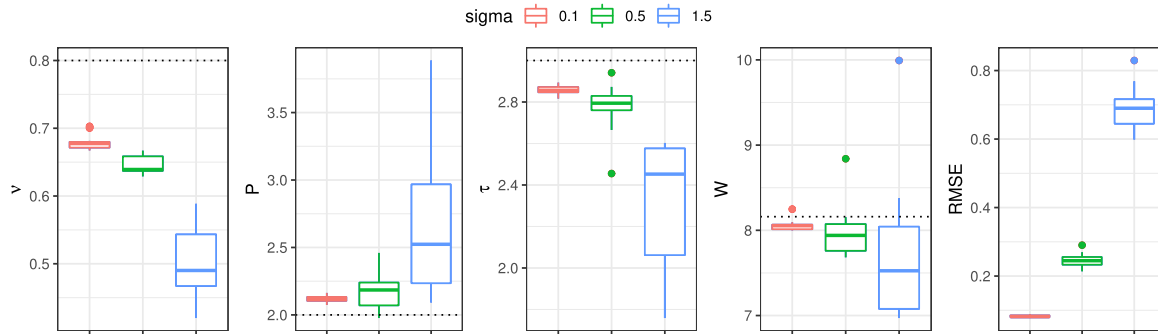


Figure 6. Influence of σ on DDE estimation. We simulate datasets with $\sigma = (0.1, 0.5, 1.5)$. A small value of σ improves the performance of estimation.

$$\frac{dx_2(t)}{dt} = x_1(t) - \mu_p x_2(t). \quad (13)$$

In Equation (13), $x_1(t)$ denotes the expression of *mRNA* at time t , and $x_2(t)$ denotes the expression of a *protein* at time t . There is a delayed repression of *mRNA* production by the *protein*. The DDE system depends on the *transcriptional delay* τ , and degradation rates μ_m and μ_p , the expression threshold p_0 , and the Hill coefficient n . As noted in Monk (2003), there is significant nonlinearity in the DDE system when the Hill coefficient $n > 4$. We simulate a DDE system and noisy observations. We use B-spline functions to represent the DDE trajectories and use SMC to estimate parameters. The posterior means of the parameters are fairly close to the true values, and the 95% credible intervals cover the true values. The estimated mean DDE trajectories are generally very close to the true DDE trajectories. The 95% pointwise confidence bands cover the true DDE trajectories. The details of setups and results are shown in the supplementary material, Section 4.2.

6. Discussion

We proposed an adaptive semi-parametric Bayesian framework to solve nonlinear differential equations and estimate the DE parameters using an efficient annealed sequential Monte Carlo method. The main idea is to represent DE trajectories using a linear combination of basis functions and to estimate the coefficients of these basis functions together with other DE parameters using an annealed sequential Monte Carlo algorithm. The proposed method avoids using DE solvers which can be computationally expensive and sensitive to the initial state and model parameters. Our work is a Bayesian method with two obvious advantages over the counterpart using a frequentist method. First, the Bayesian method can easily achieve uncertainty estimates. Second, we avoid expensive tuning for the global smoothing parameter by treating it in the same way as other parameters.

We represent DE variables with a linear combination of basis functions. A prior distribution on the basis function coefficients is used to control the tradeoff between fit to the data and fidelity to the DE model. Our model is related to the generalized profiling approaches developed by Ramsay et al. (2007), in which the coefficients of the basis functions and DE parameters are estimated by a penalized smoothing procedure.

Qi et al. (2010) investigated the asymptotic bias induced by the spline approximation in the generalized profiling approaches. Pang, Yan, and Zhou (2017) loosened the assumptions for the asymptotic properties. We refer readers to Qi et al. (2010); Pang, Yan, and Zhou (2017) for the details of assumptions, theorems, and proofs.

We developed a sequential Monte Carlo method in an annealing framework to estimate the DE parameters. The annealed SMC considers the same parameter space for all the intermediate distributions. Consequently, MCMC moves used in the literature on Bayesian inference for differential equations can be repurposed to act as SMC proposal distributions in the annealed SMC. Note that more advanced MCMC moves can be used to further improve the performance of the annealed SMC. Annealed SMC is preferred over MCMC algorithms for several reasons. First, the developed SMC method can fully explore the multi-modal posterior surface of DE parameters. Second, the proposed method is a semi-automatic algorithm that requires minimal tuning from the user; given a criterion for the rCESS and the number of particles, it can adaptively choose a scheme for the sequence of the annealing parameters that determine the intermediate target distributions of SMC. Third, the annealed SMC is an embarrassingly parallel method. Unlike running an MCMC chain for a long time until it converges, the annealed SMC is more efficient because a large number of particles can be run on different CPUs or GPUs simultaneously.

We used different simulation scenarios to explore the numerical behavior of our model and method, and demonstrated it can perform well in both ODE and DDE parameter estimation. Our simulation studies provide some guideline for choosing the value of rCESS and the number of particles. To ensure more accurate estimates of the DE parameters from the annealed SMC, a rule of thumb is to choose a large number for rCESS and to avoid using an extremely small number of particles. We also applied our method to a real data example to model the population dynamics of blowflies with a DDE. The delay parameter in DDEs is usually challenging to estimate. But our application shows that our method is superior to the previous frequentist method.

There are several improvements and extensions based on our proposed method for future work. Note there might be other satisfactory ways to construct the intermediate distributions for our SMC algorithm such that the last distribution of the sequence is our target distribution. In theory, when the

number of particles and the number of the SMC iterations are large enough, such an alternative SMC algorithm can also well approximate the same target distribution. But in practice, different sequences of intermediate distributions may perform differently given the same computing budget. In future, it is worth exploring alternative ways to construct the intermediate distributions and make comparisons. For simplicity, we have used the same reference distributions as the prior distributions for most of the parameters. The performance of the annealed SMC can be improved by using reference distributions that are close to the target distribution. In all of our current numerical experiments, we put equally spaced knots for smoothing splines and the number of knots are predetermined before running experiments. In future work, we will explore using a smaller number of knots that are well placed, and let the data determine the number of knots and their locations. The adaptive control of knots in smoothing spline for DEs will benefit the estimation of DEs, especially those with sharp changes. In practice, it is often the case that there are several DE models that are proposed to describe the same dynamic system. This requires selection among various differential equations models. One direction of future work is to explore model selection methods for DEs. Another line of future work is to develop more scalable SMC algorithms for estimating parameters in a large series of differential equations.

Funding

This research is supported by the National Natural Science Funds of China (No. 12101333), the Fundamental Research Funds for the Central Universities, Nankai University (No. 63211089), the Shanghai Science and Technology Program (No. 21010502500), the startup fund of ShanghaiTech University, and Discovery Grant (RGPIN-2019-06131) from the Natural Sciences and Engineering Research Council of Canada (NSERC).

Supplementary Materials

Technical material: The file supplementary.pdf (PDF file) provides details of the algorithms and simulation studies in the article.

R-package: The R package “smcDE” was developed to implement our proposed methods, real data analysis, and simulation studies. It is available from <https://github.com/shijaw/smcDE>.

ORCID

Shijia Wang  <http://orcid.org/0000-0003-0339-1716>

Liangliang Wang  <http://orcid.org/0000-0002-8509-7985>

References

- Ascher, U. M., Ruuth, S. J., and Spiteri, R. J. (1997), “Implicit–Explicit Runge–Kutta Methods for Time-Dependent Partial Differential Equations,” *Applied Numerical Mathematics*, 25, 151–167. [1]
- Barber, D., and Wang, Y. (2014), “Gaussian Processes for Bayesian Estimation in Ordinary Differential Equations,” in *International Conference on Machine Learning*, pp. 1485–1493. [2]
- Berezansky, L., Braverman, E., and Idels, L. (2010), “Nicholson’s Blowflies Differential Equations Revisited: Main Results and Open Problems,” *Applied Mathematical Modelling*, 34, 1405–1417. [8]
- Berry, S. M., Carroll, R. J., and Ruppert, D. (2002), “Bayesian Smoothing and Regression Splines for Measurement Error Problems,” *Journal of the American Statistical Association*, 97, 160–169. [3,4]
- Bhaumik, P., and Ghosal, S. (2015), “Bayesian Two-Step Estimation In Differential Equation Models,” *Electronic Journal of Statistics*, 9, 3124–3154. [2]
- Bhaumik, P., Ghosal, S., (2017), “Efficient Bayesian Estimation and Uncertainty Quantification in Ordinary Differential Equation Models,” *Bernoulli*, 23, 3537–3570. [2]
- Bulirsch, R., and J. Stoer (1966), “Numerical Treatment of Ordinary Differential Equations by Extrapolation Methods,” *Numerische Mathematik*, 8, 1–13. [1]
- Burden, R. L., Faires, J. D., and Reynolds, A. C. (2011), *Numerical Analysis* (9th ed.). Boston, MA: Brooks/Cole, Cengage Learning. [4]
- Butcher, J. C. (2016), *Numerical Methods for Ordinary Differential Equations*, Chichester, West Sussex: Wiley. [1]
- Calderhead, B., Girolami, M., and Lawrence, N. D. (2009), “Accelerating Bayesian Inference Over Nonlinear Differential Equations With Gaussian Processes,” in *Advances in Neural Information Processing Systems*, eds. Yoshua Bengio, Dale Schuurmans, John Lafferty, Chris Williams and Aron Culotta, pp. 217–224. Vancouver, BC: Curran Associates, Inc. [2]
- Campbell, D., and Steele, R. J. (2012), “Smooth Functional Tempering for Nonlinear Differential Equation Models,” *Statistics and Computing*, 22, 429–443. [2]
- Cao, J., Huang, J. Z., and Wu, H. (2012), “Penalized Nonlinear Least Squares Estimation of Time-Varying Parameters in Ordinary Differential Equations,” *Journal of Computational and Graphical Statistics*, 21, 42–56. [1]
- Cao, J., Wang, L., and Xu, J. (2011), “Robust Estimation for Ordinary Differential Equation Models,” *Biometrics*, 67, 1305–1313. [1]
- Carpenter, J., Clifford, P., and Fearnhead, P. (1999), “Improved Particle Filter for Nonlinear Problems,” *IEEE Proceedings-Radar, Sonar and Navigation*, 146, 2–7. [6]
- Chen, J., and Wu, H. (2008), “Efficient Local Estimation for Time-Varying Coefficients in Deterministic Dynamic Models With Applications to HIV-1 Dynamics,” *Journal of the American Statistical Association*, 103, 369–384. [1]
- Chopin, N. (2004), “Central limit theorem for sequential Monte Carlo methods and Its Application to Bayesian Inference,” *The Annals of Statistics*, 32, 2385–2411. [2,6]
- Dass, S. C., Lee, J., Lee, K., and Park, J. (2017), “Laplace Based Approximate Posterior Inference for Differential Equation Models,” *Statistics and Computing*, 27, 679–698. [2]
- De Boor, C. (1972), “On Calculating With B-Splines,” *Journal of Approximation Theory*, 6, 50–62. [3]
- Del Moral, P. (2004), *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, New York: Springer. [6]
- Del Moral, P., Doucet, A., and Jasra, A. (2006), “Sequential Monte Carlo Samplers,” *Journal of the Royal Statistical Society, Series B*, 68, 411–436. [2,5,6]
- Del Moral, P., Doucet, A., and Jasra, A. (2012), “An Adaptive Sequential Monte Carlo Method for Approximate Bayesian Computation,” *Statistics and Computing*, 22, 1009–1020. [5,7]
- Dondelinger, F., Husmeier, D., Rogers, S., and Filippone, M. (2013), “ODE Parameter Inference Using Adaptive Gradient Matching With Gaussian Processes,” in *Artificial Intelligence and Statistics*, eds. C. M. Carvalho, P. Ravikumar, pp. 216–228. Scottsdale, Arizona: PMLR. [2]
- Douc, R., and Cappé, O. (2005), “Comparison of Resampling Schemes for Particle Filtering,” in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005 (ISPA 2005)*, pp. 64–69. Zagreb, Croatia: IEEE. [6]
- Doucet, A., De Freitas, N., and Gordon, N. (2001), “An Introduction to Sequential Monte Carlo Methods,” in *Sequential Monte Carlo Methods in Practice*, eds. A. Doucet, N. de Freitas, N. Gordon, pp. 3–14. New York, NY: Springer. [2,5,6]
- Doucet, A., Godsill, S., and Andrieu, C. (2000), “On Sequential Monte Carlo Sampling Methods for Bayesian Filtering,” *Statistics and Computing*, 10, 197–208. [2,5,6]
- Fan, Y., Wu, R., Chen, M.-H., Kuo, L., and Lewis, P. O. (2011), “Choosing Among Partition Models in Bayesian Phylogenetics,” *Molecular Biology and Evolution*, 28, 523–532. [2,5]
- Hochbruck, M., Lubich, C., and Selhofer, H. (1998), “Exponential Integrators for Large Systems of Differential Equations,” *SIAM Journal on Scientific Computing*, 19, 1552–1574. [1]

- Hochbruck, M., and Ostermann, A. (2010), “Exponential Integrators,” *Acta Numerica*, 19, 209–286. [1]
- Hol, J. D., Schon, T. B., and Gustafsson, F. (2006), “On Resampling Algorithms for Particle Filters,” in *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pp. 79–82. IEEE. [6]
- Jain, M. K. (1979), *Numerical Solution of Differential Equations*, Eastern New Delhi: Wiley. [1]
- Jameson, A., Schmidt, W., and Turkel, E. (1981), “Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge Kutta Time Stepping Schemes,” in *14th Fluid and Plasma Dynamics Conference*, pp. 1259. [1]
- Kitagawa, G. (1996), “Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models,” *Journal of Computational and Graphical Statistics*, 5, 1–25. [6]
- Lee, K., Lee, J., and Dass, S. C. (2018), “Inference for Differential Equation Models Using Relaxation Via Dynamical Systems,” *Computational Statistics & Data Analysis*, 127, 116–134. [2]
- Liu, J. S., and Chen, R. (1998), “Sequential Monte Carlo Methods for Dynamic Systems,” *Journal of the American Statistical Association*, 93, 1032–1044. [2]
- May, R. M. (1976), “Models for Single Populations,” *Theoretical Ecology: Principles and Applications*, ed. R. M. May, pp. 4–25. Philadelphia Toronto: W.B. Saunders Company. [8]
- Monk, N. A. (2003), “Oscillatory Expression of Hes1, p53, and NF- κ B Driven by Transcriptional Time Delays,” *Current Biology*, 13, 1409–1413. [11,12]
- Neal, R. M. (2001), “Annealed Importance Sampling,” *Statistics and Computing*, 11, 125–139. [5,6]
- Nicholson, A. J. (1954), “An Outline of the Dynamics of Animal Populations,” *Australian Journal of Zoology*, 2, 9–65. [7,8]
- Pang, T., Yan, P., and Zhou, H. H. (2017), “Asymptotically Efficient Parameter Estimation for Ordinary Differential Equations,” *Science China Mathematics*, 60, 2263–2286. [1,12]
- Poyton, A., Varziri, M. S., McAuley, K. B., McLellan, P., and Ramsay, J. O. (2006), “Parameter Estimation in Continuous-Time Dynamic Models Using Principal Differential Analysis,” *Computers & Chemical Engineering*, 30, 698–708. [1]
- Qi, X., Zhao, H. (2010), “Asymptotic Efficiency and Finite-Sample Properties of the Generalized Profiling Estimation of Parameters in Ordinary Differential Equations,” *The Annals of Statistics*, 38, 435–481. [1,12]
- Ramsay, J. O. (2006), “Functional Data Analysis,” in <https://doi.org/10.1002/0471667196.ess3138> *Encyclopedia of Statistical Sciences*, eds. S. Kotz, C. B. Read, N. Balakrishnan, B. Vidakovic and N. L. Johnson (vol. 4), pp. 1–8. [3]
- Ramsay, J. O., Hooker, G., Campbell, D., and Cao, J. (2007), “Parameter Estimation for Differential Equations: A Generalized Smoothing Approach,” *Journal of the Royal Statistical Society, Series B*, 69, 741–796. [1,3,12]
- Ramsay, J. O., and Silverman, B. W. (2007), *Applied Functional Data Analysis: Methods and Case Studies*. New York: Springer. [1]
- Reiss, P. T., and Todd Ogden, R. (2009), “Smoothing Parameter Selection for a Class of Semiparametric Linear Models,” *Journal of the Royal Statistical Society, Series B*, 71, 505–523. [4]
- Rosenzweig, M. L., and MacArthur, R. H. (1963), “Graphical Representation and Stability Conditions of Predator–Prey Interactions,” *The American Naturalist*, 97, 209–223. [1]
- Soetaert, K., Petzoldt, T., and Setzer, R. W. (2010), “Solving Differential Equations in R: Package Desolve,” *Journal of Statistical Software*, 33. [9]
- Varah, J. M. (1982), “A Spline Least Squares Method for Numerical Parameter Estimation in Differential Equations,” *SIAM Journal on Scientific and Statistical Computing*, 3, 28–46. [1]
- Wang, L., and Cao, J. (2012), “Estimating Parameters in Delay Differential Equation Models,” *Journal of Agricultural, Biological, and Environmental Statistics*, 17, 68–83. [1,4,8]
- Wang, L., Wang, S., and Bouchard-Côté, A. (2020), “An Annealed Sequential Monte Carlo Method for Bayesian Phylogenetics,” *Systematic Biology*, 69, 155–183. [2,5,6,7]
- Wood, S. N. (2017), *Generalized Additive Models: An Introduction With R*. Boca Raton: Chapman and Hall/CRC. [3]
- Zhang, T., Yin, Q., Caffo, B., Sun, Y., Boatman-Reich, D. (2017), “Bayesian Inference of High-Dimensional, Cluster-Structured Ordinary Differential Equation Models With Applications to Brain Connectivity Studies,” *The Annals of Applied Statistics*, 11, 868–897. [2]
- Zhou, Y., Johansen, A. M., and Aston, J. A. (2016), “Toward Automatic Model Comparison: An Adaptive Sequential Monte Carlo Approach,” *Journal of Computational and Graphical Statistics*, 25, 701–726. [2,7]