THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Simulation-based parameter inference methods based on data-conditional simulation of stochastic dynamical systems

Petar Jovanovski

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Division of Applied Mathematics and Statistics
Department of Mathematical Sciences
Chalmers University of Technology and the University of Gothenburg
Göteborg, Sweden 2025

Simulation-based parameter inference methods based on data-conditional simulation of stochastic dynamical systems
Petar Jovanovski
Göteborg 2025
ISBN 978-91-8103-279-6

Doktorshavhandlingar vid Chalmers tekniska högskola
Ny serie nr 5737
ISSN 0346-718X

Division of Applied Mathematics and Statistics
Department of Mathematical Sciences
Chalmers University of Technology and the University of Gothenburg
SE-412 96 Göteborg
Sweden
Telephone +46 (0)31 772 1000

Typeset with LaTeX
Printed by Chalmers Digital Printing
Göteborg, Sweden 2025

Simulation-based parameter inference methods based on data-conditional simulation of stochastic dynamical systems

Petar Jovanovski

Division of Applied Mathematics and Statistics
Department of Mathematical Sciences
Chalmers University of Technology and the University of Gothenburg

Abstract

Statistical inference for stochastic dynamical systems is a central problem in many scientific domains, yet is complicated by intractable likelihood functions, as well as partial and noisy observations. Simulation-based methods such as Approximate Bayesian Computation (ABC) offer a general route to Bayesian inference in this setting, but standard algorithms rely on *myopic* simulation methods that are unconditional on the data, and consequently suffer from low acceptance rates. In *Paper I* we introduce a data-conditional simulator for discretely observed stochastic differential equations (SDEs). The method leverages lookahead strategies and smoothing via backward simulation to accelerate ABC-Sequential Monte Carlo. By guiding the simulated trajectories toward the data, it substantially increases acceptance rates and accelerates convergence to the posterior distribution. In *Paper II* we target chemical reaction networks described by the chemical Langevin equation, a nonlinear SDE with multiplicative, non-commutative noise that poses challenges for simulation and inference. We extend the data-conditional simulator to partially observed systems with measurement noise, allowing trajectories to be guided toward the data in this more realistic setting. Moreover, we design a novel splitting scheme for the numerical solution of SDEs that preserves state space, densities, and oscillatory behavior, and enables robust inference even with large integration steps where Euler–Maruyama fails. In *Paper III*, we improve chain mixing and reduce the rejection rate in ABC-Markov Chain Monte Carlo. Because chains are typically initialized without knowledge of the posterior's shape, we introduce a data-conditional extension of ABC-MCMC that eases initialization and increases acceptance rates.

Keywords: approximate Bayesian computation; chemical reaction networks; sequential Monte Carlo; splitting methods; stochastic differential equations.

List of publications

This thesis is based on the following papers:

I. **Jovanovski, P.**, Golightly, A., Picchini, U. (2024). Towards data-conditional simulation for ABC inference in stochastic differential equations. *Bayesian Analysis*, 1(1):1–31. doi: 10.1214/24-BA1467

II. **Jovanovski, P.**, Golightly, A., Picchini, U., Tamborrino, M. (2025). Simulation-based inference using splitting schemes for partially observed diffusions in chemical reaction networks. *arXiv preprint* arXiv:2508.11438. `https://arxiv.org/abs/2508.11438`

III. **Jovanovski, P.**, Picchini, U. (2025). Enhancing ABC-MCMC with data-conditional proposals for stochastic nonlinear models. *Manuscript*.

Author contributions

I. Conceived and developed the data-conditional simulator for discretely observed SDEs, embedded it within ABC–SMC, designed the learning of summary statistics within ABC-SMC, implemented the methods, designed and ran all simulation studies, analyzed results, and drafted and revised the manuscript.

II. Extended the data-conditional simulation framework to partially observed CRNs with measurement noise, developed the splitting scheme, implemented the methods, designed and ran all simulation studies, analyzed results, and drafted and revised the manuscript.

III. Proposed the data-conditional extension of ABC–MCMC, implemented the methods, designed and ran all simulation studies, analyzed results, and drafted and revised the manuscript.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Umberto Picchini, for his continuous support and readiness to discuss problems whenever they arose throughout my PhD journey. His constructive criticism not only guided this work but also fostered my growth as a researcher. I am also grateful to my co-supervisor, Moritz Schauer, and my examiner, Petter Mostad, for their continued guidance and support. I would like to thank Annika Lang, Johan Tykesson, Irina Pettersson, Anders Södergren, and Marija Cvijović for their support and guidance. I am also grateful to the Operations Support team at Mathematical Sciences. I also wish to thank all my colleagues, past and present, at the department for fostering a motivating and pleasant environment. I thank my friend and *kontorskamrat*, Eric, for his camaraderie and our many conversations on life, philosophy, and mathematics. I am grateful to the Chalmers AI Research Centre for funding my project, including conference travel and research visits.

I am thankful to my collaborators, Andrew Golightly (Durham University, UK) and Massimiliano Tamborrino (Warwick University, UK), whose contributions and insights greatly shaped this work. I especially thank Massimiliano for welcoming me during my research visits to Warwick University, and for the stimulating discussions and helpful support he provided throughout my stay.

Finally, I am deeply grateful to my family and friends for the everyday conversations, time together, visits, and calls.

Petar Jovanovski
Göteborg, 2025

Contents

1 Introduction

Stochastic dynamical systems arise in diverse scientific disciplines to model the evolution of systems subject to inherent randomness (Fuchs, 2013). From molecular interactions in chemical reaction networks (Wilkinson, 2018; Schnoerr et al., 2017) and fluctuating asset prices in financial markets (Shreve et al., 2004; Cont and Tankov, 2003) to the spread of infectious diseases (Andersson and Britton, 2012; Allen, 2008) and the activity of neural systems under random synaptic fluctuations (Dayan and Abbott, 2005; Greenwood and Ward, 2016), randomness is a fundamental feature that cannot be ignored. To study such phenomena, scientists represent them as mathematical models, some of which are the focus of this thesis. While such models provide a powerful framework for representing complex phenomena, they are idealizations of real-world processes: the underlying dynamics are modeled as continuous in time, yet data are available only at a finite set of observational times. Often, these observations are separated by intervals that are large in relation to the system's characteristic timescales, meaning that much of the fine-scale dynamical behavior is unobserved. Worse yet, observations may be corrupted by measurement noise or restricted to a subset of the system's components, meaning that not all dimensions of a multidimensional system are observed throughout an experiment (King et al., 2016; Roberts and Stramer, 2001). This situation is commonly referred to as the regime of partial observations and is pervasive across scientific applications (King et al., 2016; Roberts and Stramer, 2001). Examples include fluorescence assays in chemical reaction networks (Schnoerr et al., 2017; Loskot et al., 2019), to the stochastic Morris–Lecar model in neuroscience, where voltage and potassium activation are modeled but only voltage is observed (Ditlevsen and Samson, 2014), to portfolio optimization under stochastic volatility in finance, where the dynamics of a risky asset follow a stochastic volatility model but only discrete stock prices are observed (Pham and Quenez, 2001). Although the primary focus of this thesis is on continuous-time models, particularly stochastic differential equations (SDEs), the methods and ideas extend naturally to certain discrete-time stochastic systems, and such examples are included in *Paper III* to illustrate their broader applicability.

In these systems, the state at any given time is a random variable whose evolution is often assumed to be Markovian: the distribution of its future state $X(t)$, conditional on the current state $X(s)$, is independent of the past. This evolution is described by the transition density $p(X(t) \mid X(s))$, which specifies the probability of transitioning to $X(t)$ from $X(s)$ over a time interval of length $t - s$. In a parametric setting, these densities depend on unknown parameters that govern the system's behavior, such as reaction rate constants in biochemical networks, volatility in financial models, or synaptic strengths in neural systems. When the dynamics are fully observed without measurement noise, the Markov property implies that the likelihood function is given by the product of the successive transition densities. When the transition density is available in closed form, the resulting likelihood function can be evaluated exactly, enabling frequentist approaches such as maximum likelihood estimation or Bayesian inference via exact likelihood computation (Fuchs, 2013; Iacus, 2008). Closed-form transition densities are available only for a few special cases, such as the Ornstein–Uhlenbeck process, geometric Brownian motion, and the Cox–Ingersoll–Ross (CIR) process (Iacus, 2008; Øksendal, 2013). However, for the vast majority of nonlinear SDEs, the transition densities are either unknown or analytically intractable, rendering exact likelihood evaluation infeasible (Kloeden and Platen, 1992). As previously mentioned, observations may also be restricted to part of the system and may be corrupted by measurement noise. In such cases, the observed data are not Markovian, even if the latent dynamics are. The likelihood then involves integrating over the latent trajectory, which is a high-dimensional and typically analytically intractable computation (Durbin and Koopman, 2012). This motivates the use of filtering-based approaches (e.g., Kalman filters for linear-Gaussian models, particle filters for general nonlinear models) to approximate the likelihood (Doucet et al., 2001; Andrieu et al., 2010; Särkkä and Svensson, 2023).

Despite these challenges, estimating the underlying parameters is essential, as they encode mechanistic properties and determine the system's qualitative and quantitative dynamics. In pharmacokinetics/pharmacodynamics, patient-specific absorption and clearance rates govern optimal dosing regimens (Lavielle, 2014); in stochastic epidemic models, transmission and recovery rates shape the timing and size of epidemic peaks (Britton et al., 2019); in stochastic volatility models, such as Heston (Heston, 1993), parameters determine pricing and hedging strategies for complex derivatives. Across such fields, accurate parameter estimates are therefore central to reliable predictions and sound decision-making in applied contexts. Because of the central role of parameter estimation in applications, the intractability of the likelihood has spurred a rich line of methodological developments. Broadly, these include approximate likelihood methods (Florens-Zmirou, 1989; Kessler, 1997; Ozaki, 1992; Shoji and Ozaki, 1998; Aït-Sahalia, 2002; Aït-Sahalia, 2008; Pilipovic et al., 2024), methods

based on bridges and data augmentation (Pedersen, 1995; Elerian et al., 2001; Eraker, 2001; Durham and Gallant, 2002; Delyon and Hu, 2006; Beskos et al., 2006; Golightly and Wilkinson, 2008, 2011, 2015; van der Meulen and Schauer, 2017), filtering-based approaches (Andrieu et al., 2010; Del Moral and Murray, 2015; Särkkä and Svensson, 2023), and, more recently, simulation-based inference techniques such as approximate Bayesian computation (ABC) (Toni et al., 2009; Liepe et al., 2014; Picchini, 2014), the latter forming the focus of this thesis. Our contribution lies at the interface between simulation-based inference (SBI) and numerical methods for stochastic models, targeting cases where likelihoods are intractable but forward simulation is possible.

Among the various SBI techniques (Cranmer et al., 2020), ABC has emerged as a particularly appealing choice for stochastic dynamical systems (Toni et al., 2009). ABC replaces explicit likelihood evaluation with a comparison between simulated and observed data, retaining the Bayesian paradigm, while circumventing the need for tractable likelihood functions. In practice, comparisons are often made between summaries of the simulated and observed data, and the performance of ABC algorithms depends critically on the choice of summary statistics and discrepancy measures (Blum et al., 2013; Fearnhead and Prangle, 2012; Prangle, 2018, 2017). In the context of high-dimensional models such as time-series, ABC has been applied in a range of settings (Picchini, 2014; Drovandi et al., 2016; Picchini and Forman, 2016; Jasra, 2015; Tancredi, 2019; Martin et al., 2019; Maybank et al., 2017; Kypraios et al., 2017; Buckwar et al., 2020; Ditlevsen et al., 2023; Samson et al., 2025).

In its most general form, an ABC algorithm relies on four key elements (Sisson et al., 2018): a *forward* simulator to generate synthetic data under candidate parameters, a proposal distribution for those parameters, a summary statistic function to reduce data to informative low-dimensional features, and a discrepancy measure to compare simulated and observed summaries. Among these, the forward simulator plays a central role in shaping the quality and efficiency of inference. For example, default but overly simplistic simulators are sometimes employed, which can yield inference of very low quality, particularly in challenging settings such as hypoelliptic diffusions (Buckwar et al., 2020). In continuous-time models, it is typically necessary to integrate the system's dynamics numerically, often augmenting low-frequency observations with simulated intermediate states by time-discretizing the observation intervals. Finer discretizations yield more accurate approximations to the underlying SDE, but at the cost of increased computation. The choice of numerical scheme and time-step size thus has a direct impact on both the precision of the trajectories and the feasibility of ABC. Beyond numerical considerations, the simulator of a stochastic system produces different trajectories even when run at identical parameter values. While this variability is an inherent feature of the model, in

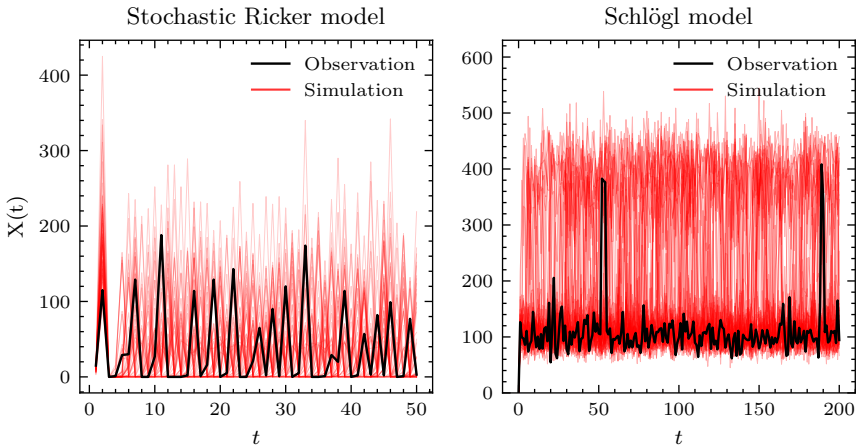

Figure 1.1: Illustration of randomness in trajectories. Left: stochastic Ricker model (ecology). With fixed parameter values, we draw one trajectory (black) to represent the observation and then generate 50 additional trajectories under the same parameter (red, translucent). Right: Schlögl model (systems biology), a bistable reaction network. Using the same procedure, one observed trajectory (black) and 50 simulated replicates (red).

ABC it can lead to apparent mismatches between simulated and observed data that arise from simulation noise rather than implausible parameters. Figure 1.1 illustrates the variability of trajectories under the same parameter.

To address the variability introduced by stochastic simulation in ABC, in this thesis we develop a *data-conditional simulation framework* for stochastic dynamical systems, with particular emphasis on SDEs. The framework builds on, and extends, the standard ABC–Sequential Monte Carlo (ABC–SMC, Toni et al., 2009) and ABC–Markov Chain Monte Carlo (ABC–MCMC, Marjoram et al., 2003) algorithms. Data-conditional simulation is applied in three settings: (i) exactly observed SDE models, (ii) partially observed SDEs with measurement noise, both using a data-conditional ABC–SMC variant, and (iii) continuous- and discrete-time stochastic models with measurement noise, using a data-conditional ABC–MCMC variant. Rather than simulating trajectories solely from the forward model given proposed parameters, the data-conditional simulator conditions the simulation on the observed data, steering paths toward regions of the state space consistent with the observed data. This conditioning has important implications: by altering the simulator, we also alter the distribution of the resulting summary statistics. In ABC–SMC, this changes the particle weights; in ABC–MCMC, it changes the acceptance probability. Crucially, these quantities now depend on new intractable densities: the original likelihood of

the summary statistics, and the likelihood of the summary statistics conditional on the observed data. A central aim of this thesis, beyond the construction of the data-conditional simulators themselves, is to address this challenge by designing efficient and accurate approximations for these terms, ensuring that the improved similarity between simulated and observed trajectories from data conditioning, and the resulting higher ABC acceptance rates, is not offset by excessive post-hoc correction costs.

Complementing these developments, we also address the role of numerical discretization in ABC for continuous-time stochastic systems. For chemical reaction networks in particular, in *Paper II* we design a structure-preserving splitting scheme, a type of numerical method that maintains key properties of the model such as state space, invariant distribution, and oscillatory behavior, allowing accurate inference even with relatively coarse time steps. This reduces computational cost while preserving the underlying dynamics, further enhancing the practical efficiency of simulation-based inference. Crucially, its numerical stability extends across the entire admissible parameter space, ensuring that synthetic data can be reliably generated for any admissible parameter proposal. This stability is vital not only for inference but also for modern approaches to constructing summary statistics automatically. Recent work has focused on learning summaries from data using regression models and neural networks (Fearnhead and Prangle, 2012; Jiang et al., 2017; Prangle, 2018; Chen et al., 2021a), which require large volumes of accurately simulated trajectories during training. If simulations become unstable or inaccurate in parts of the parameter space, learned summaries risk encoding numerical artifacts rather than genuine model features. By ensuring stable simulations everywhere, the proposed discretization provides a solid foundation for such automated methods, increasing their robustness and reliability within ABC for SDEs.

2 Bayesian inference

Bayesian statistics traces its origins to the 18th century with Reverend Thomas Bayes, who introduced a method to update the probability of a hypothesis in light of new evidence. His seminal essay, "An Essay towards solving a Problem in the Doctrine of Chances", was published posthumously in 1763 by his friend Richard Price (Bayes, 1763). Shortly after, Pierre-Simon Laplace independently developed and extended Bayes' ideas into a general mathematical framework, and applied them to solve various statistical problems. For much of the 20th century, however, Bayesian methods were viewed with skepticism by the statistical community, largely due to their reliance on subjective beliefs encoded in the prior distribution, as well as the significant computational demands they imposed. This perception began to shift in the late 20th and early 21st centuries due to the advent of modern computing and Markov Chain Monte Carlo (MCMC) algorithms, which made Bayesian inference feasible even for complex, high-dimensional models (Metropolis et al., 1953; Gelfand and Smith, 1990). Today, Bayesian methods are central to modern statistics and machine learning.

At the heart of Bayesian statistics is Bayes' theorem, which describes how to update beliefs in light of new data. We use $y^{\text{o}} \in \mathcal{Y}$ to denote the observed data, where $\mathcal{Y} \subset \mathbb{R}^n$ denotes the observation space. Let $\theta \in \Theta$ denote the parameter vector of interest, where $\Theta \subset \mathbb{R}^p$ is the parameter space. We assume that there exists a value $\theta^{\text{o}} \in \Theta$ (unknown) that corresponds to the generation of the specific data y^{o}. By treating θ as a random variable, the inference task is to determine its distribution conditional on the information provided by y^{o}, in order to quantify the uncertainty around θ^{o}. The *posterior distribution* $\pi(\theta \mid y^{\text{o}})$ is given by

$$\pi(\theta \mid y^{\text{o}}) = \frac{p_\theta(y^{\text{o}})\pi(\theta)}{p(y^{\text{o}})}, \tag{2.1}$$

where $p_\theta(y^{\text{o}})$ is the *likelihood* of the data given θ, $\pi(\theta)$ is the *prior distribution*, and $p(y^{\text{o}}) = \int_\Theta p_\theta(y^{\text{o}})\pi(\theta), \mathrm{d}\theta$ is the *marginal likelihood* (also called the "evidence"), which ensures proper normalization of the posterior. The marginal likelihood is often intractable in practice, especially for high-dimensional or non-conjugate

models (i.e., models where the posterior distribution does not belong to the same family as the prior). To circumvent this, MCMC methods offer a powerful class of algorithms for sampling from the posterior distribution without requiring explicit computation of the marginal likelihood. This is achieved by constructing a Markov chain whose stationary distribution coincides with the posterior.

One of the most commonly used MCMC methods is the *Metropolis–Hastings algorithm* (Hastings, 1970). Assume the current state of the chain is θ, then a new candidate value θ^* is proposed from a proposal distribution $g(\theta^* \mid \theta)$, and accepted with probability

$$\alpha(\theta, \theta^*) = \min\left(1, \frac{p_{\theta^*}(y^o)\,\pi(\theta^*)\,g(\theta \mid \theta^*)}{p_\theta(y^o)\,\pi(\theta)\,g(\theta^* \mid \theta)}\right). \tag{2.2}$$

This acceptance rule ensures that the chain has the posterior $\pi(\theta \mid y^o)$ as stationary distribution. Notably, knowledge about the intractable marginal likelihood $p(y^o)$ is not necessary, as this cancels out in the ratio. However, the algorithm still requires a pointwise evaluation of the likelihood $p_\theta(y^o)$, which may be unavailable or intractable in many practical models, such as CRNs and other SDE models. To circumvent this limitation, researchers typically resort to *simulation-based inference* (SBI) methods (Cranmer et al., 2020). These methods approximate the likelihood, posterior, or both, using only simulations from the underlying stochastic model. By requiring only the ability to simulate data, SBI methods have opened new possibilities for statistical inference that were previously inaccessible due to analytical or computational intractability, as summarized in Section 2.1.

In the following sections, we describe several widely used SBI approaches, with primary focus on approximate Bayesian computation (ABC) methods including ABC rejection, ABC importance sampling, ABC-MCMC and ABC-Sequential Monte Carlo (ABC-SMC), as well as a brief discussion of *synthetic likelihood* (SL) methods. We also discuss specific considerations arising from partial observability of diffusion processes, measurement noise, and sparsely sampled data, which are common scenarios in biochemical applications.

2.1 Simulation-based inference

In Statistics, an often-mentioned problem is that of the likelihood being "intractable". This means that $p_\theta(y^o)$ cannot be evaluated pointwise in any θ or y^o, as its expression is unavailable in closed form. In fact, for models of realistic complexity, $p_\theta(y^o)$ is a multidimensional integral, such as $p_\theta(y^o) = \int p_\theta(x, y^o)\,\mathrm{d}x$ for some high-dimensional variable x entering the probabilistic

model generating observations y^{o}. In the absence of a fast to evaluate and exact expression for $p_\theta(y^{\text{o}})$, both frequentist and Bayesian inference cannot be carried out. A large number of solutions have been proposed to produce some form of approximate inference, and if we restrict to the Bayesian framework a relatively recent review is in Martin et al. (2024). To overcome such a daunting computational problem, the most intuitive approach would be that of simplifying the hypothesized data-generating model (DGM) enough that exact inference methods can be applied. However, this simplification may come at the expense of realism, when accepting to use a suboptimal but tractable DGM, in place of a more realistic but intractable one. If we still decide to stick to a realistic DGM, one not excessively oversimplified for mathematical convenience, then we may happen to agree with John Tukey when he says (Tukey, 1962):

> *Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise.*

SBI, also known as likelihood-free inference, emerged as an attempt at bypassing the unavailability of a closed-form expression for $p_\theta(y)$ (for a generic $y \in \mathcal{Y}$) with the ability to *sample* from $p_\theta(y)$ which, as we will see, translates to the ability to *simulate* synthetic datasets $y \sim p_\theta(y)$ (it is standard in the literature to abuse notation and write $y \sim p_\theta(y)$ to denote a sample from the distribution of Y, even though formally $Y \sim p_\theta$ and y denotes a realization.) The crucial step is to regard the simulator $\mathcal{M}(\theta)$ as a deterministic or stochastic mechanism for generating data, conditional on a given parameter value θ, along with optional covariates and pseudorandom streams. Suppose we fix a parameter value θ^*. Running the simulator at this value produces an output $y^* = \mathcal{M}(\theta^*)$. This property defines the so-called *implicit models* (Diggle and Gratton, 1984), to express the fact that $p_\theta(y)$ is not available explicitly, but only implicitly via simulation. In Diggle and Gratton (1984) an SBI scheme was introduced to approximate the likelihood function, however, the first use of SBI to approximate the posterior was sketched in Rubin (1984) (although this was just a conceptual thought experiment). Besides the thought experiment of Rubin (1984), the first proper applications of such scheme emerged in the 1980s in population genetics, with the works of Tavaré et al. (1997); Pritchard et al. (1999), and these were later referred to as Approximate Bayesian Computation (ABC[1]) and popularised by Beaumont et al. (2002). In ABC an approximation of the posterior distribution was obtained by accepting or rejecting proposed parameters based on comparisons between simulated data from the model and observed data. These developments established ABC as a method for likelihood-free inference, particularly useful where simulating data is computationally inexpensive, but

[1]See Tavaré (2018) for an account on the origin of the acronym as well as the origins of the ABC methodology.

computing a likelihood is challenging or infeasible. We will now summarize some important contributions in SBI, and we will return to ABC in detail in Sections 2.1.3-2.1.5.

SBI has now grown into a considerable body of literature, mostly within the Bayesian paradigm, aimed at approximating the posterior $\pi(\theta \mid y^{\text{o}})$. Here, the observed data y^{o} is viewed as a specific instance $y = y^{\text{o}}$ generated from $p_{\theta^{\text{o}}}(y)$ for some true parameter θ^{o}. This, however, only holds under the unrealistic assumption that $\mathcal{M}(\theta^{\text{o}})$ is the true data-generating model, which is a restrictive assumption increasingly challenged in the literature on model misspecification (see references below). Some key SBI methods, notably "rejection ABC," simulate N independent parameter–data pairs $(\theta^{1:N}, y^{1:N})$. These pairs are then "refined" or subsampled to produce an approximation to the posterior $\pi(\theta \mid y^{\text{o}})$ (rejection ABC is the simplest form of ABC, described in Section 2.1.3). Depending on *how* θ^i has been simulated/proposed (e.g., MCMC, importance sampling, sequential Monte Carlo, normalizing flows), many different methods have been constructed, while typically y^i is always resulting from running the (forward) simulator $\mathcal{M}(\theta^i)$, a notion we challenge in this thesis where we propose other ways to simulate data. For example, SBI methods are often initialized with prior simulations $\theta^i \sim \pi(\theta)$ (i.i.d.), and as more information is acquired from the simulations, the proposals are later modified so that $\theta^i \sim g(\theta)$, still i.i.d. for $i = 1, \ldots, N$. More recently, exciting developments in machine learning have brought *neural conditional estimation* (NCDE) at the forefront of SBI. NCDE uses neural networks to learn conditional densities, see the review in Cranmer et al. (2020). More specifically, this strand of research, named *neural* SBI in Wang et al. (2024), uses normalizing flows (Rezende and Mohamed, 2015; Papamakarios et al., 2021) to approximate the likelihood function in implicit models (Papamakarios et al., 2019; Chen et al., 2021b), the posterior distribution (Papamakarios and Murray, 2016; Greenberg et al., 2019; Durkan et al., 2020; Chen et al., 2021b; Miller et al., 2021; Delaunoy et al., 2022), and simultaneously the likelihood and the posterior (Wiqvist et al., 2021; Radev et al., 2023). Comparisons between some of these methods are available, e.g., in Greenberg et al. (2019), Lueckmann et al. (2021) and Häggström et al. (2024). Moreover, NCDE has been used both to sequentially refine inference conditionally on a specific observed data set y^{o}, but also in an "amortized" way (i.e. without dependence on a specific y^{o}), see the review in Zammit-Mangion et al. (2024). Recently, thanks to the aid provided by neural network constructions, the field is expanding to accommodate generalized Bayesian approaches for model misspecification (Matsubara et al., 2022; Pacchiardi et al., 2024; Weerasinghe et al., 2025).

SBI methods that do not use neural networks to approximate conditional densities have been called *statistical* SBI in Wang et al. (2024), and obviously have a

longer history, and have been more extensively studied, also in terms of theoretical guarantees. On the other hand, neural SBI can require orders of magnitude fewer simulations (Lueckmann et al., 2021). Since our work is placed within statistical SBI, in the next sections we devote special attention to this class of SBI methods, especially for ABC and synthetic likelihoods, since these both have a prominent role in our work. Producing a detailed classification of SBI methods is a difficult task, as the methodology is fertile and rapidly expanding. We refer the reader to `https://simulation-based-inference.org/` for a large, continuously updated, resource collecting publication within SBI.

2.1.1 Statistical SBI

Since the seminal paper of Tavaré et al. (1997), ABC has evolved into a broad class of methods for approximating the posterior $\pi(\theta \mid y^{\text{o}})$. Several ABC variants proposed in the literature are linked not only to different sampling schemes, but also to deeper inferential questions. For example, ABC can be viewed as a way to robustify inference in the presence of model misspecification (see the recent connection between ABC and generalized Bayesian inference Järvenpää et al., 2025), or even as a method that yields exact inference for a noisy ("perturbed") version of the true unknown model (Wilkinson, 2013). In this chapter, however, we focus on computational aspects: specifically, we consider several ways of sampling from approximations to $\pi(\theta \mid y^{\text{o}})$, not only through ABC but also via related SBI methodologies.

In ABC we consider the following approximation to the posterior

$$\pi_\epsilon(\theta \mid y^{\text{o}}) \propto \pi(\theta) \int_{\mathcal{Y}} K_\epsilon \left(\|y - y^{\text{o}}\| \right) p_\theta(y) \, \mathrm{d}y, \tag{2.3}$$

for a non-negative kernel $K_\epsilon \left(\|y - y^{\text{o}}\| \right)$, with bandwidth $\epsilon > 0$ and assigning larger weights to values of $y \in \mathcal{Y}$ that are close to data $y^{\text{o}} \in \mathcal{Y}$ (for some distance metric $|| \cdot ||$). We discuss ABC more in detail in sections 2.1.3-2.1.5. Here we only want to emphasize the fact that the quantity $\int_{\mathcal{Y}} K_\epsilon \left(\|y - y^{\text{o}}\| \right) p_\theta(y) \, \mathrm{d}y$ can be seen as a non-parametric approximation to the likelihood $p_\theta(y^{\text{o}})$, and the approximation improves as $\epsilon \to 0$. A large body of research has been devoted to mitigating the curse of dimensionality in ABC, stemming from its nonparametric nature, where the observation space $\mathcal{Y}$ may be high-dimensional. The first and most important mitigation factor being the reduction of the dimension of the observation space to one of smaller dimension, which can be achieved with the introduction of data summarization, as discussed in Section 2.1.6. With data summarization, instead of considering inference based on observed data y^{o}, the inference problem is based on its summaries $s^{\text{o}} = S(y^{\text{o}}) \in \mathcal{S}$, for a suitable choice a summary function $S : \mathcal{Y} \to \mathcal{S}$, mapping the

data space $\mathcal{Y}$ into summary space $\mathcal{S} \subset \mathbb{R}^k$ (with $k \ll \dim(y)$). This allows to drastically reduce the value of ϵ towards obtaining accurate inference from $\pi_\epsilon(\theta \mid s^o)$ via Monte Carlo procedures, conditionally on $S(\cdot)$ being "informative" for θ. However, dependence of the problem on a small enough ϵ implies that Monte Carlo procedures are dependent on acceptance-rejection steps, typically translating in high rejection rates. For these reasons, many advanced proposal samplers have been suggested in ABC, and we describe some of those in this section.

2.1.2 (Bayesian) synthetic likelihoods

Besides ABC, another important statistical SBI methodology is the *synthetic likelihood* (SL) approach (Wood, 2010; Price et al., 2018), which (unlike ABC) proposes a parametric approximation of the summaries density $p(s^o \mid \theta)$. SL models the distribution of the k-dimensional summary statistics under the forward simulator with a k-dimensional multivariate normal density $\phi_k(\mu_\theta, \Sigma_\theta)$, for a given parameter θ. In practice, conditional on a parameter value θ^*, one forward-simulates M synthetic datasets $y^{1:M}$, i.i.d. from $\mathcal{M}(\theta^*)$. For each y^i, the corresponding summary statistics $s^i = S(y^i)$ are computed. From the resulting M vectors of summaries, we can then construct the usual sample-based unbiased estimates of the mean, $\hat{\mu}_{\theta^*}$, and the covariance matrix, $\hat{\Sigma}_{\theta^*}$ (both depending on the specific θ^*):

$$\hat{\mu}_{\theta^*} = \frac{1}{M}\sum_{i=1}^{M} s^i, \qquad \hat{\Sigma}_{\theta^*} = \frac{1}{M-1}\sum_{i=1}^{M}(s^i - \hat{\mu}_{\theta^*})(s^i - \hat{\mu}_{\theta^*})^\top,$$

with $\top$ denoting transposition. We can then evaluate the resulting density at the summary statistics of the data s^o, this yielding the "synthetic likelihood" $\hat{p}_{\theta^*}(s^o) = \phi(s^o; \hat{\mu}_{\theta^*}, \hat{\Sigma}_{\theta^*})$. This estimate can then be plugged into standard MCMC algorithms for Bayesian inference (Price et al., 2018; Picchini et al., 2023), approximate maximum likelihood estimation (Wood, 2010), or variational Bayes approximations (Ong et al., 2018). In Price et al. (2018) it was studied how MCMC sampling targeting $\hat{\pi}(\theta \mid s^o) \propto \hat{p}_\theta(s^o)\pi(\theta)$ can be accurate and with a lower rejection rate than, e.g., ABC-MCMC algorithms (described in Section 2.1.4), and also less sensitive to the curse of dimensionality for an increasing dimension k of the summaries $S(\cdot)$. The most critical problems are (i) the construction of summary statistics that are both informative for θ and approximately Gaussian distributed, although recent works are able to relax these aspects (Fasiolo et al., 2018; An et al., 2020), and (ii) the requirement to simulate M datasets (often in the order of thousands) at each value of θ, making the methodology computer-intensive (also here research is available to mitigate the effort Priddle et al., 2022, Picchini et al., 2023).

2.1.3 Rejection ABC and importance sampling ABC

We now consider ABC samplers. We start with the oldest, and simpler, ABC sampler, the *ABC rejection* algorithm (Tavaré et al., 1997; Pritchard et al., 1999). This algorithm proceeds as follows: first, a parameter $\theta^* \sim \pi(\theta)$ is sampled from the prior distribution. Given θ^*, synthetic data y^* is forward simulated from $\mathcal{M}(\theta^*)$. A low-dimensional summary statistic $s^* = S(y^*)$ is then computed, and we can write that (implicitly) $s^* \sim p_{\theta^*}(s)$, where $S : \mathcal{Y} \to \mathcal{S}$ is a k-dimensional vector of summary statistics (with $k \ll \dim(y^{\text{o}})$), see Section 2.1.1. Let $s^{\text{o}} = S(y^{\text{o}})$ denote the summary statistic of the observed data. Using a distance metric $\|\cdot\|$, the distance $\|s^* - s^{\text{o}}\|$ is computed. If this distance is less than a predefined threshold $\epsilon > 0$, the proposed parameter θ^* is accepted as an approximate draw from the posterior $\pi_\epsilon(\theta \mid s^{\text{o}})$. This scheme can be iterated N times to result in N independent draws from $\pi_\epsilon(\theta \mid s^{\text{o}})$. It is important to emphasize that ABC involves multiple layers of approximation. First, the observed and simulated data are replaced by low-dimensional summary statistics. The summary function plays a critical role in the quality of inference, and its choice will be the subject of a dedicated section. Second, as an illustration, consider the common indicator (uniform) ABC kernel given by $K_\epsilon(\|s^* - s^{\text{o}}\|) \equiv \mathbb{I}(\|s^* - s^{\text{o}}\| \leq \epsilon)$, which is non-zero only when $\|s^* - s^{\text{o}}\| \leq \epsilon$, then ABC accepts simulated summaries that lie within an ϵ-ball around s^{o}, introducing a further approximation. By recalling (2.3), the ABC posterior can be expressed as

$$\pi_\epsilon(\theta \mid s^{\text{o}}) \propto \int_{\mathcal{S}} \mathbb{I}(\|s - s^{\text{o}}\| \leq \epsilon)\; p_\theta(s)\, \pi(\theta)\, \mathrm{d}s. \tag{2.4}$$

If the summary statistic $S(\cdot)$ is highly informative for θ, then the ABC posterior $\pi_\epsilon(\theta \mid s^{\text{o}}) \approx \pi_\epsilon(\theta \mid y^{\text{o}})$, with equality holding only when $S(\cdot)$ is a sufficient statistic. Additionally, as $\epsilon \to 0$, we recover the true posterior conditioned on the summary statistic, that is, $\pi_\epsilon(\theta \mid s^{\text{o}}) \to \pi(\theta \mid s^{\text{o}})$. Therefore, when using a small threshold ϵ and an informative summary statistic $S(\cdot)$, ABC can yield a reasonable approximation to the true posterior $\pi(\theta \mid y^{\text{o}})$. Choosing an appropriate value for ϵ involves a trade-off: reducing ϵ improves the accuracy of the posterior approximation (assuming enough accepted samples are obtained), but increases computational cost due to a higher rejection rate of proposed parameters. One could also sample from an alternative proposal distribution $g(\theta)$, rather than from the prior $\pi(\theta)$. However, this requires knowing the bounding constant $C > 0$ such that $\pi(\theta) \leq Cg(\theta)$, which is difficult to determine in practice.

A more flexible alternative to ABC rejection is *ABC importance sampling* (ABC-IS, Chapter 4 in Sisson et al., 2018), as it naturally accommodates sampling from a proposal distribution other than the prior. Consider reparametrising the

problem in terms of proposing jointly $(\theta, s) \sim g(\theta, s)$ on the joint space $\Theta \times \mathcal{S}$ of parameters and summary statistics, and take $g(\theta, s) = p_\theta(s)\, g(\theta)$. Therefore each $\theta^* \sim g(\theta)$ is assigned a weight

$$\frac{\pi_\epsilon(\theta, s \mid s^{\mathrm{o}})}{g(\theta, s)} \propto \frac{\mathbb{I}\left(\|s - s^{\mathrm{o}}\| \leq \epsilon\right)\, p_\theta(s)\, \pi(\theta)}{p_\theta(s)\, g(\theta)} = \frac{\mathbb{I}\left(\|s - s^{\mathrm{o}}\| \leq \epsilon\right)\, \pi(\theta)}{g(\theta)}, \tag{2.5}$$

where

$$\pi_\epsilon(\theta, s \mid s^{\mathrm{o}}) \propto \mathbb{I}\left(\|s - s^{\mathrm{o}}\| \leq \epsilon\right)\, p_\theta(s)\, \pi(\theta).$$

Therefore, suppose we obtain (by repeated sampling) a population of N samples $(\theta^{1:N}, s^{1:N})$ such that $\mathbb{I}\big(\|s^i - s^{\mathrm{o}}\| \leq \epsilon\big) = 1$ for $i = 1, \ldots, N$. If we then assign the corresponding (unnormalized) weights $w^i \propto \pi(\theta^i)/g(\theta^i)$, we can resample N times with replacement from the weighted set $(\theta^{1:N}, w^{1:N})$ (discarding the summaries) to obtain a new set of N independent samples from $\pi_\epsilon(\theta \mid s^{\mathrm{o}})$. The chosen factorization $g(\theta, s) = p_\theta(s)\, g(\theta)$ allows the summaries likelihood term $p_\theta(s)$ to cancel in the expression above, thus making the procedure "likelihood-free". The idea of weighting accepted parameter proposals based on prior-to-proposal ratios, as in ABC-IS, naturally extends to more advanced population-based methods. In particular, *ABC Sequential Monte Carlo* (ABC-SMC) builds upon ABC-IS by operating on a population of weighted parameters (called particles), iteratively updating the proposal distribution and gradually reducing the threshold ϵ, allowing for more efficient exploration of the posterior. This procedure is carried out over multiple rounds, each round yielding independent samples from an ABC posterior that concentrates more closely around the true posterior $\pi(\theta \mid s^{\mathrm{o}})$. ABC-SMC, regarded as the state-of-art ABC sampler, is detailed in Section 2.1.5.

In parallel to this importance sampling-based line of development, another widely used class of methods is ABC-MCMC, which relies on constructing a Markov chain whose stationary distribution approximates the ABC posterior. Unlike importance sampling or SMC, ABC-MCMC does not rely on weighting, but instead uses an accept/reject mechanism embedded in an MCMC framework. However, unlike ABC rejection, ABC-IS and ABC-SMC which return independent posterior samples, ABC-MCMC instead returns correlated samples. In the following, we begin with ABC-MCMC and then proceed to ABC-SMC, both of which will serve as the foundation for the methodological contributions of this thesis.

2.1.4 ABC-MCMC

We alluded to the Metropolis-Hastings (MH) algorithm at the beginning of this chapter, and introduced its acceptance ratio in Equation (2.2). ABC variants of the MCMC algorithm were originally developed in Marjoram et al. (2003) (see

also Bortot et al., 2007; Wegmann et al., 2009; Ratmann et al., 2009; Sisson and Fan, 2011; Sisson et al., 2018 for more extensive discussions on ABC-MCMC). Below we follow the exposition as in Marjoram et al. (2003), where an indicator kernel is used, however a generic K_ϵ could equivalently be used. Similarly to ABC rejection and ABC-IS, the target distribution is the joint ABC posterior $\pi_\epsilon(\theta, s \mid s^{\mathrm{o}}) \propto \mathbb{I}(\|s - s^{\mathrm{o}}\| \leq \epsilon)\, p_\theta(s)\, \pi(\theta)$, defined on $\Theta \times \mathcal{S}$. On this space, the proposal distribution factorizes as

$$g((\theta, s), (\theta^*, s^*)) = q(\theta^* \mid \theta)\, p_{\theta^*}(s^*),$$

where $q(\theta^* \mid \theta)$ is the proposal distribution for the parameters, and $p_{\theta^*}(s^*)$ corresponds to the simulation of summary statistics from the model given the proposed parameter. As a result, the acceptance probability of a proposed move from (θ, s) to (θ^*, s^*) becomes $\min\left(1, \alpha((\theta, s), (\theta^*, s^*))\right)$, with

$$\begin{aligned}
\alpha((\theta, s), (\theta^*, s^*)) &= \frac{\pi_\epsilon(\theta^*, s^* \mid s^{\mathrm{o}})\, g((\theta^*, s^*), (\theta, s))}{\pi_\epsilon(\theta, s \mid s^{\mathrm{o}})\, g((\theta, s), (\theta^*, s^*))} \\
&= \frac{\mathbb{I}(\|s^* - s^{\mathrm{o}}\| \leq \epsilon)\, p_{\theta^*}(s^*)\, \pi(\theta^*)\, q(\theta \mid \theta^*)\, p_\theta(s)}{\mathbb{I}(\|s - s^{\mathrm{o}}\| \leq \epsilon)\, p_\theta(s)\, \pi(\theta)\, q(\theta^* \mid \theta)\, p_{\theta^*}(s^*)} \\
&= \frac{\mathbb{I}(\|s^* - s^{\mathrm{o}}\| \leq \epsilon)\, \pi(\theta^*)\, q(\theta \mid \theta^*)}{\mathbb{I}(\|s - s^{\mathrm{o}}\| \leq \epsilon)\, \pi(\theta)\, q(\theta^* \mid \theta)}.
\end{aligned} \tag{2.6}$$

Although the final expression resembles the standard MH ratio in terms of prior and proposal densities, the intractable likelihood terms are not evaluated directly. Instead, the acceptance probability includes the indicator functions that enforce proximity between the simulated and observed summary statistics. Notice, the indicator function at the denominator of (2.6) can be safely removed, as this corresponds to "accepted summaries", and as such this indicator equals 1. We embed this consideration into Algorithm 1, returning N correlated parameter samples from $\pi_\epsilon(\theta \mid s^o)$.

Algorithm 1 ABC-MCMC (s^{o}, $\varepsilon > 0$, N, initial $\theta^{(0)}$)

1: **for** $i = 1, \ldots, N$ **do**
2: Sample $\theta^* \sim q(\theta \mid \theta^{(i-1)})$, and simulate $s^* \sim p_{\theta^*}(s)$.
3: Calculate $\alpha = \min\left(1, \frac{\pi(\theta^*) q(\theta^{(i-1)} \mid \theta^*)}{\pi(\theta^{(i-1)}) q(\theta^* \mid \theta^{(i-1)})}\right) \mathbb{I}(\|s^* - s^{\mathrm{o}}\| \leq \epsilon)$.
4: Set $\theta^i = \theta^*$ with probability α, and $\theta^i = \theta^{i-1}$ otherwise.
5: **end for**
6: **Output:** Samples $\theta^{1:N}$.

ABC-MCMC achieves higher acceptance rates than ABC rejection or impor-

tance sampling when using the same threshold parameter ϵ, albeit at the cost of increased autocorrelation in the resulting Markov chain (Marjoram et al., 2003). More critically, the acceptance of a proposed move $\theta \to \theta^*$ depends not only on the prior and proposal densities, but also on the ability to generate a summary statistic $s^* \sim p_{\theta^*}(s)$ that lies within an ϵ-ball of the observed summary s^o. Such proposals are likely to be accepted frequently if they come from regions of high posterior density. However, in regions of low density the chain may repeatedly fail to produce acceptable summaries, causing it to remain stuck at the current state for many iterations. This results in poor mixing and can lead to biased posterior approximations or convergence issues. These challenges are particularly pronounced when using kernels such as the indicator function, which sharply reject distant proposals rather than downweighting them as a different ABC kernel K_ϵ would (e.g. a Gaussian kernel). The determination of ϵ, and strategies to let it vary, are considered in Bortot et al. (2007), Simola et al. (2021) and Vihola and Franks (2020).

Several extensions and improvements to ABC-MCMC have been proposed in the literature to address its well-known limitations (Bortot et al., 2007; Ratmann et al., 2007; Baragatti et al., 2013; Ratmann et al., 2009; Meeds et al., 2015; Kobayashi and Kozumi, 2015; Cabras et al., 2015; Kousathanas et al., 2016; Picchini, 2014). Our own contribution follows this line of work by focusing on improving the initial mixing and robustness of the sampler, particularly in challenging regions of the parameter space.

An alternative approach to ABC-MCMC that has gained widespread popularity is ABC-SMC. This class of methods builds directly on the ideas of ABC importance sampling and addresses many of the limitations of ABC-MCMC, by evolving a population of weighted particles through a sequence of intermediate distributions, while gradually reducing the tolerance ϵ in ways that are more intuitive that for ABC-MCMC. ABC-SMC combines the strengths of population-based inference with adaptive proposals and threshold schedules. In the next section, we introduce the standard ABC-SMC framework and lay the foundation for the improved version developed later in this thesis.

2.1.5 ABC-SMC

As alluded to in the previous section, ABC-SMC extends ABC-IS by evolving a population of weighted posterior samples, named "particles" in the SMC literature, through a sequence of intermediate distributions with decreasing thresholds. Let $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_R > 0$ denote a sequence of tolerance values, where R is the total number of "rounds", where each round corresponds to a corresponding posterior approximation. The target distribution at round r is

the ABC posterior

$$\pi_{\epsilon_r}(\theta \mid s^{\mathrm{o}}) \propto \mathbb{I}(\|s - s^{\mathrm{o}}\| \leq \epsilon_r)\, p_\theta(s)\, \pi(\theta).$$

At each round, a set of N particles are propagated and reweighted in a way that approximates the posterior distribution more closely as ϵ_r decreases. The sequence of distributions $\pi_{\epsilon_1}, \ldots, \pi_{\epsilon_R}$ thus guides the population toward regions of high posterior density, while maintaining diversity through perturbations of the particles. In the first round of ABC-SMC, particles are sampled independently from the prior distribution $\pi(\theta)$. For each sampled parameter θ_1^i, synthetic data is generated under the model, and a summary statistic s^i is computed. The parameter θ_1^i is retained if the simulated summary satisfies $\|s^i - s^{\mathrm{o}}\| \leq \epsilon_1$, and is then assigned a weight of $1/N$. The ABC posterior at this stage is approximated using the retained set of N weighted particles $(\theta_1^{1:N}, w_1^{1:N})$. This yields the empirical approximation

$$\pi_{\epsilon_1}(\theta \mid s^{\mathrm{o}}) \approx \sum_{i=1}^{N} w_1^i\, \delta_{\theta_1^i}(\theta),$$

where δ is the Dirac delta function. In subsequent rounds $r = 2, \ldots, R$, particles are no longer sampled from the prior, but instead are generated by perturbing the previous population. More precisely, a particle θ^* is sampled from the previous round's population $\theta_{r-1}^{1:N}$ with probability proportional to its normalized weight w_{r-1}^i, and then perturbed using a transition kernel $\mathcal{K}(\theta \mid \theta^*)$. The resulting perturbed parameter θ_r^i is accepted if the simulated summary statistic $s^i \sim p_{\theta_r^i}(s)$ satisfies the ABC condition $\|s^i - s^{\mathrm{o}}\| \leq \epsilon_r$. Once accepted, the importance weight of particle θ_r^i is computed as

$$w_r^i \propto \frac{\pi(\theta_r^i)}{\sum_{j=1}^{N} w_{r-1}^j\, \mathcal{K}(\theta_r^i \mid \theta_{r-1}^j)}, \tag{2.7}$$

and the weights are normalized to sum to one. This correction accounts for the fact that particles are sampled from a mixture distribution rather than directly from the prior. A common and practical choice for the transition kernel is the Gaussian one

$$\mathcal{K}(\theta \mid \theta^*) = \mathcal{N}(\theta; \theta^*, \Sigma_r),$$

where Σ_r is typically set to twice the weighted empirical covariance matrix of the particles in the previous round $\Sigma_r = 2 \times \mathrm{Cov}((\theta_{r-1}^{1:N}, w_{r-1}^{1:N}))$ (Beaumont et al., 2009; Toni et al., 2009). This choice helps maintain diversity in the population while preserving concentration around regions of high posterior density. Algorithm 2 outlines ABC-SMC. Other kernel choices and adaptive strategies producing more informed and efficient proposal kernels are in Filippi et al. (2013)

and Picchini and Tamborrino (2024).

Algorithm 2 ABC-SMC ($\epsilon_1 > \ldots > \epsilon_R$)

for $i = 1$ to N **do**
  while parameter not accepted **do**
    Sample parameter $\theta_1^i \sim g(\theta)$ and summary $s_1^i \sim p_{\theta_1^i}(s)$.
    if $\|s_1^i - s^{\mathrm{o}}\| \leq \epsilon_1$ **then**
      Accept θ_1^i and compute $w_1^i = \pi(\theta_1^i)/g(\theta_1^i)$.
    end if
  end while
end for
Normalize $w_1^{1:N}$.
for $r = 2$ to R **do**
  Compute particle covariance $\Sigma_r = 2 \times \mathrm{Cov}((\theta_{r-1}^{1:N}, w_{r-1}^{1:N}))$.
  for $i = 1$ to N **do**
    while parameter not accepted **do**
      Sample θ^* from $\theta_{r-1}^{1:N}$ with probabilities $w_{r-1}^{1:N}$.
      Sample $\theta_r^i \sim \mathcal{N}(\theta^*, \Sigma_r)$ and simulate $s^i \sim p_{\theta_r^i}(s)$.
      if $\|s^i - s^{\mathrm{o}}\| < \epsilon_r$ **then**
        Accept θ_r^i and compute $w_r^i = \pi(\theta_r^i) / \sum_{j=1}^N w_{r-1}^j \phi_p(\theta_r^i \mid \theta_{r-1}^j, \Sigma_r)$.
      end if
    end while
  end for
  Normalize $w_r^{1:N}$.
end for
Output: Weighted sample $(\theta_R^{1:N}, w_R^{1:N})$ of the ABC posterior density.

An alternative approach to improve upon the standard ABC-SMC algorithm is to revisit the simulation step itself. In this thesis, we propose an improvement to the standard ABC-SMC algorithm that is specifically tailored to SDE models. Our approach is based on a *data-conditional simulator*, in which sample paths are generated conditionally on the observed data y^{o}, rather than unconditionally from the model. This changes the nature of the simulation step: instead of sampling summary statistics from the marginal distribution $p_\theta(s)$, we sample from the conditional distribution $p_\theta(s \mid y^{\mathrm{o}})$. The result is a set of simulated trajectories that more closely follow the observed data, allowing for more informative comparisons. As a consequence, the importance weights in ABC-SMC must also be modified. Since particles are no longer drawn from $p_\theta(s)$, the standard ABC-SMC weight formula no longer applies directly. In *Paper I* we derive the appropriate weight correction that accounts for the data-conditional simulation, and show how this leads to improved efficiency and accuracy in posterior inference for SDE models.

2.1.6 Summary statistics

We have already stated that reducing the dimensionality of the data is crucial in ABC: the approximation error grows as data dimension increases (Prangle, 2018). In practice, well-chosen summary statistics can preserve the information relevant to the parameters while improving ABC's efficiency. Initially, practitioners relied on handcrafted summary statistics based on domain expertise (e.g. mean, variance, autocorrelations), and while expert-chosen features may work well, they provide no guarantee about their adequacy in retaining enough information about θ. Moreover, different models and datasets require different summaries, making manual selection labor-intensive. This motivates a shift towards data-driven or automatic approaches to constructing summary statistics that retain relevant information for the parameters (Blum et al., 2013; Prangle, 2018).

A landmark idea was to construct summaries by regressing the parameter on the data (Fearnhead and Prangle, 2012). The authors proved that the optimal summary statistic (minimizing information loss) for ABC is the posterior mean of the parameters, when considering a quadratic loss. Of course, the posterior mean is not known, but one can estimate it via simulation and regression, which is a key idea in Fearnhead and Prangle (2012). Before the inference task, a simulated set of parameter-data samples is generated from the prior predictive distribution as $(\theta^i, y^i) \sim \pi(\theta)\, p_{\theta^i}(y)$, and then a regression model is fit to predict the parameter value from the simulated data. More precisely, assuming $\dim(\theta) = p$, then for the jth parameter ($j = 1, \ldots, p$), a linear regression model is fit of the form

$$\theta_j^i = \mathbb{E}(\theta_j \mid y^i) + \xi_j^i = b_{0j} + b_j^\top h(y^i) + \xi_j^i, \quad j = 1, \ldots, p,$$

where ξ_j^i is a mean-zero random variable with constant variance, and $h(\cdot)$ is a function of the data, for example it may return powers of y or other transformations. The p different regression models are estimated separately by least squares, and the fitted values $\hat{b}_{0j} + \hat{b}_j^\top h(y)$ provide an estimate for $\mathbb{E}(\theta_j \mid y)$, which can then be used as a summary statistic due to the optimality of the posterior mean. The fitting is performed before the start of ABC and hence the summary statistics function remains fixed throughout. This work was further developed and fully automated by Jiang et al. (2017), who modeled the regression function via a deep neural network:

$$\theta^i = \mathbb{E}(\theta \mid y^i) + \xi^i = f_\beta(y^i) + \xi^i,$$

for the complete multidimensional parameter θ^i, where $f_\beta(\cdot)$ is a neural network parameterized by weights β. Due to the greater representational power of neural networks, the deep learning approach outperforms the linear regression

method, albeit at a higher computational cost. Jiang et al. (2017) found that the neural network summaries led to posteriors as accurate as those using the theoretically optimal or domain-specific summaries. Since then, the use of feed-forward neural nets and other machine learning methods in ABC has become increasingly popular, as discussed in Section 2.1. *Partially exchangeable networks* (PENs; Wiqvist et al., 2019) are a class of invariant neural networks designed to further enhance ABC by learning the posterior mean as a summary statistic. PENs exploit the partially exchangeable structure inherent in Markov chains and are particularly suited for SDE models. Compared to the approach of Jiang et al. (2017), PENs require significantly less training data, as the architecture itself encodes the Markovian structure and therefore does not need to learn it from data. Wiqvist et al. (2019) demonstrate that for models invariant under a subset of the symmetric group, specifically those respecting d-block-switch transformations, the network should also be d-block-switch invariant. To that end, they propose the following regression model for the posterior mean:

$$\theta^i = \mathbb{E}(\theta \mid y^i) + \xi^i = \rho_{\beta_\rho}\left(y^i_{1:d}, \sum_{l=1}^{n-d} \phi_{\beta_\phi}(y^i_{l:l+d})\right) + \xi^i, \tag{23}$$

where ρ_{β_ρ} and ϕ_{β_ϕ} are neural networks, and the sum runs over all overlapping d-blocks of the input sequence. The summary statistics estimator can be trained before running the inference algorithm. However, for this strategy to be effective, very many samples from the prior-predictive distribution are required, and this number is affected by how "vague" the prior is. This is a significant cost, but it is amortized: once the network is trained, producing a summary or performing inference on new data is fast (just a forward pass).

Another approach is to dynamically learn the summary statistics, within a sequential inference algorithm like ABC-SMC, see Chen et al. (2021a). The main idea is to learn the summary statistics and to approximate the posterior density at the same time, over multiple rounds. In ABC-SMC, this entails progressively populating an initial dataset with accepted parameters and their corresponding simulated data. Moreover, the approach of Chen et al. (2021a) involves constructing nearly sufficient summary statistics by maximizing the mutual information between the summary $S(y)$ and the parameter θ. They use deep neural networks to encode the data into $S(y)$ and employ an InfoMax principle (or the principle of maximum information preservation) to ensure that the mutual information $I(\theta, S(y))$ is as high as possible. In particular, Chen et al. (2021a) explore several approaches for learning informative statistics; among them, they propose using distance correlation as a dependency measure, offering a non-parametric alternative to KL-divergence-based mutual information estimators. More concretely, Chen et al. (2021a) propose

optimizing the following objective:

$$\max_{S} \mathcal{L}(S) = \frac{\mathbb{E}_{p(x,\theta)p(x',\theta')}[h(S(x), S(x'))h(\theta, \theta')]}{\sqrt{\mathbb{E}_{\pi(\theta)\pi(\theta')}[h^2(\theta, \theta')] \cdot \mathbb{E}_{p(x)p(x')}[h^2(S(x), S(x'))]}}, \tag{2.8}$$

where $h(a, b) = \|a - b\| - \mathbb{E}_{p(b')}\|a - b'\| - \mathbb{E}_{p(a')}\|a' - b\| + \mathbb{E}_{p(a')p(b')}\|a' - b'\|$. Here, $S(x)$ is parameterized as a neural network and trained to maximize distance correlation between its outputs and the parameters θ, ensuring that the extracted statistics retain information necessary for inference while reducing dimensionality. Through experiments, Chen et al. (2021a) showed that using these learned summaries can boost the performance of not only traditional ABC but also the newer neural likelihood approaches.

2.2 Conclusions

This chapter reviewed simulation-based inference with an focus on ABC and its variants: ABC rejection, ABC importance sampling, ABC–MCMC, ABC–SMC, and synthetic likelihoods. We highlighted the central role of summary statistics and recent approaches for learning them using neural networks. These methods lay the groundwork for the new algorithms developed in the remainder of this thesis, which aim to improve inference for stochastic nonlinear models.

Across the papers, we extend standard ABC samplers to *data-conditional* variants: in *Papers I–II* we develop data-conditional ABC–SMC, while in *Paper III* we propose a data-conditional ABC–MCMC. To mitigate the curse of dimensionality, we employ learned summary statistics: *Papers I–II* use PENs tailored to Markov/time-series structure, and *Paper III* again uses PENs and additionally adopts InfoMax summary statistics.

3 Stochastic differential equations

SDEs are a standard modelling tool for stochastic dynamical systems. They are widely used across numerous scientific areas, including biology, finance, and neuroscience (Fuchs, 2013). In this thesis we apply the parameter inference methods developed across *Papers I-III* to the Chan–Karolyi–Longstaff–Sanders (CKLS Chan et al., 1992) family of financial models and to SDEs arising from chemical reaction networks, among other systems. In general, the likelihood function for SDEs is intractable because the transition densities are characterized as solutions to the Kolmogorov forward/backward equations and they admit no closed-form expression except in special cases (Iacus, 2008; Øksendal, 2013; Fuchs, 2013). ABC algorithms rely on simulating a large amount of sample paths via numerical methods, so the choice of numerical scheme and step size directly trades off accuracy against computational cost (Kloeden and Platen, 1992). In this chapter we briefly recall essentials for SDEs: Brownian motion and Itô calculus, transition densities via Kolmogorov equations, and numerical methods.

3.1 Notation

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. An $\mathcal{X}$-valued *stochastic process* is a family of random variables $(X(t))_{t\in\mathcal{T}}$ defined on $\mathcal{T} \times \Omega$ taking values in $\mathcal{X}$. The random variables are functions of the form

$$X(t, \omega) : \mathcal{T} \times \Omega \to \mathcal{X}.$$

For $\mathcal{T} = \mathbb{N}$, we have a *discrete-time* process, and for $\mathcal{T} \subset \mathbb{R}$ we have a *continuous-time* process. Although the primary focus of this thesis is on continuous-time processes with $\mathcal{T} = [0, \infty)$, *Paper III* specifically addresses the discrete-time case. Each outcome $\omega \in \Omega$ determines a *trajectory* or *sample path* of the

stochastic process X, denoted by $(X(t,\omega))_{t\in\mathcal{T}}$. In the remainder of this section, we will take $\mathcal{T} = [0,\infty)$ and write stochastic processes as $(X(t))_{t\geq 0}$. The progressive accumulation of information generated by a stochastic process is described by a *filtration*, which is a family of sub-σ-algebras $(\mathcal{F}(t))_{t\geq 0}$ such that $\mathcal{F}(s) \subseteq \mathcal{F}(t) \subseteq \mathcal{F}$ for all $0 \leq s \leq t$. Intuitively, $\mathcal{F}(t)$ represents the information available up to time t. A stochastic process $(X(t))_{t\geq 0}$ is adapted to the filtration $(\mathcal{F}(t))_{t\geq 0}$ if, for every $t \geq 0$, the random variable $X(t)$ is $\mathcal{F}(t)$-measurable. A particularly important example of a continuous-time stochastic process is Brownian motion, which plays a central role in both the theory and simulation of stochastic systems. We now turn to its definition and properties.

3.2 Brownian Motion

Brownian motion, also known as the Wiener process, is a cornerstone of stochastic modeling. It serves as the building block for more complex stochastic processes, including stochastic differential equations (SDEs), and underpins many methods in mathematical finance, physics, and biology. Intuitively, Brownian motion models the random and continuous movement of particles suspended in a fluid, as first observed by Robert Brown in the 19th century. Mathematically, it is defined as a continuous-time stochastic process with stationary and independent increments, almost surely continuous paths, and Gaussian increments. We now introduce the standard Brownian motion and summarize its defining properties. A *standard Brownian motion* $(W(t))_{t\geq 0}$ is a real-valued, $\mathcal{F}(t)$-adapted process that satisfies the following properties:

- $W(0) = 0$ almost surely.
- The sample paths $t \mapsto W(t,\omega)$ are almost surely continuous.
- For all $0 \leq s < t$, the increment $W(t) - W(s)$ is independent of $\mathcal{F}(s)$, and its distribution depends only on the length of the interval $t - s$.
- For all $0 \leq s < t$, the increment $W(t) - W(s)$ is normally distributed with mean zero and variance $t - s$, i.e. $W(t) - W(s) \sim \mathcal{N}(0, t-s)$.

From these properties, it follows that Brownian motion can be simulated in a straightforward manner. Given a time increment $h > 0$, one can construct a discrete-time approximation of the process on an interval $[0,T]$ by sampling independent increments from the normal distribution $\mathcal{N}(0,h)$. Specifically, for any $t \geq 0$, $W(t+h) - W(t) \sim \mathcal{N}(0,h)$, so that $W(t+h) = W(t) + \sqrt{h}\cdot Z$ where $Z \sim \mathcal{N}(0,1)$ independently across steps. Repeating this iteratively yields a simulated trajectory of Brownian motion. A multi-dimensional standard Brownian motion is a vector-valued process whose components are independent one-dimensional standard Brownian motions. An example of a simulated trajectory in both one and two dimensions is shown in Figure 3.1. Brownian

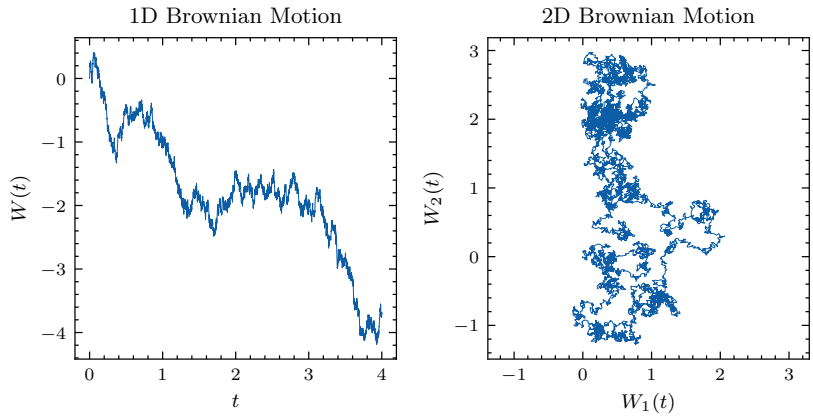

Figure 3.1: Simulated trajectories of Brownian motion. Left: one-dimensional Brownian motion. Right: two-dimensional Brownian motion.

motion is rarely sufficient to model the dynamics of real-world systems on its own. In many applications, we are interested in systems that evolve under the influence of both deterministic trends and stochastic fluctuations. This leads naturally to the study of SDEs, where Brownian motion serves as the source of randomness, alongside a deterministic component.

3.3 Itô Processes and SDEs

In the context of SDEs, a broad class of solutions can be described using Itô processes. These are continuous, adapted processes that can be written as the sum of a deterministic integral term and a stochastic integral with respect to Brownian motion. The definition of an Itô process provides the foundation for much of stochastic calculus and will serve as the basis for the models considered throughout this thesis; a standard reference is Øksendal (2013). We begin by considering scalar Itô processes.

Definition 3.3.1 (Scalar Itô Process)**.** *Let $(\Omega, \mathcal{F}, (\mathcal{F}(t))_{t\geq 0}, \mathbb{P})$ be a filtered probability space, and let $W(t)$ denote a standard Brownian motion adapted to the filtration $(\mathcal{F}(t))_{t\geq 0}$. A* scalar Itô process *is a continuous, $\mathcal{F}(t)$-adapted process $X(t)$ defined on $t \geq 0$ that satisfies the stochastic integral equation*

$$X(t) = X(0) + \int_0^t \mu(X(s))\,\mathrm{d}s + \int_0^t \sigma(X(s))\,\mathrm{d}W(s),$$

where the drift $\mu : \mathcal{X} \to \mathbb{R}$ and diffusion coefficient $\sigma : \mathcal{X} \to \mathbb{R}$ are measurable functions satisfying the integrability conditions

$$\int_0^T |\mu(X(s))| \, \mathrm{d}s < \infty, \quad \int_0^T \sigma^2(X(s)) \, \mathrm{d}s < \infty \quad \textit{almost surely, for all } T < \infty.$$

In differential form, the process is said to satisfy the SDE

$$\mathrm{d}X(t) = \mu(X(t)) \, \mathrm{d}t + \sigma(X(t)) \, \mathrm{d}W(t).$$

The definition of a scalar Itô process extends naturally to multidimensional systems. Given a continuous $\mathcal{F}(t)$-adapted process $X(t) \in \mathcal{X}$ driven by an m-dimensional Brownian motion, we assume the existence of a drift function $\mu : \mathcal{X} \to \mathbb{R}^d$ and a diffusion function $\sigma : \mathcal{X} \to \mathbb{R}^{d \times m}$. Each component μ_i and σ_{ij} is assumed to satisfy the conditions stated in Definition 3.3. Explicitly, each $X_i(t)$ satisfies

$$\mathrm{d}X_i(t) = \mu_i(X(t)) \, \mathrm{d}t + \sum_{j=1}^{m} \sigma_{ij}(X(t)) \, \mathrm{d}W_j(t), \quad \text{for } i = 1, \ldots, d.$$

The Itô integral $\int_0^t \varphi(r) \, \mathrm{d}W(r)$ is defined as a stochastic integral with respect to Brownian motion. While we do not detail its construction here, we recall two important properties in the case where the integrand is deterministic. Let $\varphi : [s, t] \to \mathbb{R}$ be deterministic with $\int_s^t \varphi(r)^2 \, \mathrm{d}r < \infty$. Then

$$\int_s^t \varphi(r) \, \mathrm{d}W(r) \sim \mathcal{N}\left(0, \, \int_s^t \varphi(r)^2 \, \mathrm{d}r\right), \quad 0 \le s < t \le T.$$

Additionally, for any constant $c \in \mathbb{R}$,

$$\int_s^t c \, \mathrm{d}W(r) = c\big(W(t) - W(s)\big).$$

Many applications in stochastic modeling require transforming solutions of SDEs via nonlinear functions. The tool that enables such transformations is Itô's formula, which generalizes the chain rule to stochastic processes. Let $X(t)$ be as defined in Definition 3.3 and let $\psi : \mathcal{X} \to \mathbb{R}$ be a measurable function, twice continuously differentiable. Then the process $Z(t) = \psi(X(t))$ is again a

continuous Itô process and satisfies

$$\begin{aligned} \mathrm{d}Z(t) = &\left(\mu(X(t))\frac{\partial \psi}{\partial x}(X(t)) + \frac{1}{2}\sigma^2(X(t))\frac{\partial^2 \psi}{\partial x^2}(X(t))\right)\mathrm{d}t \\ &+ \sigma(X(t))\frac{\partial \psi}{\partial x}(X(t))\,\mathrm{d}W(t). \end{aligned}$$

for all $t \in [0, T]$. Itô's formula admits an extension to d-dimensional systems; however, in this thesis we will only apply it to scalar processes, typically in a component-wise manner.

The transformation of SDEs into simpler or more analytically tractable forms is a recurring theme in stochastic modeling. In particular, one often seeks to simplify the diffusion coefficient of an SDE—ideally transforming it into a constant. A classical tool for this purpose is the *Lamperti transform*, which maps a one-dimensional diffusion process with non-constant diffusion into one with unit diffusion (Iacus, 2008). The Lamperti transform often enables analytical tractability and plays a crucial role in the construction of certain exact simulation algorithms and variance reduction techniques. We now state a general version of the one-dimensional Lamperti transform, see Møller and Madsen (2010) for the multivariate Lamperti transform.

Theorem 3.3.2 (Lamperti Transform)**.** *Let $X(t)$ be as defined in Definition 3.3 and define*

$$\psi(X(t)) = \int \frac{1}{\sigma(x)}\,\mathrm{d}x\bigg|_{x=X(t)}.$$

If $\psi(\cdot) : \mathcal{X} \to \mathbb{R}$ is one-to-one for each $t \in [0, \infty)$, then we define $Z(t) = \psi(X(t))$. Otherwise, if $\sigma(X(t)) > 0$ for all t and all $X(t)$, define

$$Z(t) = \psi(X(t)) = \int_y^x \frac{1}{\sigma(u)}\,\mathrm{d}u\bigg|_{x=X(t)},$$

where y is an arbitrary point in $\mathcal{X}$. Then $Z(t)$ satisfies an SDE with unit diffusion:

$$\mathrm{d}Z(t) = \left(\frac{\mu(\psi^{-1}(Z(t)))}{\sigma(\psi^{-1}(Z(t)))} - \frac{1}{2}\frac{\partial \sigma}{\partial x}(\psi^{-1}(Z(t)))\right)\mathrm{d}t + \mathrm{d}W(t).$$

This transformation reduces the complexity of the diffusion term and can greatly simplify the analysis of the process. The transformed process $Z(t)$ evolves with constant volatility, and if the inverse ψ^{-1} can be computed in closed form, the original process $X(t)$ can be explicitly reconstructed from $Z(t)$. The geometric Brownian motion is a classical example of a diffusion

with multiplicative noise, and applying the Lamperti transform simplifies the analysis by converting the model into one with constant diffusion.

Example 3.3.3 (Geometric Brownian Motion)**.** *Consider the SDE*

$$\mathrm{d}X(t) = aX(t)\,\mathrm{d}t + \sigma X(t)\,\mathrm{d}W(t), \quad X(0) = 1,$$

where $a, \sigma \in \mathbb{R}$ are constants. The Lamperti transform $Z(t) = \log(X(t))/\sigma$ reduces the SDE to one with constant diffusion:

$$\mathrm{d}Z(t) = \left(\frac{a}{\sigma} - \frac{1}{2}\sigma\right)\mathrm{d}t + \mathrm{d}W(t).$$

The resulting linear SDE can be solved explicitly, yielding a process $Z(t)$ with a Gaussian distribution whose mean and variance are known in closed form, and since $X(t) = e^{\sigma Z(t)}$, it follows that $X(t)$ is log-normally distributed.

3.4 Solutions to SDEs

Having introduced the class of Itô processes and their role in modeling stochastic dynamics, we now turn to the question of what it means to solve an SDE. Unlike deterministic differential equations, SDEs admit multiple notions of solution depending on how randomness is handled. In particular, we distinguish between *strong* and *weak* solutions, which differ in whether the Brownian motion driving the system is fixed in advance or constructed alongside the solution. We also discuss the standard assumptions under which solutions exist and are unique, such as Lipschitz continuity and linear growth conditions on the drift and diffusion coefficients (see Kloeden and Platen, 1992). These notions are essential for both the theoretical analysis of SDEs and the design of numerical methods for their approximation.

A solution to a SDE may be understood in several senses, depending on how the underlying probability space and driving Brownian motion are treated. A *strong solution* is a process $X(t)$ that is adapted to a given filtration and constructed on a fixed probability space, with respect to a fixed Brownian motion $W(t)$. It satisfies the SDE, almost surely for all $t \geq 0$. In contrast, a *weak solution* consists of a new probability space, filtration, Brownian motion, and adapted process $(Y(t), W(t))$ such that the SDE is satisfied in distribution, but without requiring that the solution be constructed from a given Brownian motion. Existence and uniqueness of strong solutions typically rely on structural assumptions on the drift and diffusion functions. If μ and σ satisfy a Lipschitz

condition—that is, there exists a constant $C > 0$ such that for all $x, y \in \mathcal{X}$,

$$\|\mu(x) - \mu(y)\| + \|\sigma(x) - \sigma(y)\| \leq C\|x - y\|,$$

then there exists a unique strong solution. Moreover, if the initial condition satisfies $\mathbb{E}\|X(0)\|^2 < \infty$ and there exists a constant $D > 0$ such that

$$\|\mu(x)\|^2 + \|\sigma(x)\|^2 \leq D(1 + \|x\|^2)$$

for all $x \in \mathcal{X}$, then the solution is non-explosive and has finite second moments.

The concept of *pathwise uniqueness* plays a key role in the theory of strong solutions. It states that if two processes $X_1(t)$ and $X_2(t)$ are both solutions to the same SDE with the same Brownian motion and initial condition, then they must coincide almost surely:

$$\mathbb{P}\left(\sup_{t \in \mathcal{T}} \|X_1(t) - X_2(t)\| > 0\right) = 0.$$

That is, $X_1(t) = X_2(t)$ for all $t \geq 0$ with probability one. Pathwise uniqueness, together with the existence of a weak solution, implies the existence of a strong solution under general conditions. Finally, we distinguish between pathwise uniqueness and *uniqueness in law*. The latter refers to the property that all solutions to the SDE—regardless of the underlying probability space—have the same distribution. While pathwise uniqueness implies uniqueness in law, the converse does not hold in general.

3.4.1 Strong Approximations

Exact solutions to SDEs are available only in a limited number of special cases. In practice, one often seeks to construct numerical approximations that converge to the true solution of the SDE in a suitable sense. In this section, we consider *strong approximations*, which aim to approximate individual sample paths of the solution process. This is in contrast to weak approximations, which target the convergence of distributions or expectations of functionals.

Let $W(t) = (W_1(t), \ldots, W_m(t))^\top$ be an m-dimensional vector of independent standard Brownian motions. Consider the d-dimensional Itô process satisfying the SDE

$$\mathrm{d}X(t) = \mu(X(t))\,\mathrm{d}t + \sigma(X(t))\,\mathrm{d}W(t), \quad t \in [0, T], \tag{3.1}$$

with initial condition $X(0) = X_0$, $X(t) \in \mathcal{X} \subset \mathbb{R}^d$, the drift function $\mu : \mathcal{X} \to \mathbb{R}^d$, and the diffusion function $\sigma : \mathcal{X} \to \mathbb{R}^{d \times m}$ satisfy suitable integrability

conditions. The corresponding integral form of the equation is

$$X(t) = X_0 + \int_0^t \mu(X(s))\,\mathrm{d}s + \sum_{j=1}^{m} \int_0^t \sigma_j(X(s))\,\mathrm{d}W_j(s), \quad t \in [0,T], \tag{3.2}$$

where σ_j denotes the jth column of the matrix-valued function σ.

Since explicit solutions to (3.1) are rarely available, we now turn to numerical methods for approximating the sample paths of $X(t)$. The idea is to discretize the interval $[0,T]$ using a step size $h > 0$, and to approximate the stochastic integrals over each subinterval using simple quadrature rules. In particular, we now derive the Euler–Maruyama method by applying the left-point (rectangular) rule to both the deterministic and stochastic integrals appearing in the SDE. Let $X(\tau_k)$ be the value of the process at time $\tau_k = kh$, and consider the integral form of the SDE over a single time step on the interval $[\tau_k, \tau_{k+1}]$, where $\tau_{k+1} = (k+1)h$ and $k = 0, \ldots, N-1$. To construct a numerical approximation, we apply the Euler (rectangular) rule to the stochastic integral. Specifically, each integral term

$$\int_{\tau_k}^{\tau_{k+1}} \sigma_j(X(s))\,\mathrm{d}W_j(s)$$

is approximated by evaluating the integrand at the left endpoint:

$$\begin{aligned}\int_{\tau_k}^{\tau_{k+1}} \sigma_j(X(s))\,\mathrm{d}W_j(s) &\approx \sigma_j(X(\tau_k)) \int_{\tau_k}^{\tau_{k+1}} \mathrm{d}W_j(s) \\ &= \sigma_j(X(\tau_k))\big(W_j(\tau_{k+1}) - W_j(\tau_k)\big).\end{aligned}$$

We denote the Brownian increment over this step by $\Delta W_{j,k+1} := W_j(\tau_{k+1}) - W_j(\tau_k)$. This approximation yields the *Euler–Maruyama scheme*, a method for simulating sample paths of SDEs. We denote the numerical approximation of the solution by X^{EuM}, where $X_{i,k}^{\mathrm{EuM}}$ represents the ith component of the approximation at time $\tau_k = kh$. The update rule is given by

$$X_{i,k+1}^{\mathrm{EuM}} = X_{i,k}^{\mathrm{EuM}} + \mu_i(X_{1:d,k}^{\mathrm{EuM}})\,h + \sum_{j=1}^{m} \sigma_{ij}(X_{1:d,k}^{\mathrm{EuM}})\,\Delta W_{j,k+1},$$

for $i = 1, \ldots, d$, where $X_{1:d,k}^{\mathrm{EuM}} = (X_{1,k}^{\mathrm{EuM}}, \ldots, X_{d,k}^{\mathrm{EuM}})^\top$ denotes the full vector of approximated components at time τ_k.

3.4.2 Limitations of higher-order methods for non-commutative noise

While the Euler–Maruyama scheme provides a simple and widely used method for numerically solving SDEs, it is only of strong order $\mathcal{O}(\sqrt{h})$. To improve accuracy, one can attempt to derive higher-order schemes using an Itô–Taylor expansion. The Milstein scheme is a classical example of such a method, and in the scalar case it offers a strong order of convergence $\mathcal{O}(h)$ without additional complications. To make this improvement concrete, we now consider one additional term in the Itô–Taylor expansion, which yields the Milstein scheme. For $k \in \{0, ..., N-1\}$, the Milstein approximation $X^{\mathrm{M}}_{1:d,k+1}$ given $X^{\mathrm{M}}_{1:d,k}$ is defined by

$$X^{\mathrm{M}}_{1:d,k+1} := X^{\mathrm{M}}_{1:d,k} + \mu(X^{\mathrm{M}}_{1:d,k})h + \sum_{i=1}^{m} \sigma_i(X^{\mathrm{M}}_{1:d,k}) I^h_i + \sum_{i,j=1}^{m} \sigma'_i(X^{\mathrm{M}}_{1:d,k})\sigma_j(X^{\mathrm{M}}_{1:d,k}) I^h_{j,i},$$

where $\sigma_i(x)$ again denotes the ith column of $\sigma(x) \in \mathbb{R}^{d\times m}$, and

$$\sigma'_i(x) := \begin{pmatrix} \frac{\partial}{\partial x_1}\sigma_{i,1}(x) & \cdots & \frac{\partial}{\partial x_d}\sigma_{i,1}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_1}\sigma_{i,d}(x) & \cdots & \frac{\partial}{\partial x_d}\sigma_{i,d}(x) \end{pmatrix}$$

is the Jacobian of $\sigma_i(x)$. The stochastic integrals appearing in the scheme are defined as

$$I^h_i := \int_{\tau_k}^{\tau_{k+1}} \mathrm{d}W_i(r), \qquad I^h_{j,i} := \int_{\tau_k}^{\tau_{k+1}} \left(\int_{\tau_k}^{r} \mathrm{d}W_j(p) \right) \mathrm{d}W_i(r).$$

Using the identities $I^h_{i,i} = \left((I^h_i)^2 - h\right)/2$, and $I^h_{i,j} + I^h_{j,i} = I^h_i I^h_j$, the last Milstein correction term can be written as

$$\begin{aligned} \sum_{i,j=1}^{m} \sigma'_i(X^{\mathrm{M}}_{1:d,k})\sigma_j(X^{\mathrm{M}}_{1:d,k}) I^h_{j,i} &= \frac{1}{2}\sum_{i=1}^{m} \sigma'_i(X^{\mathrm{M}}_{1:d,k})\sigma_i(X^{\mathrm{M}}_{1:d,k}) \left((I^h_i)^2 - h\right) \\ &+ \frac{1}{2}\sum_{\substack{i,j=1 \\ i<j}}^{m} \left(\sigma'_i(X^{\mathrm{M}}_{1:d,k})\sigma_j(X^{\mathrm{M}}_{1:d,k}) + \sigma'_j(X^{\mathrm{M}}_{1:d,k})\sigma_i(X^{\mathrm{M}}_{1:d,k})\right) I^h_i I^h_j \\ &+ \sum_{\substack{i,j=1 \\ i<j}}^{m} \left(\sigma'_i(X^{\mathrm{M}}_{1:d,k})\sigma_j(X^{\mathrm{M}}_{1:d,k}) - \sigma'_j(X^{\mathrm{M}}_{1:d,k})\sigma_i(X^{\mathrm{M}}_{1:d,k})\right) A^h_{ij}, \end{aligned}$$

where

$$A_{ij}^h := \frac{1}{2}\left(I_{i,j}^h - I_{j,i}^h\right)$$

are the Lévy area terms, which are computationally expensive to simulate; see Rydén and Wiktorsson (2001); Wiktorsson (2001); Malham and Wiese (2014) for various approaches. However, if the commutativity condition

$$\sigma_i'(x)\sigma_j(x) = \sigma_j'(x)\sigma_i(x), \quad \text{for all } i,j \in \{1,\dots,m\},$$

is satisfied, then the Lévy area terms vanish. This condition always holds in the scalar case with one driving noise, and more generally when the noise is commutative.

In summary, while the Milstein scheme offers higher-order accuracy in theory, its application to multidimensional SDEs with non-commutative noise involves nontrivial terms that are difficult to simulate efficiently. As a result, the Euler–Maruyama scheme remains the most commonly used method in high-dimensional settings despite its lower order of convergence.

3.5 Transition densities and the Kolmogorov forward equation

Given the Itô process $X(t)$, the *transition density* is a function $p(t, y \mid s, x)$ such that for any Borel set $A \subseteq \mathcal{X}$ and times $0 \le s < t$, the conditional probability of the process moving from state x at time s to the set A at time t is given by

$$\mathbb{P}\big(X(t) \in A \mid X(s) = x\big) = \int_A p(t, y \mid s, x)\,\mathrm{d}y.$$

When it exists, the transition density characterizes the distribution of $X(t)$ conditioned on $X(s) = x$. The transition density plays a central role in both the theoretical and applied study of SDEs. It provides a complete probabilistic description of the evolution of the system and underlies quantities of interest such as marginal distributions, first-passage times, and expectations of functionals. In statistical inference, the transition density is often the key object used to evaluate the likelihood of observed data.

To ensure the existence and regularity of the transition density, we impose standard conditions on the drift and diffusion coefficients. Specifically, we assume that (i) the drift μ is continuously differentiable and each σ_{ij} is twice continuously differentiable; (ii) the first and second derivatives of μ and σ are locally bounded; and (iii) the diffusion matrix $\sigma(x)\sigma(x)^\top$ is uniformly elliptic, i.e. $\xi^\top\sigma(x)\sigma(x)^\top\xi \ge \lambda\|\xi\|^2$, for all $x \in \mathcal{X}, \xi \in \mathbb{R}^d$, for some constant $\lambda > 0$.

Under these conditions, the transition density $p(t, y \mid s, x)$ exists, is smooth in y, and satisfies the *Kolmogorov forward equation* (also known as the *Fokker–Planck equation*):

$$\frac{\partial}{\partial t}\, p(t, y \mid s, x) = -\sum_{i=1}^{d} \frac{\partial}{\partial y_i}\Big(\mu_i(y)\, p(t, y \mid s, x)\Big) + \frac{1}{2}\sum_{i,j=1}^{d} \frac{\partial^2}{\partial y_i\, \partial y_j}\Big((\sigma(y)\sigma(y)^\top)_{ij}\, p(t, y \mid s, x)\Big),$$

with $p(s, y \mid s, x) = \delta_x(y)$.

In general, closed-form expressions for $p(t, y \mid s, x)$ are only available in a limited number of special cases, such as the Ornstein–Uhlenbeck process, geometric Brownian motion or the Cox–Ingersoll–Ross model. For most nonlinear or high-dimensional systems, solving the Kolmogorov PDE analytically is infeasible. Even numerical solutions are often impractical due to the curse of dimensionality. As a result, simulation-based methods play a central role in approximating the transition density, particularly in the context of parameter inference.

3.6 The Cox–Ingersoll–Ross process

The Cox–Ingersoll–Ross (CIR) process, originally introduced by Cox et al. (1985) for modeling short-term interest rates, is a one-dimensional SDE widely used to model positive-valued quantities such as interest rates, volatility, and population sizes. It is defined by

$$\mathrm{d}X(t) = b(a - X(t))\,\mathrm{d}t + \sigma\sqrt{X(t)}\,\mathrm{d}W(t), \tag{3.3}$$

where $a, b, \sigma > 0$ are constants and $W(t)$ is a standard Brownian motion. The CIR process has two important features: mean reversion towards a at a rate b, and state-dependent volatility proportional to the square root of the current value $X(t)$. Due to the square-root diffusion, the process is non-negative for all t. In particular, if the Feller condition $2ab \geq \sigma^2$ holds, the boundary at zero is unattainable and the process stays strictly positive almost surely. Even if an explicit solution to the CIR (3.3) is not available in closed form, we can simulate from it exactly, as its conditional transition density is known to follow a non-central chi-square distribution. Specifically, the density of transitioning from state x at time s to state y at time $t > s$ is given by

$$p(t, y \mid s, x) = ce^{-u-v}\left(\frac{u}{v}\right)^{q/2} I_q\left(2\sqrt{uv}\right), \qquad x, y \in \mathbb{R}_+,$$

where

$$c = \frac{2b}{\sigma^2(1 - e^{-b(t-s)})}, \qquad q = \frac{2ab}{\sigma^2} - 1, \qquad u = cxe^{-b(t-s)}, \qquad v = cy.$$

Here, $I_q(\cdot)$ denotes the modified Bessel function of the first kind of order q.

Nevertheless, several numerical schemes have been proposed in the literature to approximate its strong solution, as it is often considered an important model to evaluate new numerical methods, due to the numerical challenges it poses. Standard schemes (such as EuM) fail to preserve the positivity of its trajectories, and the typical assumptions required for weak and strong convergence results (Kloeden and Platen, 1992) do not hold as the diffusion coefficient is not Lipschitz continuous. This explains why EuM does not converge, despite being extensively used in practice, often with added modifications to ensure non-negativity, such as truncating at zero or taking absolute values. Modified versions of EuM or of the Milstein scheme have been proposed to tackle this, e.g. tamed EuM, truncated EuM or truncated Milstein (Higham and Mao, 2005; Lord et al., 2010; Cozma and Reisinger, 2020; Hefter and Herzwurm, 2018). In *Paper II* we focus on a different approach consisting of: i) transforming the SDE with multiplicative noise (3.3) into one with additive noise by applying the Lamperti transformation $Z(t) = \sqrt{X(t)}$ to (3.3), ii) discretizing the resulting SDE, e.g. with an implicit or splitting scheme (Alfonsi, 2005, 2013; Dereich et al., 2012; Chassagneux et al., 2016; Kelly and Lord, 2023; Kelly et al., 2022), iii) invert the Lamperti transform, mapping the numerical solution of the Lamperti transformed SDE to the original SDE, i.e. $X(t) = Z(t)^2$. More precisely, the Lamperti transformed SDE takes the form

$$\mathrm{d}Z(t) = \left(-\frac{bZ(t)}{2} + \frac{4ab - \sigma^2}{8Z(t)}\right)\mathrm{d}t + \frac{\sigma}{2}\,\mathrm{d}W(t).$$

The square root in the diffusion coefficient is now removed, mitigating the issue of negative values for Taylor-based schemes (such as EuM). However, care must still be taken to prevent solutions from reaching zero due to the nonlinear term $1/Z(t)$ appearing in the drift of the Lamperti transformed SDE. We refer to Kelly and Lord (2023) and to Section 8.5.2 of Kelly (2024) for a comprehensive review on these simulation techniques.

3.7 Conclusions

This chapter has introduced the key tools from stochastic calculus that will be used throughout the thesis, with particular emphasis on strong approximations to SDEs, the Lamperti transform, the CIR process, and transition densities. All

of which play a key role in the inference and simulation tasks that follow. The CIR process, for instance, will form the basis for a class of processes we term *perturbed-conditionally-CIR-type*, while the Lamperti transform will be used to simplify their diffusion structure. The Kolmogorov forward equation will serve as a valuable tool in the approximation of Markov jump processes by a diffusion, aiming to replace the discrete dynamics with a continuous-state process (commonly referred to as a diffusion approximation) for systems with large molecular populations and frequent reaction events. This is especially in the context of chemical reaction networks. We will also investigate numerical schemes for the Chemical Langevin Equation (CLE), a diffusion approximation to such jump processes. As the CLE involves non-commutative noise, it serves as a natural setting where standard higher-order schemes require special attention.

4 Chemical reaction networks

Chemical reaction networks (CRNs) provide a powerful framework for modeling the interactions among chemical species that drive the behavior of biochemical systems. At the core of many cellular processes such as gene regulation and signal transduction, lie sequences of molecular interactions that can be described as reactions between chemical species (Wilkinson, 2018; Lei, 2021; Schnoerr et al., 2017). While the underlying biology is complex, CRNs offer a simplified representation of these processes by abstracting away molecular details. Interactions are encoded through the stoichiometry of each reaction, specifying how species are produced and consumed. These interactions are influenced by both intrinsic noise, arising from the probabilistic nature of molecular reactions, and extrinsic noise, stemming from environmental fluctuations (Arkin et al., 1998; McAdams and Arkin, 1997; Feinberg, 2014; Gupta et al., 2011; Paulsson et al., 2000; Tian and Burrage, 2006). In cellular environments, especially when the copy numbers of certain molecules are low, stochastic fluctuations can lead to significant variability in system behavior. This motivates a probabilistic treatment of the dynamics, where the system evolves as a stochastic process governed by propensity functions that quantify the likelihood of each reaction occurring over time.

With this in mind, CRNs serve as the starting point for a host of mathematical models. At the most fundamental level, they define a continuous-time Markov jump process (MJP) whose probability distribution satisfies the so-called *chemical master equation* (CME). For large networks with many species and reactions, or when the state space is high-dimensional (e.g., many possible molecule counts), exact simulation or inference becomes computationally infeasible. In such cases, tractable approximations are available, including deterministic rate equations and SDEs such as the *chemical Langevin equation* (CLE). Regardless of the chosen modeling scale, the reaction network remains the central object: it defines the species, their interactions, and the rules that drive the system's evolution.

Time-course data, such as measurements of protein concentrations or gene

expression levels, offer a window into these processes. Such data are typically obtained through fluorescence microscopy or other high-resolution experimental techniques, often at discrete time points and subject to measurement error (Young et al., 2012; Bar-Joseph et al., 2012; Locke and Elowitz, 2009). Additionally, not all components of a system are directly observed, and this partial observability, combined with measurement noise, poses significant challenges for inferring relevant biological information (Golightly and Wilkinson, 2008; Komorowski et al., 2009; Bronstein et al., 2015; Stathopoulos and Girolami, 2013).

To make these concepts precise, we now introduce the mathematical formalism of CRNs, which serve as the foundation for the stochastic models and approximations discussed throughout this chapter.

4.1 Notation

A CRN consists of a set of d chemical species, $X_1, \ldots, X_d$, $X_i \in \mathbb{N}$, that interact via a network of r reactions

$$\begin{aligned}
\nu^-_{1,1}X_1 + \nu^-_{2,1}X_2 + \ldots + \nu^-_{d,1}X_d &\xrightarrow{k_1} \nu^+_{1,1}X_1 + \nu^+_{2,1}X_2 + \ldots + \nu^+_{d,1}X_d \\
\nu^-_{1,2}X_1 + \nu^-_{2,2}X_2 + \ldots + \nu^-_{d,2}X_d &\xrightarrow{k_2} \nu^+_{1,2}X_1 + \nu^+_{2,2}X_2 + \ldots + \nu^+_{d,2}X_d \\
&\vdots \\
\nu^-_{1,r}X_1 + \nu^-_{2,r}X_2 + \ldots + \nu^-_{d,r}X_d &\xrightarrow{k_r} \nu^+_{1,r}X_1 + \nu^+_{2,r}X_2 + \ldots + \nu^+_{d,r}X_d
\end{aligned}$$

where the stoichiometric coefficients $\nu^-_{i,j}$ and $\nu^+_{i,j}, i = 1, \ldots, d, j = 1, \ldots, r$ are the non-negative integer numbers of reactant and product molecules, respectively, and $k_i > 0$ are the kinetic rate parameters. Commonly modeled by deterministic rate equations under the law of mass action, which provide a good approximation when molecule numbers are high, CRNs may instead require stochastic models when low molecule counts make fluctuations significant (Elowitz et al., 2002).

Under well-mixed conditions and assuming thermal equilibrium, the dynamics of a chemical reaction system in a closed compartment of volume Ω depend only on molecule counts (Schnoerr et al., 2017; Fuchs, 2013). With these assumptions, Gillespie (1992) derived the CME, whose solution gives the transition probability governing the continuous-time, discrete-valued Markov process, known as a MJP. For simplicity, we will use X_j to denote the count of species j, avoiding the need for a separate variable. The dynamics of this system are described by the process $X(t) = (X_1(t), \ldots, X_d(t))$, where each component represents the count of a given species over time. Since chemical reactions occur

at random times, governed by the probabilistic rules encoded in the propensity functions, $X(t)$ is a continuous-time stochastic process. It can be shown that the probability of the j-th reaction occurring in an infinitesimal time step $\mathrm{d}t$ is $a_j(x)\,\mathrm{d}t$, where $a_j(x)$ is the reaction's propensity function, proportional to the combinations of reactant molecules in $X(t) = x$. The general form of the propensity functions take the form

$$a_j(x) = k_j \prod_{i=1}^{d} \binom{x_i}{\nu_{i,j}^-}. \tag{4.1}$$

Propensity functions of this form are called mass-action kinetics type (Van Kampen, 1992). The specific shape of the propensity function depends on the number of reactant molecules involved in the reaction – a concept referred to as the reaction *order*. This notion arises naturally from combinatorial considerations and determines the functional form of each propensity, as seen in Equation (4.1). Below, we distinguish between reactions of zeroth, first, second, and higher orders, and explain how each case yields a specific functional form.

4.2 Order of reactions and non-mass-action functions

Zeroth-order reactions represent events that occur independently of the current state of the system, for example a constant influx or spontaneous appearance of molecules due to an external source. A reaction of the form

$$\mathcal{R}_j : \varnothing \xrightarrow{k_j} X_i,$$

has a constant propensity $a_j(x) = k_j$, which reflects the fact that the reaction rate is independent of the population of any species. While mass cannot be created from nothing, such reactions serve as useful idealizations. *First-order reactions* involve a single reactant molecule and occur with a rate proportional to the number of available molecules of that species. If x_i molecules of species X_i are present, and each can undergo a reaction independently with rate k_j, then the total propensity is $a_j(x) = k_j x_i$. These reactions have the form

$$\mathcal{R}_j : X_i \xrightarrow{k_j} \dots$$

Such reactions model processes like the change of a molecule into other molecules (such as radioactive decay), or the spontaneous dissociation of a complex molecule into simpler molecules. *Second-order reactions* describe interactions between two molecules. When two distinct species X_j and X_k react, and there

are x_j and x_k molecules of each, the number of possible interacting pairs is $x_j x_k$, leading to the propensity $a_j(x) = k_j x_i x_k$. These reactions have the form

$$\mathcal{R}_j : X_i + X_k \xrightarrow{k_j} \dots$$

In the case where both reactants are of the same species (e.g., $2X_i \to \dots$), the number of unique unordered pairs is $\binom{x_i}{2} = x_i(x_i - 1)/2$, so the corresponding propensity becomes $a_j(x) = k_j x_i(x_i - 1)/2$. While the derivation of zeroth-, first-, and second-order reaction propensities follows directly from simple counting arguments, *higher-order reactions* (involving three or more reactant molecules) are rarely modeled explicitly. The probability that three molecules simultaneously collide is generally negligible in dilute systems. Nonetheless, higher-order reactions can be formally described by generalizing the same combinatorial reasoning. For example, in a trimerization reaction

$$\mathcal{R}_j \colon 3X_i \xrightarrow{k_j} \dots,$$

the number of combinations of three molecules from a population of x is $\binom{x_i}{3} = \frac{x_i(x_i-1)(x_i-2)}{6}$, yielding the propensity

$$a_j(x) = k_j \frac{x_i(x_i - 1)(x_i - 2)}{6}.$$

In practice, however, such high-order interactions are more realistically modeled by decomposing the reaction into a sequence of lower-order reactions. For instance, the trimerization above could be represented by two consecutive second-order reactions:

$$2X \to X_2, \quad X_2 + X \to X_3.$$

Such factorizations better reflect the underlying biophysical reality and allow for simpler analytical treatment and more stable numerical simulation.

Overall, the order of a reaction encapsulates the number of simultaneously interacting molecules and directly informs the shape of the associated propensity function. These forms, when inserted into the CME or its approximations such as the CLE, drive the time evolution of the system and are central to both theoretical analysis and practical simulation. Other types of propensity functions are also useful for various processes, such as Michaelis-Menten or Hill functions. These non-mass-action functions typically act as effective reactions, replacing several underlying microscopic reactions. The Michaelis-Menten model is one of the best known models of enzyme kinetics (Michaelis and Menten, 1913). It arises in enzyme-catalyzed reactions where a substrate, S,

binds reversibly to an enzyme, E, to form a complex, ES, which then converts into a product P. Schematically, this process can be represented as

$$S + E \underset{k_1^-}{\overset{k_1^+}{\rightleftharpoons}} ES \xrightarrow{k_2} EP \xrightarrow{k_3} P + E.$$

The dynamics of the molecular concentrations can be described by a system of ordinary differential equations. Under the quasi-steady-state assumption, the production rate of P from the substrate S is approximately given by

$$v = k_3[\mathrm{EP}] = \frac{C[S]}{K + [S]} E_{\text{total}}, \quad \text{where } C = \frac{k_2 k_3}{k_2 + k_3}, \quad K = \frac{k_1^- + k_2}{k_1^+ \left(1 + \frac{k_2}{k_3}\right)},$$

and E_{total} is the total enzyme concentration. In systems where an enzyme can bind to multiple substrate molecules simultaneously, the Hill function provides a useful model for the production rate. This leads to the generalized reaction

$$nS + E \underset{k_1^-}{\overset{k_1^+}{\rightleftharpoons}} ES_n \xrightarrow{k_2} EP_n \xrightarrow{k_3} nP + E.$$

Assuming quasi-steady-state conditions and rapid equilibration, the production rate of P is given by the Hill equation:

$$v = \frac{v_{\max}[S]^n}{K^n + [S]^n},$$

where $v_{\max}$ is the maximum reaction rate, K is the substrate concentration at which the rate is half-maximal, and n is the Hill coefficient, indicating the degree of cooperativity in substrate binding. When $n > 1$, the system exhibits a sigmoidal response, which is typical in cooperative enzyme binding. This function appears in the Repressilator model (Elowitz and Leibler, 2000), which is considered in this thesis. The Repressilator provides an interesting test case for illustrating our numerical method and performing parameter inference, including the estimation of the Hill coefficient.

4.3 Stochastic dynamics and diffusion approximation

CRNs are commonly described as continuous-time MJPs, where the system evolves through a sequence of discrete jumps corresponding to individual reactions. The probability distribution over discrete molecular states satisfies the

CME (Van Kampen, 1992), which provides a complete probabilistic description of the system. However, this description becomes computationally infeasible for larger systems, as the state space grows combinatorially with the number of species. Let $p(t, y \mid s, x)$ denote the transition probability mass function representing the probability of the system being in a particular state at time t, given its state at an earlier time s. The transition probability is given by the solution to the CME

$$\frac{\partial}{\partial t}\, p(t, y \mid s, x) = \sum_{j=1}^{r} \big(a_j(y - \nu_j)\, p(t, y - \nu_j \mid s, x) - a_j(y)\, p(t, y \mid s, x)\big), \quad (4.2)$$

where $\nu_j = (\nu_{1,j}, \ldots, \nu_{d,j})^T$ is the jth column of the stoichiometry matrix ν, with elements $\nu_{i,j} = \nu^+_{i,j} - \nu^-_{i,j}$ (Wilkinson, 2018). Analytical solutions to the CME are only known for a limited class of systems and a few special cases; see, for example Jahnke and Huisinga (2007); Shahrezaei and Swain (2008); Zhou et al. (2012). Although exact sample paths can be simulated using Gillespie's stochastic simulation algorithm (Gillespie, 1977), this approach becomes prohibitively expensive for larger systems due to the need to simulate every individual reaction event (Gillespie et al., 2013). The computational demands of inference methods that rely on this description pose a significant challenge for large-scale reaction networks. Consequently, extensive research has focused on approximation methods (Schnoerr et al., 2017). To address this, a diffusion approximation, aiming to replace the discrete dynamics with continuous-state approximation for systems with large molecular populations and frequent reaction events, has been proposed (Wilkinson, 2018; Golightly and Wilkinson, 2011). The resulting SDE, known as the CLE, captures the intrinsic stochasticity of the biochemical system while significantly reducing its computational complexity. However, the absence of closed-form solutions for the SDE transition densities presents challenges for parameter inference, necessitating simulation-based methods for statistical estimation (Wilkinson, 2018; Fuchs, 2013), or further approximations to give a tractable process (Fearnhead et al., 2014; Stathopoulos and Girolami, 2013; Finkenstädt et al., 2013; Komorowski et al., 2009). In what follows, we briefly outline the derivation of the chemical Fokker–Planck equation via the Kramers–Moyal expansion, and establish its connection to the CLE.

4.3.1 Diffusion approximation

The CLE and the associated Fokker–Planck equation provide a diffusion approximation to the CME, in the sense that the CLE approximates the underlying Markov jump process with a continuous-state stochastic process governed by an SDE. The CLE can be derived in a number of more or less formal ways.

For instance, truncating terms in a Taylor expansion of the CME yields the Fokker–Planck equation, the PDE describing the evolution of the probability density of a diffusion process (and thus a continuous-time, continuous-valued Markov process rather than a discrete-valued one, see Kramers, 1940; Moyal, 1949), governed by an underlying SDE. More precisely, using Taylor's expansion one can write

$$a_j(y-\nu_j)\,p(t,y-\nu_j \mid s,x) = a_j(y)\,p(t,y \mid s,x) + \sum_{|v|\geq 1} \frac{(-1)^{|v|}}{v_1!\cdots v_d!}\left(\prod_{i=1}^{d} \nu_{ij}^{v_i}\right) \frac{\partial^{|v|}}{\partial y_1^{v_1}\cdots \partial y_d^{v_d}}\Big(a_j(y)\,p(t,y \mid s,x)\Big), \tag{4.3}$$

where $v = (v_1, ..., v_d)$ is the multi-index ranging over non-negative integers with total degree $|v| = \sum_i v_i$. By substituting (4.3) into the CME (4.2) one obtains the chemical Kramers-Moyal equation

$$\frac{\partial}{\partial t}\,p(t,y \mid s,x) = \sum_{|v|\geq 1} \frac{(-1)^{|v|}}{v_1!\cdots v_d!}\,\frac{\partial^{|v|}}{\partial y_1^{v_1}\cdots \partial y_d^{v_d}}\Big(A^v(y)\,p(t,y \mid s,x)\Big),$$

where

$$A^v(y) = \sum_{j=1}^{r}\left(\prod_{i=1}^{d} \nu_{ij}^{v_i}\right) a_j(y).$$

Truncating this series at second order leads to the *chemical Fokker–Planck equation*:

$$\frac{\partial}{\partial t}\,p(t,y \mid s,x) = -\sum_{i=1}^{d}\frac{\partial}{\partial y_i}\Big(A_i(y)\,p(t,y \mid s,x)\Big) + \frac{1}{2}\sum_{i,j=1}^{d}\frac{\partial^2}{\partial y_i\,\partial y_j}\Big(B_{ij}(y)\,p(t,y \mid s,x)\Big),$$

where

$$A_i(y) = \sum_{\ell=1}^{r}\nu_{i\ell}\,a_\ell(y), \qquad B_{ij}(y) = \sum_{\ell=1}^{r}\nu_{i\ell}\,\nu_{j\ell}\,a_\ell(y),$$

and the initial condition is $p(s,y \mid s,x) = \delta_x(y)$. This PDE describes the evolution of the probability density under a diffusion approximation. However, rather than working with the probability density directly, it is often more convenient to describe the dynamics in terms of individual stochastic trajectories. The corresponding SDE is the CLE. The stochastic dynamics of the ith chemical

species X_i under the CLE are given by the scalar SDE:

$$\mathrm{d}X_i(t) = \sum_{j=1}^{r} \nu_{i,j} a_j(X(t))\,\mathrm{d}t + \sum_{j=1}^{r} \nu_{i,j} \sqrt{a_j(X(t))}\,\mathrm{d}W_j(t), \quad X_i(0) = x_i(0), \tag{4.4}$$

for $i = 1, \ldots, d$, where $W_j(t)$, $j = 1, \ldots, r$, are independent Brownian motions, and $x_i(0) \in \mathbb{R}_+$ is the initial count of species i. For an intuitive derivation of this SDE, we refer the reader to Golightly and Wilkinson (2011). For a bounded diffusion coefficient and an initial state $x_0 = (x_1(0), \ldots, x_d(0))$, there is a unique solution to (7.1), remaining in $\mathbb{R}_+^d$ with probability one (Mao, 2006). Like the CME, the CLE generally lacks explicit analytical solutions for most systems. However, CLE simulations are computationally more efficient than CME simulations, as their cost scales with the number of species d rather than the frequency of reaction events, as it happens for the MJPs (Gillespie, 1977).

All species entering into the CME and the CLE (7.1) have a non-negative count, i.e., $X_i(t) \geq 0, t \geq 0, i = 1, \ldots, d$, so zero is not an exit boundary, as it may be attained but not crossed. This may not be the case when solving the CLE numerically, as the time discretization of the CLE may yield square roots of negative values, resulting in an ill-defined process (Anderson et al., 2019; Wilkie and Wong, 2008; Szpruch and Higham, 2010; Dana and Raha, 2011; Schnoerr et al., 2014). This is what happens, for example, when considering the commonly used EuM method, which motivates us to derive an alternative boundary-preserving numerical scheme, see Section 6. Negative values in the EuM simulations are handled by either truncating at zero or by taking their absolute values. However, this introduces a bias in the model dynamics, whose quantification and impact on the inference is not easily quantified.

4.4 Conclusions

In this chapter, we introduced CRNs as a modeling framework for stochastic biochemical systems. We discussed how the dynamics of such systems can be described at multiple levels of approximation, beginning with the CME and moving toward the CLE via the Kramers–Moyal expansion and the chemical Fokker–Planck equation. Along the way, we formalized key concepts such as mass-action kinetics, reaction order, and non-mass-action propensity functions, including Michaelis-Menten and Hill kinetics. These tools provide the foundation for both the numerical methods and inference strategies developed in the remainder of the thesis. In Chapter 6 we exploit the CLE as our forward model and develop a numerical method tailored to it. Across the included papers we perform parameter inference for (partially observed, noisy) CLE models using *ABC-MCMC* and *ABC-SMC*.

5 Sequential Monte Carlo and backward simulation

In Chapter 2, we discussed several simulation-based approaches for Bayesian inference in models with intractable likelihoods. Particularly, ABC-SMC relies on a population of parameter particles that are evolved across a sequence of distributions to approximate the posterior. In this chapter, we shift focus to a different use of SMC: as a method for state inference in dynamical models, such as state-space models. Unlike ABC-SMC, the particles here represent latent state trajectories rather than parameters. These SMC methods are particularly well suited for models with temporal structure (Gordon et al., 1993; Kong et al., 1994; Kitagawa, 1996; Kim et al., 1998; Pitt and Shephard, 1999; Doucet et al., 2001; Godsill et al., 2004) .

A defining feature of SMC is the recursive update of a population of weighted particles through propagation, weighting, and resampling steps Doucet et al. (2001). A central advantage of this approach is its ability to process data in a forward fashion, making it ideal for filtering problems. However, tasks such as trajectory smoothing require processing information backwards in time as well. This motivates the use of backward simulation, which reconstructs latent trajectories by sampling from approximate smoothing distributions. In what follows, we introduce the formal setup for state-space models, review particle filtering and lookahead strategies, and then we discuss how backward simulation methods can be built on top of forward filtering to efficiently reconstruct latent trajectories from smoothing distributions.

5.1 State-space models

We consider a class of *state-space models* in which the latent process $(X(t))_{t\geq 0}$ evolves in continuous time, typically governed by an SDE, and noisy observations are available at discrete time points $0 \leq t_1 < \cdots < t_n$. The latent state

$X(t) \in \mathcal{X}$ satisfies the Markov property, with transition dynamics given by the transition density $p_\theta\big(x(t_{l+1}) \mid x(t_l)\big)$, derived either exactly or approximately from the SDE. We keep density arguments and sampling statements in lowercase to match the notation in Chapter 2: $x(t_{l+1}) \mid x(t_l) \sim p_\theta\big(x(t_{l+1}) \mid x(t_l)\big)$ and $y(t_l) \mid x(t_l) \sim p_\theta\big(y(t_l) \mid x(t_l)\big)$. The initial state $x(t_1)$ may be fixed or drawn from an initial distribution $p_1(x)$.

At each observation time t_l, a noisy measurement $Y(t_l) \in \mathcal{Y}$ is observed, depending only on the current latent state $X(t_l)$. The observations $Y(t_{1:n}) = (Y(t_1), Y(t_1), \ldots, Y(t_n))$ are assumed conditionally independent given the latent states $X(t_{1:n}) = (X(t_1), X(t_1), \ldots, X(t_n))$, with observation model $p_\theta\big(y(t_l) \mid x(t_l)\big)$. We denote the realized (observed) values of the measurement process by $y^{\text{o}}(t_l)$; that is, $Y(t_l)$ is a random variable, while $y^{\text{o}}(t_l)$ is its realization. The generative process is defined as follows: draw $x(t_1) \sim p_1(x)$, and then for each $l = 1, \ldots, n-1$, simulate the latent dynamics and the observation via

$$x(t_{l+1}) \mid x(t_l) \sim p_\theta\big(x(t_{l+1}) \mid x(t_l)\big), \qquad y(t_l) \mid x(t_l) \sim p_\theta\big(y(t_l) \mid x(t_l)\big).$$

The conditional independence structure for the random variables is

$$Y(t_l) \perp\!\!\!\perp \big(X(t_j), Y(t_j)\big)_{j \neq l} \,\big|\, X(t_l).$$

the joint distribution over the latent states and observations is given by

$$p_\theta(x(t_{1:n}), y(t_{1:n})) = p_1(x(t_1)) \prod_{l=1}^{n-1} p_\theta\big(x(t_{l+1}) \mid x(t_l)\big) \prod_{l=1}^{n} p_\theta\big(Y(t_l) \mid x(t_l)\big).$$

The goal of inference is to approximate the *filtering distributions* $p_\theta\big(x(t_l) \mid y^{\text{o}}(t_{1:l})\big)$ and the *smoothing distribution* $p_\theta\big(x(t_{1:n}) \mid y^{\text{o}}(t_{1:n})\big)$. In nonlinear and non-Gaussian models, these distributions are intractable, and SMC methods provide a principled framework for their approximation using weighted particles. We now turn to SMC methods for approximating the filtering and smoothing distributions. We begin with particle filtering, which targets the sequence of filtering distributions through a recursive sampling and weighting procedure, and later extend the discussion to particle smoothing methods for reconstructing latent trajectories.

5.1.1 Particle filtering

Let $g_\theta\big(x(t_{1:l}) \mid y^{\text{o}}(t_{1:l})\big)$ denote a sequence of proposal densities, and let $p_\theta\big(x(t_{1:l}) \mid y^{\text{o}}(t_{1:l})\big)$ denote the corresponding sequence of target densities. In a standard IS scenario, we can generate P independent samples $x^j(t_{1:l})$, for

$j = 1, ..., P$, from the proposal density and assign importance weights to these samples via

$$\omega_l^j \propto \frac{p_\theta\big(x^j(t_{1:l}) \mid y^o(t_{1:l})\big)}{g_\theta\big(x^j(t_{1:l}) \mid y^o(t_{1:l})\big)}.$$

These weights are computed only up to proportionality, as the normalizing constant of the target is typically intractable. For simplicity, we reuse the notation ω_l^j to denote the normalized weights. In high-dimensional state spaces, such as those arising in time-series models, naive importance sampling becomes increasingly inefficient due to the curse of dimensionality (Doucet et al., 2001; Del Moral et al., 2006). To address this, a sequential implementation is sought, and the proposal density is assumed to be factorizable as

$$g_\theta\big(x(t_{1:l}) \mid y^o(t_{1:l})\big) = g_\theta\big(x(t_l) \mid x(t_{l-1}), y^o(t_l)\big) g_\theta\big(x(t_{1:l-1}) \mid y^o(t_{1:l-1})\big),$$

or recursively as

$$g_\theta\big(x(t_{1:l}) \mid y^o(t_{1:l})\big) = g_\theta\big(x(t_1) \mid y^o(t_1)\big) \prod_{s=2}^{l} g_\theta\big(x(t_s) \mid x(t_{s-1}), y^o(t_s)\big).$$

Thus, to sample $x^j(t_{1:l}) \sim g_\theta\big(x(t_{1:l}) \mid y^o(t_{1:l})\big)$, one can start with $x^j(t_1) \sim g_\theta\big(x(t_1) \mid y^o(t_1)\big)$, and then $x^j(t_2) \sim g_\theta\big(x(t_2) \mid x^j(t_1), y^o(t_2)\big)$, iteratively until $x^j(t_n) \sim g_\theta\big(x(t_n) \mid x^j(t_{n-1}), y^o(t_n)\big)$. This factorization enables a recursive formulation of the importance weights, avoiding the need to evaluate high-dimensional joint densities directly. The corresponding target densities can be decomposed as

$$\begin{aligned} &p_\theta\big(x(t_{1:l}) \mid y^o(t_{1:l})\big) \propto \\ &\qquad p_\theta\big(y^o(t_l) \mid x(t_l)\big) p_\theta\big(x(t_l) \mid x(t_{l-1})\big) p_\theta\big(x(t_{1:l-1}) \mid y^o(t_{1:l-1})\big). \end{aligned}$$

With these factorizations in mind, the importance weights take the form

$$\omega_l^j \propto \frac{p_\theta\big(y^o(t_l) \mid x^j(t_l)\big) p_\theta\big(x^j(t_l) \mid x^j(t_{l-1})\big)}{g_\theta\big(x^j(t_l) \mid x^j(t_{l-1}), y^o(t_l)\big)} \frac{p_\theta\big(x^j(t_{1:l-1}) \mid y^o(t_{1:l-1})\big)}{g_\theta\big(x^j(t_{1:l-1}) \mid y^o(t_{1:l-1})\big)}.$$

Hence, if we define the weight function for $l = 2, ..., n$,

$$W_l\big(x(t_{l-1:l}), y^o(t_l)\big) = \frac{p_\theta\big(y^o(t_l) \mid x(t_l)\big) p_\theta\big(x(t_l) \mid x(t_{l-1})\big)}{g_\theta\big(x(t_l) \mid x(t_{l-1}), y^o(t_l)\big)},$$

the importance weights can be recursively computed by

$$\omega_l^j \propto W_l\big(x^j(t_{l-1:l}), y^o(t_l)\big) \omega_{l-1}^j.$$

This recursive formulation defines the *sequential importance sampling (SIS)* procedure. However, SIS suffers from a well-known issue: *weight degeneracy*. Over time, the variance of the particle weights increases, and most particles contribute negligibly to the approximation. To address this, when the effective sample size (ESS) $\mathrm{ESS}_l := 1/\sum_{j=1}^{P}(w_l^j)^2$ of the weights $w_l^{1:P}$ falls below a threshold, a resampling step is triggered, typically using a multinomial scheme, to refresh the particles (Gordon et al., 1993; Kitagawa, 1996; Del Moral et al., 2006; Chopin, 2004). This procedure is called *sequential-importance-resampling (SIR)*. An algorithmic description is shown in Algorithm 3.

Algorithm 3 Particle filter $(\theta, y^{\mathrm{o}}(t_{1:n}), \tau)$

for $j = 1$ to P **do**
  Sample initial state $x^j(t_1) \sim p_1(x(t_1))$
  Compute weight $\omega_1^j \propto W_1\big(x^j(t_1), y^{\mathrm{o}}(t_1)\big)$
end for
Normalize weights $\omega_1^{1:P}$
for $l = 2$ to n **do**
  Compute $\mathrm{ESS}_{l-1} \leftarrow 1/\sum_{j=1}^{P}(\omega_{l-1}^j)^2$
  if $\mathrm{ESS}_{l-1} < \tau$ **then**
    Resample ancestor indices $a_l^{1:P} \sim \mathrm{Categorical}(\omega_{l-1}^{1:P})$
    Reset weights $\omega_{l-1}^{1:P} \leftarrow 1/P$
  else
    Set $a_l^j \leftarrow j$ for $j = 1, \ldots, P$ (no resampling)
  end if
  for $j = 1$ to P **do**
    Set $x^j(t_{l-1}) \leftarrow x^{a_l^j}(t_{l-1})$
    Sample $x^j(t_l) \sim g_\theta\big(x(t_l) \mid x^j(t_{l-1}), y^{\mathrm{o}}(t_l)\big)$
    Compute weight $\omega_l^j \propto W_l\big(x^j(t_{l-1:l}), y^{\mathrm{o}}(t_l)\big)\,\omega_{l-1}^j$
  end for
  Normalize weights $\omega_l^{1:P}$
end for
Output: Weighted particles $\big(x^{1:P}(t_{1:n}), \omega_n^{1:P}\big)$

A common and simple choice of proposal density is the model transition density itself:

$$g_\theta\big(x(t_l) \mid x(t_{l-1}), y^{\mathrm{o}}(t_l)\big) = p_\theta\big(x(t_l) \mid x(t_{l-1})\big).$$

Under this choice, the weight function reduces to the likelihood term:

$$W_l\big(x(t_l), y^{\mathrm{o}}(t_l)\big) = p_\theta\big(y^{\mathrm{o}}(t_l) \mid x(t_l)\big),$$

and the resulting algorithm is known as the *bootstrap particle filter* (Gordon et al., 1993). One can use more informed proposal distributions that take the upcoming observation into account, for example, the so-called optimal proposal, given by:

$$g_\theta\big(x(t_l) \mid x(t_{l-1}), y^{\text{o}}(t_l)\big) = p_\theta\big(x(t_l) \mid x(t_{l-1}), y^{\text{o}}(t_l)\big), \tag{5.1}$$

which conditions not only on the previous state but also on the current observation. This proposal minimizes the variance of the importance weights at time t_l, however, it is often intractable to sample from or evaluate in general SDE models. More broadly, in settings where the observations are highly informative (for example if there is no measurement error, as considered in *Paper I*) or where the prior and posterior are poorly aligned, naive sampling and weighting strategies can lead to weight degeneracy. To mitigate this, *lookahead* strategies have been proposed, where the proposal distribution or weighting function is adapted using information from future observations. The optimal proposal is one such instance, but more flexible schemes include approximate lookahead, guided proposals, and bridging constructs. One important scheme that we build upon is the bridge particle filter (Del Moral and Murray, 2015).

5.2 Lookahead principles

As alluded to previously, the basic approach can suffer from particle degeneracy when observations are highly informative, since many particles may be drawn from regions of low-probability (Lin et al., 2013). Lookahead strategies aim to mitigate this by incorporating future observations into the sampling or weighting process. In essence, lookahead methods blur the line between filtering (using data up to the current time) and smoothing (using future data). By "looking" ahead, these methods produce particle approximations that are more informed about the true state. Many dynamical systems exhibit strong memory. For instance, in target tracking applications, measurements obtained *after* time t can improve inference of the target's position at time t. Similarly, the construction of polymer conformations can be formulated as a stochastic dynamical system with long memory (Rosenbluth and Rosenbluth, 1955), where lookahead techniques have proven effective (Zhang and Liu, 2002). Over the years, various lookahead strategies have been developed. We provide a brief overview of these methods and focus in more detail on the bridge particle filter (Del Moral and Murray, 2015).

In theory, the optimal lookahead strategy would utilize all future observations when sampling or weighting the particles (essentially it would target the densities $p_\theta\left(x(t_l) \mid y^{\text{o}}(t_{1:n})\right)$, for $l = 1, ..., n$.) Even access to exact samples from a partially informed distribution $p_\theta\left(x(t_l) \mid y^{\text{o}}(t_{1:l+L})\right)$ for some lookahead hori-

zon $L \geq 1$, would already yield more accurate inference of the current state, as compared to myopic propagation or weighting. However, this is intractable because it requires computing multistep predictive distributions or integrating out future latent states. For $L = 1$, a principled approximation leads to the auxiliary particle filter (APF, Pitt and Shephard, 1999), which modifies the standard particle filter by incorporating a one-step lookahead into the resampling mechanism. Instead of resampling purely based on current weights, the APF evaluates the expected utility of each particle with respect to the next observation, allowing for more informed resampling. A broad range of lookahead strategies have been proposed that incorporate future observations into the particle propagation or resampling steps. These include lookahead weighting (Clapp and Godsill, 1999), exact lookahead sampling (Chen et al., 2000; Doucet et al., 2006), pilot lookahead sampling (Wang et al., 2002; Zhang and Liu, 2002), multilevel pilot lookahead sampling (Guo et al., 2004), and combinations thereof (Wang et al., 2002). Lin et al. (2010) introduced a backward pilot method that estimates optimal resampling weights by simulating backwards, on a finer discretization of the observational interval $[t_l, t_{l+1}]$, from time t_{l+1}, in order to guide forward particle trajectories. This method cannot handle measurement noise or partially observed data, and these limitations are addressed by more recent methods. The *bridge particle filter* (BPF) accommodates both of these cases by choosing a suitable intermediate particle weighting scheme in such a way that, by looking ahead to the next observation point, those particle trajectories that are not consistent with that observation will be given small weights and pruned by resampling at selected intermediate times (e.g., via an ESS trigger.) More detail will be given in the next subsection. A similar method is the guided intermediate resampling filter (Park and Ionides, 2020) which combines the lookahead approach with intermediate propagation in high-dimensional settings. The iterated APF (Guarniero et al., 2017) is an alternative procedure that recursively approximates the optimal guiding function in the backward direction via mixtures of normals, assuming tractable transition densities. Further approximations to the optimal proposal kernel have been proposed, notably the implicit particle filter (Chorin and Tu, 2009; Morzfeld et al., 2012; Chorin et al., 2013), which generates particles near the mode of the optimal proposal distribution (5.1). The equivalent-weights particle filter guides particles toward the next observation over intermediate time steps by leveraging a local Gaussian approximation (Van Leeuwen, 2010; Ades and Van Leeuwen, 2015). Other approaches include ensemble-Kalman-inspired propagation (Papadakis et al., 2010), and Gaussian flow filters (Bunch and Godsill, 2016).

5.2.1 Bridge particle filter

The BPF (Del Moral and Murray, 2015) modifies the standard particle filter to incorporate future observations into intermediate resampling and weighting steps, reducing degeneracy when final states are highly informative or without measurement error. For illustration purposes, we will consider a generic interval $[0, t]$, with an initial state $x(0)$ and an observation $x(t)$. It is assumed that this observation is without measurement error. Let $0 = \tau_0 < \tau_1 < ... < \tau_A = t$ be a discretization of the interval $[0, t]$ into A subintervals. The bridge particle filter aims to approximate the conditional distribution

$$p_\theta(x(\tau_{1:A-1}) \mid x(0), x(t)) \propto \\ p_\theta(x(t) \mid x(\tau_{A-1})) \cdots p_\theta(x(\tau_2) \mid x(\tau_1)) p_\theta(x(\tau_1) \mid x(0)),$$

by sampling forward from $x(0)$ using a standard proposal and applying importance weights that adjust for the likelihood of reaching $x(t)$. Observe that

$$\frac{p_\theta(x(t) \mid x(\tau_{A-1}))}{p_\theta(x(t) \mid x(0))} = \prod_{k=1}^{A-1} \frac{p_\theta(x(t) \mid x(\tau_k))}{p_\theta(x(t) \mid x(\tau_{k-1}))}.$$

If one defines the weighting functions

$$W_k(x(\tau_{k-1:k}), x(t)) = \frac{p_\theta(x(t) \mid x(\tau_k))}{p_\theta(x(t) \mid x(\tau_{k-1}))}, \tag{5.2}$$

one can incrementally simulate $x(\tau_k) \sim p_\theta(x(\tau_k) \mid x(\tau_{k-1}))$ and weight with $W_k(x(\tau_{k-1:k}), x(t))$ for $k = 1, ..., A-1$. When the weighting functions (5.2) are not available in closed form due to intractable transition densities, suitable approximations can be employed (Del Moral and Murray, 2015). An algorithmic description is shown in Algorithm (4). Figure 5.1 illustrates the BPF for an Ornstein–Uhlenbeck (OU) latent process with Gaussian weights, repeated between multiple consecutive observation points. Measurement noise can also be accommodated in the BPF by incorporating the measurement density into the weights. Specifically, if the observation model is $y(t) \sim p_\theta(y(t) \mid x(t))$, the weight update of the particles in the intermediate stages can be modified as

$$W_k(x(\tau_{k-1:k}), y(t)) = \frac{p_\theta(y(t) \mid x(\tau_k))}{p_\theta(y(t) \mid x(\tau_{k-1}))}.$$

This weighting scheme encourages particles that better align with the noisy observation at time t.

Algorithm 4 Bridge particle filter $(\theta, x(0), x(t), \tau)$

1: Initialize particles: $x^j(0) = x(0)$, and weights $\omega_0^j = 1/P$ for $j = 1, \ldots, P$.
2: **for** $k = 1$ to $A - 1$ **do**
3: Compute $\mathrm{ESS}_{k-1} \leftarrow 1/\sum_{j=1}^{P}(\omega_{k-1}^j)^2$.
4: **if** $\mathrm{ESS}_{k-1} < \tau$ **then**
5: Sample ancestor indices $a_k^{1:P} \sim \mathrm{Categorical}(\omega_{k-1}^{1:P})$
6: Set $x^j(\tau_{k-1}) \leftarrow x^{a_k^j}(\tau_{k-1})$ and $\omega_{k-1}^j \leftarrow 1/P$ for $j = 1, \ldots, P$
7: **end if**
8: **for** $j = 1$ to P **do**
9: Sample $x^j(\tau_k) \sim p_\theta\big(x(\tau_k) \mid x^j(\tau_{k-1})\big)$
10: Update weight: $\omega_k^j \propto \omega_{k-1}^j \cdot \dfrac{p_\theta\big(x(t) \mid x^j(\tau_k)\big)}{p_\theta(x(t) \mid x^j(\tau_{k-1}))}$
11: **end for**
12: Normalize $\omega_k^{1:P}$
13: **end for**
14: **Output:** Weighted particles $(x^{1:P}(\tau_{1:A}), \omega_{1:A}^{1:P})$.

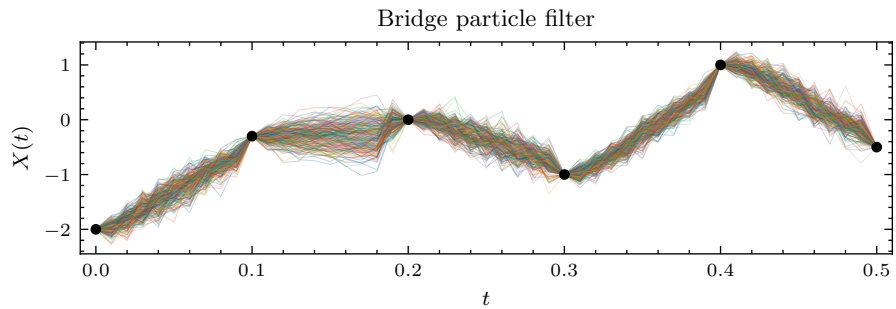

Figure 5.1: Illustration of the BPF with an OU process: how weighting and resampling steers particles through the observations. Black circles depict the observations. The particles are propagated by simulating the OU process exactly, and the exact transition densities are used as weights.

5.3 Backward simulation for sequential Monte Carlo

In many applications involving state-space models, we are interested not only in filtering, but also in smoothing, where the goal is to recover the distribution of latent trajectories conditional on all observations. More formally, given a sequence of observations $y^o(t_{1:n})$, the smoothing distribution is the posterior $p_\theta\big(x(t_{1:n}) \mid y^o(t_{1:n})\big)$. In this section, we describe the forward filtering backward simulation (FFBS) algorithm.

Assume that filtering has already been performed, yielding approximations of the filtering densities $p_\theta\left(x(t_l) \mid y^{\mathrm{o}}(t_{1:l})\right)$, for each time step $l = 1, \ldots, n$, represented by the particle systems $\left(x^{1:P}(t_l), \omega_l^{1:P}\right)$. More precisely, for $l = 1, ..., n$, the particle systems yield the approximated filtering densities

$$\hat{p}_\theta\left(x(t_l) \mid y^{\mathrm{o}}(t_{1:l})\right) = \sum_{j=1}^{P} \omega_l^j \delta_{x^j(t_l)}(x(t_l)). \tag{5.3}$$

The primary goal of backward simulation in the context of particle smoothing is to generate samples from the smoothing density to gain insight about the latent stochastic process. The smoothing distribution can be factorized as

$$p_\theta\big(x(t_{1:n}) \mid y^{\mathrm{o}}(t_{1:n})\big) = p_\theta\big(x(t_n) \mid y^{\mathrm{o}}(t_{1:n})\big) \prod_{l=1}^{n-1} p_\theta\big(x(t_l) \mid x(t_{l+1:n}), y^{\mathrm{o}}(t_{1:n})\big), \tag{5.4}$$

where, under the Markov property of the latent process and the conditional independence of the observations, we have

$$\begin{aligned} p_\theta\big(x(t_l) \mid x(t_{l+1:n}), y^{\mathrm{o}}(t_{1:n})\big) &= p_\theta\big(x(t_l) \mid x(t_{l+1}), y^{\mathrm{o}}(t_{1:l})\big) \\ &\propto p_\theta\big(x(t_l) \mid y^{\mathrm{o}}(t_{1:l})\big)\, p_\theta\big(x(t_{l+1}) \mid x(t_l)\big). \end{aligned}$$

Interestingly, this shows that, conditionally on $y^{\mathrm{o}}(t_{1:n})$, the latent process $X(t_{1:n})$ forms an inhomogeneous Markov process. This property is important when feeding the trajectory into PEN, as described in Section 2.1.6.

The key ingredient in backward simulation is the *backward kernel*

$$\mathcal{B}_l(\mathrm{d}x \mid x(t_{l+1})) = \mathbb{P}\big(x(t_l) \in \mathrm{d}x \mid x(t_{l+1}), y^{\mathrm{o}}(t_{1:l})\big), \tag{5.5}$$

which admits the density

$$p_\theta\big(x(t_l) \mid x(t_{l+1}), y^{\mathrm{o}}(t_{1:l})\big) \propto p_\theta\big(x(t_{l+1}) \mid x(t_l)\big)\, p_\theta\big(x(t_l) \mid y^{\mathrm{o}}(t_{1:l})\big). \tag{5.6}$$

Using this backward kernel, we obtain the following expression for the backward recursion:

$$p_\theta\big(x(t_{l:n}) \mid y^{\mathrm{o}}(t_{1:n})\big) = p_\theta\big(x(t_l) \mid x(t_{l+1}), y^{\mathrm{o}}(t_{1:l})\big)\, p_\theta\big(x(t_{l+1:n}) \mid y^{\mathrm{o}}(t_{1:n})\big), \tag{5.7}$$

starting from $p_\theta\left(x(t_n) \mid y^{\mathrm{o}}(t_{1:n})\right)$ at the final time point. The recursion in Equation (5.7) implies the following sampling scheme. First, sample the terminal

state from the marginal smoothing distribution:

$$\widetilde{x}(t_n) \sim p_\theta\left(x(t_n) \mid y^{\mathrm{o}}(t_{1:n})\right). \tag{5.8}$$

Then, going backward in time for $l = n-1, \ldots, 1$, recursively sample

$$\widetilde{x}(t_l) \sim p_\theta\left(x(t_l) \mid x(t_{l+1}), y^{\mathrm{o}}(t_{1:l})\right). \tag{5.9}$$

After a full backward sweep, the trajectory $(\widetilde{x}(t_1), \ldots, \widetilde{x}(t_n))$ constitutes a sample from the smoothing distribution $p_\theta\left(x(t_{1:n}) \mid y^{\mathrm{o}}(t_{1:n})\right)$.

The backward kernel density at time t_l depends on the transition density $p_\theta\left(x(t_{l+1}) \mid x(t_l)\right)$, which can be approximated using the methods described in Chapter 3.5, and on the filtering density $p_\theta\left(x(t_l) \mid y^{\mathrm{o}}(t_{1:l})\right)$, which is approximated by the particle system obtained during the forward pass. Substituting the particle approximation of the filtering densities (5.3) into the backward kernel (5.6) yields the approximate backward kernel:

$$\hat{\mathcal{B}}_l(\mathrm{d}x \mid x(t_{l+1})) = \sum_{j=1}^{P} \frac{\omega_l^j \, p_\theta\left(x(t_{l+1}) \mid x^j(t_l)\right)}{\sum_{m=1}^{P} \omega_l^m \, p_\theta\left(x(t_{l+1}) \mid x^m(t_l)\right)} \, \delta_{x^j(t_l)}(\mathrm{d}x). \tag{5.10}$$

We can now make use of these approximations to sample an approximate backward trajectory by sampling $\widetilde{x}(t_n) \sim \hat{p}_\theta(x(t_n) \mid y^{\mathrm{o}}(t_{1:n}))$, and then $\widetilde{x}(t_l) \sim \hat{\mathcal{B}}_l(\tilde{x}(t_l) \mid \widetilde{x}(t_{l+1}))$, for $l = n-1, \ldots, 1$. Figure 5.2 depicts a run of the forward

Algorithm 5 Backward-simulation particle smoother $((x^{1:P}(t_{1:n}), \omega_{1:n}^{1:P}), \theta)$

1: Sample particle index $j \sim \text{Categorical}(\omega_n^{1:P})$ and set $\widetilde{x}(t_n) = x^j(t_n)$.
2: **for** $l = n-1$ to 1 **do**
3: **for** $j = 1$ to P **do**
4: Compute $\widetilde{\omega}_l^j \propto \omega_l^j \, p_\theta\left(\widetilde{x}(t_{l+1}) \mid x^j(t_l)\right)$.
5: **end for**
6: Normalize the smoothing weights $\widetilde{\omega}_l^{1:P}$.
7: Sample particle index $j \sim \text{Categorical}(\widetilde{\omega}_l^{1:P})$ and set $\widetilde{x}(t_l) = x^j(t_l)$.
8: **end for**
9: **Output:** Backward trajectory $\widetilde{x}(t_{1:n})$.

filtering backward smoothing algorithm (Algorithm 3 followed by Algorithm 5) for the Ornstein–Uhlenbeck model with Gaussian noise.

With this, the smoothing density $p_\theta\left(x(t_{1:n}) \mid y^{\mathrm{o}}(t_{1:n})\right)$ can be approximated by sampling multiple backward trajectories $\widetilde{x}^{1:P}(t_{1:n})$ by repeating Algorithm 5 P times. Conditionally on the forward filter particles, the trajectories $\widetilde{x}^{1:P}(t_{1:n})$

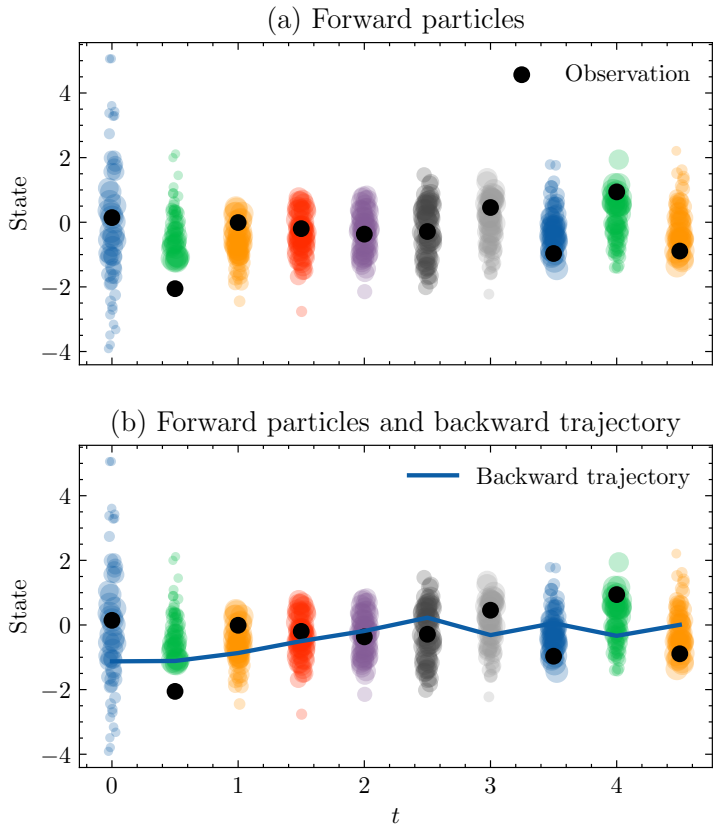

Figure 5.2: Forward bootstrap particle filter and backward simulation smoother for the Ornstein–Uhlenbeck model with Gaussian noise. Black circles are observations. (a) Forward particle cloud at each time point; marker size is proportional to the weight. (b) Same cloud with a single trajectory sampled from the smoothing distribution (FFBS).

are i.i.d., and define a point-mass approximation

$$\hat{p}_\theta\left(\widetilde{x}(t_{1:n}) \mid y^{\mathrm{o}}(t_{1:n})\right) = \sum_{j=1}^{P} \delta_{\widetilde{x}^j(t_{1:n})}(\widetilde{x}(t_{1:n})). \tag{5.11}$$

In *Paper I*, we make use of the output from the backward simulator and feed it into the ABC algorithm. Naturally, the summary statistics computed from this trajectory $S(\widetilde{x}(t_{1:n}))$, do not follow the same distribution as $s \sim p_\theta(s)$ under the forward model, because $\widetilde{x}(t_{1:n})$ is not a sample from the forward model.

This raises the important question:

> *What is the distribution of $S(\widetilde{x}(t_{1:n}))$, and are we able to approximate it efficiently in a form suitable for ABC-SMC or ABC-MCMC?*

To answer this question we first need to look into the point-mass approximation of the distribution of $\widetilde{x}(t_{1:n})$. By plugging the approximations of the filtering densities (5.3) and the backward kernels (5.5) into the joint smoothing density (5.4), we obtain the following approximation:

$$\hat{p}_\theta(\widetilde{x}(t_{1:n}) \mid y^{\mathrm{o}}(t_{1:n})) = \sum_{j_1=1}^{P} \cdots \sum_{j_n=1}^{P} \left(\prod_{l=1}^{n-1} \frac{\omega_l^{j_l}\, p_\theta(x^{j_{l+1}}(t_{l+1}) \mid x^{j_l}(t_l))}{\sum_{k=1}^{P} \omega_l^k\, p_\theta(x^{j_{l+1}}(t_{l+1}) \mid x^k(t_l))} \right) \omega_n^{j_n} \times \delta_{(x^{j_1}(t_1),\ldots,x^{j_n}(t_n))}(\widetilde{x}(t_{1:n})). \quad (5.12)$$

This defines a discrete distribution on $\mathcal{X}^n$, and can be interpreted as follows: at each observation time t_l, $l = 1, \ldots, n$, the particles $x^{1:P}(t_l)$ generated in the forward pass form a set of P elements in $\mathcal{X}$. By selecting one particle at each time point, we construct a trajectory $(x^{j_1}(t_1), x^{j_2}(t_2), \ldots, x^{j_n}(t_n)) \in \mathcal{X}^n$. There are P^n such trajectories corresponding to all combinations of indices $j_1, \ldots, j_n \in \{1, \ldots, P\}$. Although the explicit evaluation of (5.12) is computationally expensive, it establishes a direct connection to the backward simulation procedure in Algorithm 5. The approximation $\hat{p}_\theta(x(t_{1:n}) \mid y^{\mathrm{o}}(t_{1:n}))$ implicitly depends on the forward particle system $\big(x^{1:P}(t_{1:n}), \omega^{1:P}(t_{1:n})\big)$, and is thus more precisely written as

$$\hat{p}_\theta(x(t_{1:n}) \mid y^{\mathrm{o}}(t_{1:n}), (x^{1:P}(t_{1:n}), \omega_{1:n}^{1:P})).$$

Hence, conditionally on the particle system from the forward pass, the backward simulator generates i.i.d. Markovian trajectories from the distribution (5.12). To this end, the summary statistics likelihood $p_\theta(S(\widetilde{x}(t_{1:n})) \mid y^{\mathrm{o}}(t_{1:n}))$ can be approximated using the synthetic likelihood method, using a Gaussian approximation. The forward pass is run once per parameter θ, after which the particle system $\big(x^{1:P}(t_{1:n}), \omega^{1:P}(t_{1:n})\big)$ can be held fixed while the additional P backward trajectories are sampled.

Several methods have been developed to reduce the computational burden of FFBS. One approach uses rejection sampling to avoid computing all backward weights (Douc et al., 2011), while early stopping rules can truncate the backward pass adaptively (Taghavi et al., 2013). Metropolis–Hastings steps have also been introduced to approximately sample from the backward kernel (Bunch and Godsill, 2012; Dubarry and Douc, 2011). Rao-Blackwellization has also been explored (Kok et al., 2024). While more advanced backward-

simulation algorithms exist, we adopt Algorithm 5 for its simplicity. Given our relatively small number of particles, the added complexity of more sophisticated methods is unnecessary.

5.4 Conclusions

We presented SMC methods for state-space models with a particular emphasis on approximating filtering and smoothing distributions. We began with a general review of particle filtering, showing how different choices for the proposal distribution lead to well-known variants such as the bootstrap filter and the auxiliary particle filter. We then briefly reviewed lookahead strategies, culminating in the bridge particle filter, which performs intermediate resampling across a discretized time interval to better align particle trajectories with future observations. Finally, we described how backward simulation can be built on top of forward filtering to reconstruct entire latent trajectories.

These ideas underpin the data-conditional simulation framework developed in this thesis: in *Paper I*, we use a weighting scheme inspired by the BPF for exact observations to weight trajectories at every observational time, and extract a single trajectory by backward simulation; in *Paper II*, we handle noisy and partial observations by using the observational density to weight the particles, and extract a single trajectory by directly sampling from the filtering densities. In both papers, the data-conditional simulators do not use resampling in the forward direction. In *Paper III*, we utilize the bootstrap particle filter in the forward direction with resampling, and extract trajectories by sampling directly from the filtering densities.

6 Splitting schemes

Splitting methods are a class of geometric numerical integrators that exploit the structure of a dynamical system by decomposing it into simpler sub-problems whose flows can be solved more easily or even in closed form. Originally developed in the context of Hamiltonian systems and geometric integration (Hairer et al., 2006; McLachlan and Quispel, 2002), they have since become widely used across physics, chemistry, and applied probability. An important advantage of these methods is their ability to preserve qualitative properties of the underlying dynamics, such as symplecticity in Hamiltonian systems, invariants of motion, or recurrent behaviors such as oscillations and limit cycles. For instance, structure-preserving splitting schemes have been successfully applied to Hodgkin–Huxley type neuronal models, where they retain the oscillatory behavior and limit cycles inherent in the system (Chen et al., 2020).

6.1 Lie–Trotter and Strang compositions

Consider an initial value problem given by the dynamical system:

$$\mathrm{d}x(t) = f(x(t))\,\mathrm{d}t; \quad x(0) = x_0; \quad t \in [0, T] \tag{6.1}$$

where $f : \mathbb{R}^d \to \mathbb{R}^d$ is a (generally nonlinear) vector field, $x : [0, T] \to \mathbb{R}^d$ is the trajectory, and $T > 0$ denotes the final integration time. In general, for a nonlinear vector field f, the flow (solution) of (6.1) cannot be written in closed form. Hence we approximate the solution at discrete time points $\tau_k = kh$, $k \in \mathbb{N}$, where $h > 0$ is a fixed step size. We denote these approximations by $x_{1:d,k} \approx x(\tau_k)$. We illustrate splitting methods by decomposing the system into two additive components, upon which we introduce the the Lie–Trotter and Strang compositions. While we restrict to two components here for clarity, in practice the vector field can be split into an arbitrary number of subfields $f^{(1)}, \ldots, f^{(m)}$, and the corresponding flows $\varphi_h^{(1)}, \ldots, \varphi_h^{(m)}$ can be composed in analogous ways to construct higher-order integrators. Indeed, this is the

approach in (Chen et al., 2020); details are given in subsection 6.3.

We split the vector field into two additive components,

$$\mathrm{d}x(t) = f^{(1)}(x(t))\,\mathrm{d}t + f^{(2)}(x(t))\,\mathrm{d}t; \quad x(0) = x_0; \quad t \in [0, T], \tag{6.2}$$

where $f^{(j)} : \mathbb{R}^d \to \mathbb{R}^d$, $j = 1, 2$. Let $\varphi_h^{(j)}$ denote the exact flow of subsystem j over one step of size h. Then, given an approximation $x_{1:d,k} \approx x(\tau_k)$ at time $\tau_k = kh$, the next step is obtained by composing subflows. The *Lie–Trotter splitting* reads

$$x_{1:d,k+1} = \varphi_h^{(2)} \circ \varphi_h^{(1)}(x_{1:d,k}), \tag{6.3}$$

which is first-order accurate, while the symmetric *Strang splitting* achieves second order,

$$x_{1:d,k+1} = \varphi_{h/2}^{(1)} \circ \varphi_h^{(2)} \circ \varphi_{h/2}^{(1)}(x_{1:d,k}). \tag{6.4}$$

For illustration, and as a prototype for the stochastic setting, it is common to split the system into a linear and a nonlinear part, since both subsystems may admit closed-form flows.

6.2 Linear-nonlinear splitting

Consider the splitting

$$f(x(t)) = Ax(t) + N(x(t)), \quad A \in \mathbb{R}^{d\times d},\ N : \mathbb{R}^d \to \mathbb{R}^d, \tag{6.5}$$

and define the sub-vector fields $f^{(1)}(x(t)) = Ax(t)$ and $f^{(2)}(x(t)) = N(x(t))$. This induces the sub-systems

$$\begin{aligned}
\mathrm{d}x^{(1)}(t) &= Ax^{(1)}(t)\,\mathrm{d}t; \quad x^{(1)}(0) = x_0^{(1)}; \quad t \in [0, T],\\
\mathrm{d}x^{(2)}(t) &= N(x^{(2)}(t))\,\mathrm{d}t; \quad x^{(2)}(0) = x_0^{(2)}; \quad t \in [0, T].
\end{aligned}$$

The first system can be solved exactly as $x^{(1)}(t) = e^{At}x_0^{(1)}$, yielding the h-time flow $x^{(1)}(\tau_{k+1}) = \varphi_h^{(1)}(x^{(1)}(\tau_k))$. If we assume that the nonlinear system can also be solved exactly, we obtain $x^{(2)}(t) = g(x_0^{(2)}, t)$, which defines the h-time flow $x^{(2)}(\tau_{k+1}) = \varphi_h^{(2)}(x^{(2)}(\tau_k))$. The Lie–Trotter and Strang splittings then read

$$\begin{aligned}
x_{1:d,k+1}^{\mathrm{LT}} &= (\varphi_h^{(1)} \circ \varphi_h^{(2)})(x_{1:d,k}^{\mathrm{LT}}),\\
x_{1:d,k+1}^{\mathrm{S}} &= (\varphi_{h/2}^{(2)} \circ \varphi_h^{(1)} \circ \varphi_{h/2}^{(2)})(x_{1:d,k}^{\mathrm{S}}) = g\big(e^{Ah} g(x_{1:d,k}^{\mathrm{S}}, h/2), h/2\big)\,.
\end{aligned}$$

This splitting has been considered in the case of nonlinear SDEs with additive noise (Buckwar et al., 2022)

$$\mathrm{d}X(t) = \mu(X(t))\,\mathrm{d}t + \sigma\,\mathrm{d}W(t). \tag{6.6}$$

where $\mu : \mathbb{R}^d \to \mathbb{R}^d$, $\mu(X(t)) = AX(t) + N(X(t))$, $\sigma : \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$, and $W(t)$ d-dimensional Brownian motion. The vector field μ can be split as for the ODE, but the noise term is included in the linear sub-equation. This gives us two sub-systems,

$$\mathrm{d}X^{(1)}(t) = AX^{(1)}(t)\,\mathrm{d}t + \sigma\,\mathrm{d}W(t), \quad X^{(1)}(0) = x_0^{(1)}, \quad t \in [0,T], \tag{6.7}$$

$$\mathrm{d}X^{(2)}(t) = N(X^{(2)}(t))\,\mathrm{d}t, \quad X^{(2)}(0) = x_0^{(2)}, \quad t \in [0,T]. \tag{6.8}$$

The second sub-equation and its flow remain unchanged. The first sub-equation has exact solution

$$X^{(1)}(t) = e^{At}x_0^{(1)} + \int_0^t e^{A(t-s)}\sigma\,\mathrm{d}W(s), \tag{6.9}$$

where the stochastic integral is Gaussian with mean 0 and covariance

$$C(t) = \int_0^t e^{A(t-s)}\sigma\sigma^T(e^{A^{(t-s)}})^T\,\mathrm{d}s. \tag{6.10}$$

In many cases the covariance admits a closed-form expression, readily obtained with symbolic computation software such as Mathematica or Maple. The h-time flow is given by $X^{(1)}(\tau_{k+1}) = \varphi_h^{(1)}(X^{(1)}(\tau_k)) = e^{Ah}X^{(1)}(\tau_k) + \xi_k$, with $\xi_k \sim \mathcal{N}(0, C(h))$ i.i.d. The resulting schemes are

$$X^{\mathrm{LT}}_{1:d,k+1} = e^{Ah}g(X^{\mathrm{LT}}_{1:d,k}, h) + \xi_k, \tag{6.11}$$

$$X^{\mathrm{S}}_{1:d,k+1} = g\big(e^{Ah}g(X^{\mathrm{S}}_{1:d,k}, h/2) + \xi_k,\, h/2\big)\,. \tag{6.12}$$

However, complications arise when the diffusion term involves square roots of the state, as in multidimensional systems such as the CLE (7.1). In such cases the direct splitting into linear and nonlinear parts is no longer tractable. To address this, we instead adopt a different strategy based on conditionally linear splitting for ODEs, which extend to the stochastic setting in *Paper II*. In the following subsection we review the conditionally linear splitting of ODEs and show that they can be readily applied to a deterministic description of a CRN.

6.3 Splitting schemes for conditionally linear ODEs

Consider a system of ODEs (6.1) where $f : \mathbb{R}^d \to \mathbb{R}^d$ is the vector field with components $f_i(x(t)) = a_i(x)x_i(t) + b_i(x(t))$, for $i = 1, \ldots, d$, i.e.

$$\mathrm{d}x_i(t) = (a_i(x(t))x_i(t) + b_i(x(t)))\,\mathrm{d}t, \quad i = 1, \ldots, d, \tag{6.13}$$

where a_i, b_i are functions depending on x_j for $j \neq i$. These systems have the property that, if all $x_j, j \neq i$ are fixed (i.e., they are considered to be constant), then x_i satisfies a first-order linear ODE, which can be solved exactly. Hence, Chen et al. (2020) proposed to split (6.1) as

$$\mathrm{d}x^{(j)}(t) = f^{(j)}(x(t))\,\mathrm{d}t, \quad j = 1, \ldots, d, \tag{6.14}$$

where the sub-vector fields $f^{(j)} : \mathbb{R}^d \to \mathbb{R}^d$ are given by

$$f_i^{(j)}(x(t)) = \begin{cases} f_i(x(t)) = a_i(x(t))x_i(t) + b_i(x(t)) & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \tag{6.15}$$

such that $f = f^{(1)} + \ldots + f^{(d)}$. Hence, when solving (6.14), all components of $x_i^{(j)}, i \neq j$ are constant, while that for $i = j$ is obtained by solving the linear ODE in (6.15). Subsequently, the solution of (6.1) in τ_k starting at time τ_{k-1} can be obtained by composing the flows (solutions) of the d subequations (6.14) via the Lie-Trotter or Strang compositions. The obtained splitting scheme effectively preserves limit cycles, see Chen et al. (2020).

6.3.1 Illustration for (deterministic) CRNs

CRNs can also be described using ODEs, which provide a macroscopic, deterministic view of the dynamics. Interestingly, many ODE descriptions of CRNs exhibit a conditionally linear structure. For example, the Repressilator model, a popular CRN model, is a 6-dimensional system given by

$$\mathrm{d}P_i = \beta(M_i - P_i)\,\mathrm{d}t \tag{6.16}$$

$$\mathrm{d}M_i = (\alpha_0 + H(P_j) - M_i)\,\mathrm{d}t, \quad j = (i + 1 \mod 3) + 1 \tag{6.17}$$

for $i = 1, 2, 3$, where $H(P_j) = \alpha K^n/(K^n + P_j^n)$ is the Hill function (Gesztelyi et al., 2012). By looking at these equations, it is clear that the Repressilator exhibits conditional linearity, as the ODEs (6.16) are linear, while those in (6.17) are conditionally linear in M_i for fixed P_j. Moreover, it has also oscillatory behavior, so it is a compelling example for applying the splitting integrator of Chen et al. (2020). The preservation of the limit cycle by the splitting scheme

is demonstrated in Figure 6.1, bottom row, while the commonly used Runge-Kutta method fails in doing so for larger time steps.

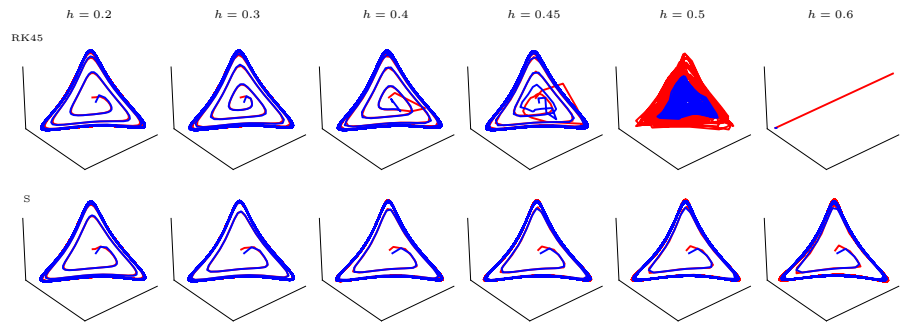

Figure 6.1: Deterministic Repressilator model: numerical solution of the ODEs (6.16)-(6.17) with different time steps h using the Runge-Kutta method (top row) and conditionally linear splitting with Strang composition (bottom row). The 3D trajectories of mRNAs $(M_1(t), M_2(t), M_3(t))$ are shown in blue, and the 3D trajectories of proteins $(P_1(t), P_2(t), P_3(t))$ are shown in red. It can be observed that the limit cycle is preserved by the splitting scheme even at large time steps.

6.4 Conclusions

We reviewed classical Lie–Trotter and Strang splittings for ODEs and their stochastic analogues under a linear–nonlinear decomposition, then introduced conditionally linear splitting (componentwise exact subflows) and illustrated its structure-preserving behavior on deterministic CRNs (e.g., the Repressilator). In *Paper II*, we extend this idea to the CLE. Although the CLE is not conditionally linear, conditioning on the remaining coordinates yields subproblems that are *perturbed Cox–Ingersoll–Ross*-type SDEs (square-root diffusions with linear drift and additional Brownian motions). This enables us to solve each subsystem numerically with advanced CIR integrators (e.g., positivity-preserving schemes or exact/closed-form transition samplers where available) and then compose the subflows via *Lie–Trotter* or *Strang* composition, selected according to the application.

7 Summary of included papers

The remainder of this thesis consists of the included papers. Before presenting them, we briefly summarize their contributions.

Paper I

We seek to infer parameters θ from discrete (fully observed, no measurement noise) observations $x^{\mathrm{o}} = (x^{\mathrm{o}}(t_0), \ldots, x^{\mathrm{o}}(t_n))$ of the Itô diffusion $(X(t))_{t \geq 0}$ satisfying the SDE

$$\mathrm{d}X(t) = \mu\ (X(t), \theta)\ \mathrm{d}t + \sigma\ (X(t), \theta)\ \mathrm{d}W(t), \qquad X(0) = x(0),$$

where $W(t)$ is a standard Brownian motion. Observations occur at $t_i = i\Delta$ with fixed $\Delta > 0$. We perform Bayesian inference for θ via ABC-SMC.

For the simulator, we refine each interval $[t_i, t_{i+1}]$ into A substeps of size $h = \Delta/A$ and set $\tau_{iA+k} = t_i + kh$ for $k = 0, \ldots, A$. Throughout, we use EuM as the forward simulator on the fine grid with step $h = \Delta/A$. The simulated value at time τ_{iA+k} is denoted by x_{iA+k}. Taking every A-th value recovers simulated states at the observation times $t_0, \ldots, t_n$ for use in ABC-SMC. We refer to the baseline ABC–SMC scheme that uses only this forward simulator as *ABC–SMC–F*.

In this paper we introduce a *data-conditional* (DC) simulator for SDEs that generates trajectories consistent with the observation, and embed it in an ABC–SMC algorithm with sequentially learned summary statistics. We refer to this algorithm as *ABC-SMC-DC*. In Figure 7.1 we show a diagram outlining the structure and flow of our proposed inference pipeline.

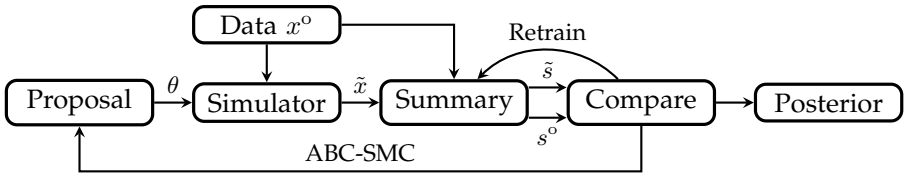

Figure 7.1: Diagram of dynamic ABC-SMC with data-conditional simulation. Note the dependence of the simulator on the observed trajectory x^o.

The DC simulator is based on the forward/backward idea. In the forward direction it utilizes a lookahead sequential importance sampling (LSIS) step to weight a set of particles according to the observation. The particles are propagated on the fine grid using the forward simulator, and assigned a lookahead weight $q(x^o(t_{i+1}) \mid x_{iA+k})$ which depends on the next observation. See Figure 7.2 for an illustration.

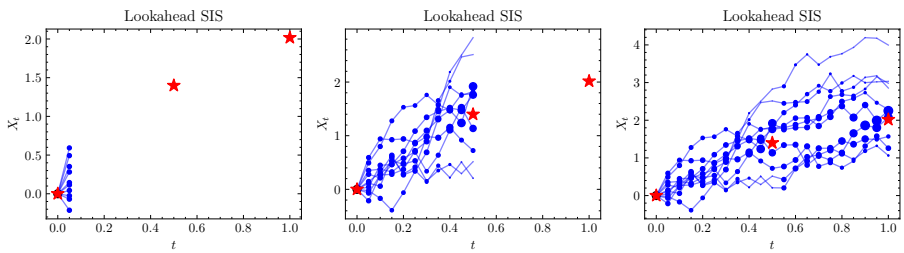

Figure 7.2: Evolution of a particle system obtained from the Lookahead SIS algorithm. Observations (red stars) and particles (blue). The size of each particle is proportional to its weight.

In the backward direction the standard FFBS algorithm is used to extract a trajectory. See Figure 7.3 for an illustration.

Because $\tilde{x}$ is sampled from the conditional distribution $p_\theta(x \mid x^o)$ rather than the forward distribution $p_\theta(x)$, the induced summaries $\tilde{s} = S(\tilde{x})$ follow $p_\theta(s \mid x^o)$ instead of $p_\theta(s)$. In the ABC–SMC algorithm we correct for this with an importance factor involving $p_\theta(s)/p_\theta(s \mid x^o)$, which we approximate efficiently via synthetic likelihoods fitted from forward and backward trajectories.

Within both ABC–SMC-F and ABC-SMC-DC, summaries are learned sequentially via a partially exchangeable network trained on *forward* trajectories (the closest, by Euclidean distance, among the forward paths simulated per parameter), not on data-conditional paths.

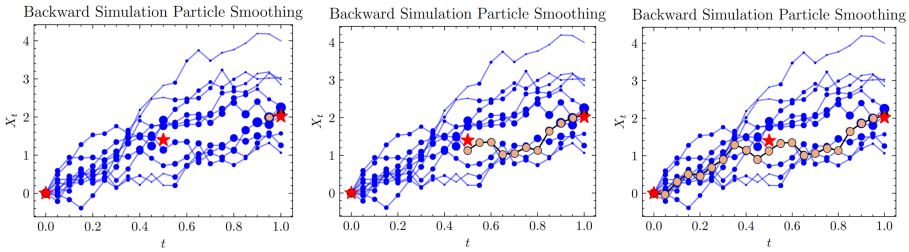

Figure 7.3: Evolution of a trajectory (apricot color) that is obtained via the backward simulation particle smoother on the particle system in Figure 7.2.

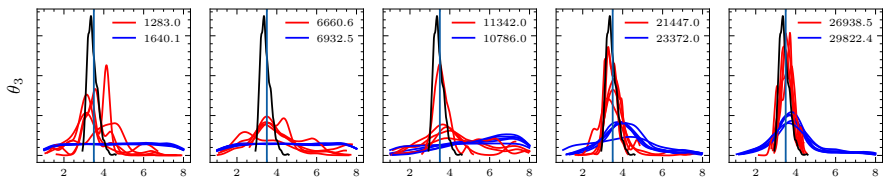

Figure 7.4: Schlögl model. Left to right: rounds 1, 6, 9, 16, and the final round; marginal posterior distributions for ABC-SMC-DC displayed in red, and ABC-SMC-F in blue. The reference marginal posteriors (via particle MCMC) are depicted in black, and the true parameter values as black vertical lines. Each panel reports, in the upper right corner, the number of seconds since the algorithm started.

In experiments the DC simulator markedly increases acceptance rates and concentrates the posterior in the early rounds. Perhaps the most striking result is for the Schlögl model, a stochastic bistable system that we mentioned in the Introduction in Figure 1.1, which is notoriously difficult for ABC. While ABC–SMC-DC accelerates inference, the key gain is the recovery of a well–concentrated posterior for θ_3, whereas ABC–SMC-F fails due to highly variable simulated trajectories.

Paper II

We seek to infer parameters θ from partial and noisy $y^{\text{o}} = (y^{\text{o}}(t_0), \ldots, y^{\text{o}}(t_n))$ of a CRN modeled by the CLE. Let d be the number of chemical species and r the number of reactions. The stochastic dynamics of the ith chemical species

X_i under the CLE are given by the scalar SDE

$$\mathrm{d}X_i(t) = \sum_{j=1}^{r} \nu_{i,j} a_j(X(t))\,\mathrm{d}t + \sum_{j=1}^{r} \nu_{i,j}\sqrt{a_j(X(t))}\,\mathrm{d}W_j(t), \quad X_i(0) = x_i(0), \tag{7.1}$$

where $i = 1, \ldots, d$ for d chemical species, $a_j(\cdot)$ is the propensity function for reaction j, $W_j(t)$, $j = 1, \ldots, r$, are independent Brownian motions, and $x_i(0) \in \mathbb{R}_+$. The parameter vector $\theta = (k_1, k_2, \ldots, k_r, \lambda)$, includes kinetic rate constants k_j for each reaction j, as well as additional elements $\lambda \in \Lambda \subseteq \mathbb{R}^p$, that may represent, for example, Hill constants and the variance of the measurement noise.

Observations occur at $t_i = i\Delta$ and may be partial and noisy:

$$y^{\mathrm{o}}(t_i) = LX(t_i) + \xi_i, \quad \xi_i \sim \mathcal{N}_{d_o}(0, \Sigma^{\mathrm{o}}),$$

where L is a $d_{\mathrm{o}} \times d$ matrix that maps the state vectors to the observed components, and Σ^{o} is the covariance matrix of the measurement noise. The corresponding likelihood $p_\theta(y^{\mathrm{o}})$ is intractable: it includes the intractable transition densities of the process $X(t)$, and an integral over all the possible latent trajectories. We therefore perform Bayesian inference for θ via ABC–SMC.

In this paper we extend the data-conditional simulator to handle *partial* and *noisy* observations. We also derive a structure-preserving splitting scheme tailored to the CLE, that preserves nonnegativity and oscillatory behavior, and permits inference with larger time steps. Embedded in ABC–SMC, these additions increase acceptance rates and deliver faster and more stable inference. To asses the impact of the numerical schemes, Figure 7.5 compares marginal posteriors obtained at the final ABC-SMC-F round for the stochastic Repressilator under Euler–Maruyama (blue) versus our splitting scheme (red) as the integration step $h = \Delta/A$ varies ($A \in \{4, 8, 32, 64\}$). The splitting scheme yields stable posterior inference even at coarse integration steps (e.g., $A = 4$, $h = 0.5$), whereas EuM results in biased and unstable posteriors. Figure 7.6 compares phase portraits of the stochastic Lotka–Volterra model for increasing step sizes h: the splitting scheme preserves the oscillatory dynamics across h, whereas Euler–Maruyama distorts the cycles from $h \approx 0.05$ and breaks down at $h = 0.1$.

Paper III

We propose an extension of ABC-MCMC with DC simulation (ABC–MCMC–DC) that improves early chain mixing and reduces rejection rates, especially when small tolerances are used and initial values are far from the posterior mode.

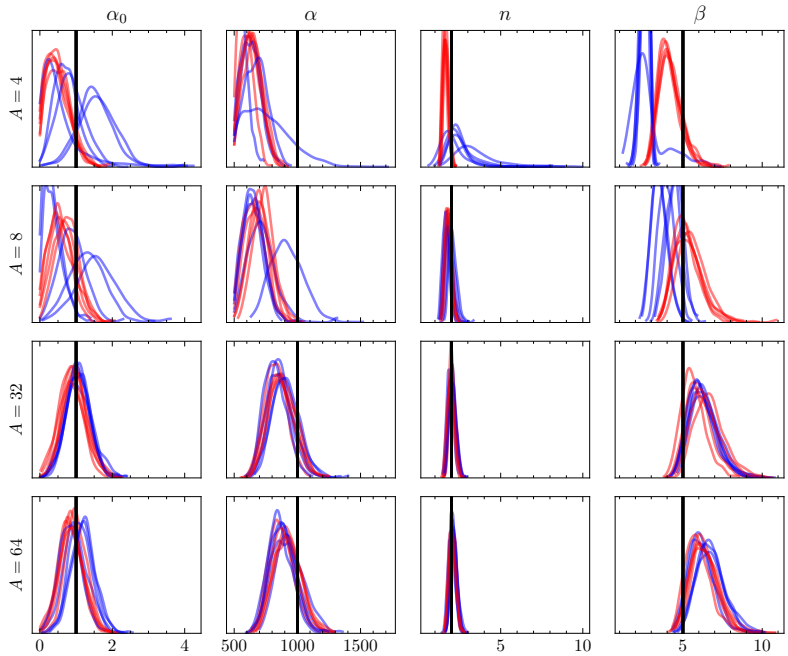

Figure 7.5: Stochastic Repressilator model: marginal posterior distributions using EuM (blue) vs Splitting (red). Rows represent the number of inter-observational intervals and columns represent parameters.

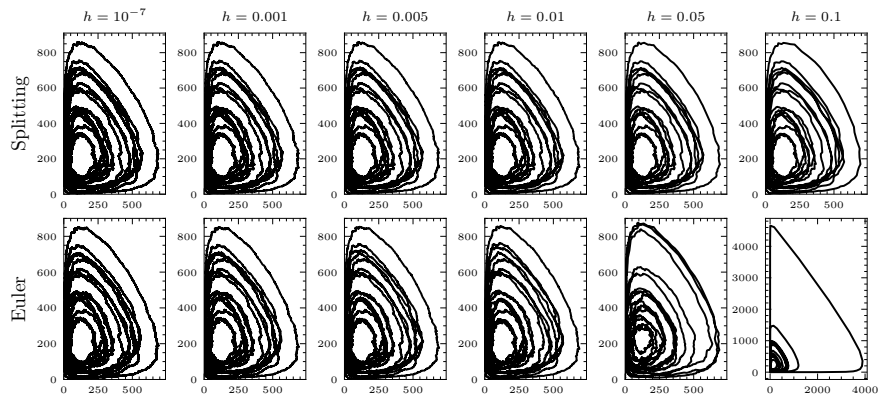

Figure 7.6: Stochastic Lotka–Volterra: phase portraits for increasing time step h. Splitting (top) preserves oscillations across h; Euler–Maruyama (bottom) distorts cycles from $h \approx 0.05$ and fails at $h = 0.1$. Leftmost column shows a near-exact reference ($h = 10^{-7}$).

The method offers a practical way to ease initialization difficulties and increase acceptance probabilities without relaxing the ABC threshold.

Let $y^{\mathrm{o}} = (y^{\mathrm{o}}(t_0), \ldots, y^{\mathrm{o}}(t_n))$ be the observed time series for a state–space model with parameter $\theta \in \Theta$. Standard ABC–MCMC targets the joint ABC posterior

$$\pi_\varepsilon(\theta, s \mid s^{\mathrm{o}}) \propto \pi(\theta)\, p_\theta(s)\, \mathbb{I}\{d(s, s^{\mathrm{o}}) < \varepsilon\},$$

and at each MH step proposes $\theta^* \sim q(\theta^* \mid \theta)$ and $s^* \sim p_{\theta^*}(s)$. In ABC–MCMC–DC we replace the forward draw $s^* \sim p_{\theta^*}(s)$ by a *DC* draw $s^* \sim p_{\theta^*}(s \mid y^{\mathrm{o}})$, constructed with a bootstrap particle filter: we propagate and resample particles against y^{o}, simulate corresponding pseudo-observations along each particle path, and select one trajectory $\tilde{y}$. We then set $s^* = S(\tilde{y}^*)$. This concentrates proposals near the observation. With proposal

$$Q(\theta^*, s^* \mid \theta, s) = q(\theta^* \mid \theta)\, p_{\theta^*}(s^* \mid y^{\mathrm{o}}),$$

the MH acceptance probability for the joint target is

$$\alpha(\theta^*, s^* \mid \theta, s) = \min\left\{1,\ \frac{\pi(\theta^*)\, p_{\theta^*}(s^*)\, q(\theta \mid \theta^*)\, p_\theta(s \mid y^{\mathrm{o}})}{\pi(\theta)\, p_\theta(s)\, q(\theta^* \mid \theta)\, p_{\theta^*}(s^* \mid y^{\mathrm{o}})}\right\} \mathbb{I}\{d(s^*, s^{\mathrm{o}}) < \varepsilon\}.$$

Same as for *Papers I-II*, the intractable summary densities are approximated using synthetic likelihoods; see the paper for details.

The DC proposal increases the chance that $d(s^*, s^{\mathrm{o}}) < \varepsilon$ even when θ^* is proposed from a diffuse $q(\cdot \mid \cdot)$ and the chain is far from the posterior mode. In some cases this mitigates long initial rejection runs, improves early mixing, and reduces the need to start from large a ε. Empirically, keeping the number of particles for the DC proposal low avoids over-conditioning, which would otherwise yield spurious acceptances.

Figure 7.7 compares 20 independent chains with a small ε and an initial point far from the mode. Standard ABC–MCMC frequently fails to move and produces degenerate marginals. In contrast, ABC-MCMC-DC achieves higher acceptances and mixing, with posterior mass concentrating near the true parameters.

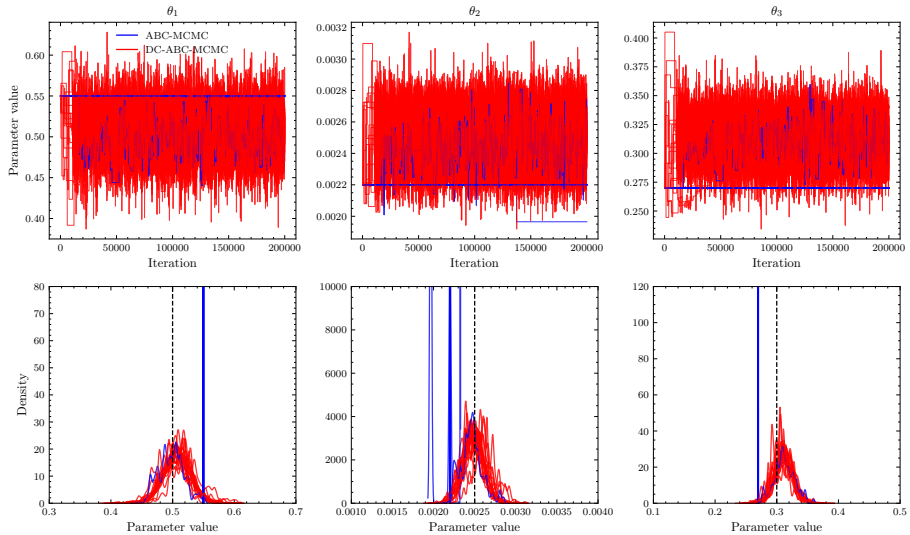

Figure 7.7: Posterior samples from 20 independent ABC-MCMC and ABC-MCMC-DC runs for the stochastic Lotka–Volterra model. Top row: MCMC trace plots for the three model parameters $(\theta_1, \theta_2, \theta_3)$. Bottom row: Kernel density estimates of the posterior marginal distributions. While standard ABC-MCMC chains (blue) frequently remain stuck at their initial values (19 out of 20 runs), resulting in degenerate posteriors, ABC-MCMC-DC method (red) consistently explores the parameter space and concentrates around the true parameter values (black dashed lines).

Bibliography

Ades, M. and Van Leeuwen, P. J. (2015). The equivalent-weights particle filter in a high-dimensional system. *Quarterly Journal of the Royal Meteorological Society*, 141(687):484–503.

Aït-Sahalia, Y. (2002). Maximum likelihood estimation of discretely sampled diffusions: a closed-form approximation approach. *Econometrica*, 70(1):223–262.

Aït-Sahalia, Y. (2008). Closed-form likelihood expansions for multivariate diffusions. *The Annals of Statistics*, 36(2):906–937.

Alfonsi, A. (2005). On the discretization schemes for the cir (and bessel squared) processes. *Monte Carlo Methods and Applications*, 11(4):355–384.

Alfonsi, A. (2013). Strong order one convergence of a drift implicit Euler scheme: Application to the CIR process. *Statistics & Probability Letters*, 83(2):602–607.

Allen, L. J. (2008). An introduction to stochastic epidemic models. In *Mathematical epidemiology*, pages 81–130. Springer.

An, Z., Nott, D. J., and Drovandi, C. (2020). Robust bayesian synthetic likelihood via a semi-parametric approach. *Statistics and Computing*, 30(3):543–557.

Anderson, D. F., Higham, D. J., Leite, S. C., and Williams, R. J. (2019). On constrained Langevin equations and (bio) chemical reaction networks. *Multiscale Modeling & Simulation*, 17(1):1–30.

Andersson, H. and Britton, T. (2012). *Stochastic epidemic models and their statistical analysis*, volume 151. Springer Science & Business Media.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.

Arkin, A., Ross, J., and McAdams, H. H. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected escherichia coli cells. *Genetics*, 149(4):1633–1648.

Bar-Joseph, Z., Gitter, A., and Simon, I. (2012). Studying and modelling dynamic biological processes using time-series gene expression data. *Nature Reviews Genetics*, 13(8):552–564.

Baragatti, M., Grimaud, A., and Pommeret, D. (2013). Likelihood-free parallel tempering. *Statistics and Computing*, 23:535–549.

Bayes, T. (1763). LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S. *Philosophical transactions of the Royal Society of London*, (53):370–418.

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. (2009). Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983–990.

Beaumont, M. A., Zhang, W., and Balding, D. J. (2002). Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035.

Beskos, A., Papaspiliopoulos, O., Roberts, G. O., and Fearnhead, P. (2006). Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion). *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):333–382.

Blum, M. G., Nunes, M. A., Prangle, D., and Sisson, S. A. (2013). A comparative review of dimension reduction methods in approximate bayesian computation. *Statistical Science*, 28(2).

Bortot, P., Coles, S. G., and Sisson, S. A. (2007). Inference for stereological extremes. *Journal of the American Statistical Association*, 102(477):84–92.

Britton, T., Pardoux, E., Ball, F., Laredo, C., Sirl, D., and Tran, V. C. (2019). *Stochastic epidemic models with inference*, volume 2255. Springer.

Bronstein, L., Zechner, C., and Koeppl, H. (2015). Bayesian inference of reaction kinetics from single-cell recordings across a heterogeneous cell population. *Methods*, 85:22–35.

Buckwar, E., Samson, A., Tamborrino, M., and Tubikanec, I. (2022). A splitting method for SDEs with locally Lipschitz drift: Illustration on the Fitzhugh-Nagumo model. *Applied Numerical Mathematics*, 179:191–220.

Buckwar, E., Tamborrino, M., and Tubikanec, I. (2020). Spectral density-based and measure-preserving ABC for partially observed diffusion processes. an illustration on Hamiltonian SDEs. *Statistics and Computing*, 30(3):627–648.

Bunch, P. and Godsill, S. (2012). Improved particle approximations to the joint smoothing distribution using markov chain monte carlo. *IEEE Transactions on Signal Processing*, 61(4):956–963.

Bunch, P. and Godsill, S. (2016). Approximations of the optimal importance density using gaussian particle flow importance sampling. *Journal of the American Statistical Association*, 111(514):748–762.

Cabras, S., Castellanos Nueda, M. E., and Ruli, E. (2015). Approximate bayesian computation by modelling summary statistics in a quasi-likelihood framework. *Bayesian Analysis*, 10(2).

Chan, K. C., Karolyi, G. A., Longstaff, F. A., and Sanders, A. B. (1992). An empirical comparison of alternative models of the short-term interest rate. *The journal of finance*, 47(3):1209–1227.

Chassagneux, J.-F., Jacquier, A., and Mihaylov, I. (2016). An explicit euler scheme with strong rate of convergence for financial SDEs with non-Lipschitz coefficients. *SIAM Journal on Financial Mathematics*, 7(1):993–1021.

Chen, R., Wang, X., and Liu, J. S. (2000). Adaptive joint detection and decoding in flat-fading channels via mixture kalman filtering. *IEEE transactions on Information Theory*, 46(6):2079–2094.

Chen, Y., Zhang, D., Gutmann, M., Courville, A., and Zhu, Z. (2021a). Neural approximate sufficient statistics for implicit models. In *International Conference on Learning Representations*.

Chen, Y., Zhang, D., Gutmann, M. U., Courville, A., and Zhu, Z. (2021b). Neural approximate sufficient statistics for implicit models. In *International Conference on Learning Representations*.

Chen, Z., Raman, B., and Stern, A. (2020). Structure-preserving numerical integrators for Hodgkin–Huxley-type systems. *SIAM Journal on Scientific Computing*, 42(1):B273–B298.

Chopin, N. (2004). Central limit theorem for sequential monte carlo methods and its application to bayesian inference. *The Annals of Statistics*, 32(6):2385–2411.

Chorin, A. J., Morzfeld, M., and Tu, X. (2013). A survey of implicit particle filters for data assimilation. In *State-Space Models: Applications in Economics and Finance*, pages 63–88. Springer.

Chorin, A. J. and Tu, X. (2009). Implicit sampling for particle filters. *Proceedings of the National Academy of Sciences*, 106(41):17249–17254.

Clapp, T. C. and Godsill, S. J. (1999). Fixed-lag smoothing using sequential importance sampling. *Bayesian Statistics 6*, pages 743–752.

Cont, R. and Tankov, P. (2003). *Financial modelling with jump processes*. Chapman and Hall/CRC.

Cox, J. C., Ingersoll, J. E., and Ross, S. A. (1985). A theory of the term structure of interest rates. *Econometrica*, 53(2):385–407.

Cozma, A. and Reisinger, C. (2020). Strong order $1/2$ convergence of full truncation euler approximations to the cox–ingersoll–ross process. *IMA Journal of Numerical Analysis*, 40(1):358–376.

Cranmer, K., Brehmer, J., and Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062.

Dana, S. and Raha, S. (2011). Physically consistent simulation of mesoscale chemical kinetics: The non-negative fis-α method. *Journal of Computational Physics*, 230(24):8813–8834.

Dayan, P. and Abbott, L. F. (2005). *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT press.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436.

Del Moral, P. and Murray, L. M. (2015). Sequential Monte Carlo with highly informative observations. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):969–997.

Delaunoy, A., Hermans, J., Rozet, F., Wehenkel, A., and Louppe, G. (2022). Towards reliable simulation-based inference with balanced neural ratio estimation. *Advances in Neural Information Processing Systems*, 35:20025–20037.

Delyon, B. and Hu, Y. (2006). Simulation of conditioned diffusion and application to parameter estimation. *Stochastic Processes and their Applications*, 116(11):1660–1675.

Dereich, S., Neuenkirch, A., and Szpruch, Ł. (2012). An euler-type method for the strong approximation of the cox–ingersoll–ross process. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 468(2140):1105–1115.

Diggle, P. J. and Gratton, R. J. (1984). Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 46(2):193–212.

Ditlevsen, S. and Samson, A. (2014). Estimation in the partially observed stochastic Morris–Lecar neuronal model with particle filter and stochastic approximation methods. *The Annals of Applied Statistics*, 8(2):674–702.

Ditlevsen, S., Tamborrino, M., and Tubikanec, I. (2023). Network inference in a stochastic multi-population neural mass model via approximate bayesian computation. *arXiv preprint arXiv:2306.15787*.

Douc, R., Garivier, A., Moulines, É., and Olsson, J. (2011). Sequential monte carlo smoothing for general state space hidden markov models. *The Annals of Applied Probability*, 21(6):2109–2145.

Doucet, A., Briers, M., and Sénécal, S. (2006). Efficient block sampling strategies for sequential monte carlo methods. *Journal of Computational and Graphical Statistics*, 15(3):693–711.

Doucet, A., De Freitas, N., Gordon, N. J., et al. (2001). *Sequential Monte Carlo methods in practice*, volume 1. Springer.

Drovandi, C. C., Pettitt, A. N., and McCutchan, R. A. (2016). Exact and approximate Bayesian inference for low integer-valued time series models with intractable likelihoods. *Bayesian Analysis*, 11(2):325–352.

Dubarry, C. and Douc, R. (2011). Particle approximation improvement of the joint smoothing distribution with on-the-fly variance estimation. *arXiv preprint arXiv:1107.5524*.

Durbin, J. and Koopman, S. J. (2012). *Time series analysis by state space methods*. Oxford university press.

Durham, G. B. and Gallant, A. R. (2002). Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes. *Journal of Business & Economic Statistics*, 20(3):297–338.

Durkan, C., Murray, I., and Papamakarios, G. (2020). On contrastive learning for likelihood-free inference. In *International conference on machine learning*, pages 2771–2781. PMLR.

Elerian, O., Chib, S., and Shephard, N. (2001). Likelihood inference for discretely observed nonlinear diffusions. *Econometrica*, 69(4):959–993.

Elowitz, M. B. and Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338.

Elowitz, M. B., Levine, A. J., Siggia, E. D., and Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186.

Eraker, B. (2001). Mcmc analysis of diffusion models with application to finance. *Journal of Business & Economic Statistics*, 19(2):177–191.

Fasiolo, M., Wood, S. N., Hartig, F., and Bravington, M. V. (2018). An extended empirical saddlepoint approximation for intractable likelihoods. *Electronic Journal of Statistics*, 12(1):1544–1578.

Fearnhead, P., Giagos, V., and Sherlock, C. (2014). Inference for reaction networks using the linear noise approximation. *Biometrics*, 70(2):457–466.

Fearnhead, P. and Prangle, D. (2012). Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3):419–474.

Feinberg, A. P. (2014). Epigenetic stochasticity, nuclear structure and cancer: the implications for medicine. *Journal of internal medicine*, 276(1):5–11.

Filippi, S., Barnes, C. P., Cornebise, J., and Stumpf, M. P. (2013). On optimality of kernels for approximate Bayesian computation using sequential Monte Carlo. *Statistical applications in genetics and molecular biology*, 12(1):87–107.

Finkenstädt, B., Woodcock, D. J., Komorowski, M., Harper, C. V., Davis, J. R., White, M. R., and Rand, D. A. (2013). Quantifying intrinsic and extrinsic noise in gene transcription using the linear noise approximation: An application to single cell data. *The Annals of Applied Statistics*, pages 1960–1982.

Florens-Zmirou, D. (1989). Approximate discrete-time schemes for statistics of diffusion processes. *Statistics: A Journal of Theoretical and Applied Statistics*, 20(4):547–557.

Fuchs, C. (2013). *Inference for diffusion processes: with applications in life sciences*. Springer Science & Business Media.

Gelfand, A. E. and Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409.

Gesztelyi, R., Zsuga, J., Kemeny-Beke, A., Varga, B., Juhasz, B., and Tosaki, A. (2012). The hill equation and the origin of quantitative pharmacology. *Archive for history of exact sciences*, 66:427–438.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*, 81(25):2340–2361.

Gillespie, D. T. (1992). A rigorous derivation of the chemical master equation. *Physica A: Statistical Mechanics and its Applications*, 188(1-3):404–425.

Gillespie, D. T., Hellander, A., and Petzold, L. R. (2013). Perspective: Stochastic algorithms for chemical kinetics. *The Journal of chemical physics*, 138(17).

Godsill, S. J., Doucet, A., and West, M. (2004). Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, 99(465):156–168.

Golightly, A. and Wilkinson, D. J. (2008). Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics & Data Analysis*, 52(3):1674–1693.

Golightly, A. and Wilkinson, D. J. (2011). Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface focus*, 1(6):807–820.

Golightly, A. and Wilkinson, D. J. (2015). Bayesian inference for Markov jump processes with informative observations. *Statistical applications in genetics and molecular biology*, 14(2):169–188.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113.

Greenberg, D., Nonnenmacher, M., and Macke, J. (2019). Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, pages 2404–2414. PMLR.

Greenwood, P. E. and Ward, L. M. (2016). *Stochastic neuron models*, volume 1. Springer.

Guarniero, P., Johansen, A. M., and Lee, A. (2017). The iterated auxiliary particle filter. *Journal of the American Statistical Association*, 112(520):1636–1647.

Guo, D., Wang, X., and Chen, R. (2004). Multilevel mixture kalman filter. *EURASIP Journal on Advances in Signal Processing*, 2004(15):413780.

Gupta, P. B., Fillmore, C. M., Jiang, G., Shapira, S. D., Tao, K., Kuperwasser, C., and Lander, E. S. (2011). Stochastic state transitions give rise to phenotypic equilibrium in populations of cancer cells. *Cell*, 146(4):633–644.

Häggström, H., Rodrigues, P. L. C., Oudoumanessah, G., Forbes, F., and Picchini, U. (2024). Fast, accurate and lightweight sequential simulation-based inference using Gaussian locally linear mappings. *Transactions on Machine Learning Research*. https://openreview.net/forum?id=Q0nzpRcwWn.

Hairer, E., Hochbruck, M., Iserles, A., and Lubich, C. (2006). Geometric numerical integration. *Oberwolfach Reports*, 3(1):805–882.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1).

Hefter, M. and Herzwurm, A. (2018). Strong convergence rates for Cox–Ingersoll–Ross processes—full parameter range. *Journal of Mathematical Analysis and Applications*, 459(2):1079–1101.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2):327–343.

Higham, D. J. and Mao, X. (2005). Convergence of Monte Carlo simulations involving the mean-reverting square root process. *Journal of Computational Finance*, 8(3):35–61.

Iacus, S. M. (2008). *Simulation and Inference for Stochastic Differential Equations: With R Examples*. Springer.

Jahnke, T. and Huisinga, W. (2007). Solving the chemical master equation for monomolecular reaction systems analytically. *Journal of mathematical biology*, 54(1):1–26.

Järvenpää, M., Corander, J., and Pesonen, H. (2025). Surrogate-based ABC matches generalized Bayesian inference under specific discrepancy and kernel choices. *arXiv preprint arXiv:2502.11738*.

Jasra, A. (2015). Approximate bayesian computation for a class of time series models. *International Statistical Review*, 83(3):405–435.

Jiang, B., Wu, T.-y., Zheng, C., and Wong, W. H. (2017). Learning summary statistic for approximate Bayesian computation via deep neural network. *Statistica Sinica*, pages 1595–1618.

Kelly, C. (2024). *Computation and Simulation for Finance: An Introduction with Python*. Springer Nature.

Kelly, C., Lord, G., and Maulana, H. (2022). The role of adaptivity in a numerical method for the Cox–Ingersoll–Ross model. *Journal of Computational and Applied Mathematics*, 410:114208.

Kelly, C. and Lord, G. J. (2023). An adaptive splitting method for the Cox-Ingersoll-Ross process. *Applied Numerical Mathematics*, 186:252–273.

Kessler, M. (1997). Estimation of an ergodic diffusion from discrete observations. *Scandinavian Journal of Statistics*, 24(2):211–229.

Kim, S., Shephard, N., and Chib, S. (1998). Stochastic volatility: likelihood inference and comparison with arch models. *The review of economic studies*, 65(3):361–393.

King, A. A., Nguyen, D., and Ionides, E. L. (2016). Statistical inference for partially observed markov processes via the r package pomp. *Journal of Statistical Software*, 69:1–43.

Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25.

Kloeden, P. E. and Platen, E. (1992). *Numerical Solution of Stochastic Differential Equations*. Springer.

Kobayashi, G. and Kozumi, H. (2015). Generalized multiple-point metropolis algorithms for approximate bayesian computation. *Journal of Statistical Computation and Simulation*, 85(4):675–692.

Kok, M., Solin, A., and Schön, T. B. (2024). Rao-blackwellized particle smoothing for simultaneous localization and mapping. *Data-Centric Engineering*, 5:e15.

Komorowski, M., Finkenstädt, B., Harper, C. V., and Rand, D. A. (2009). Bayesian inference of biochemical kinetic parameters using the linear noise approximation. *BMC bioinformatics*, 10:1–10.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288.

Kousathanas, A., Leuenberger, C., Helfer, J., Quinodoz, M., Foll, M., and Wegmann, D. (2016). Likelihood-free inference in high-dimensional models. *Genetics*, 203(2):893–904.

Kramers, H. A. (1940). Brownian motion in a field of force and the diffusion model of chemical reactions. *physica*, 7(4):284–304.

Kypraios, T., Neal, P., and Prangle, D. (2017). A tutorial introduction to bayesian inference for stochastic epidemic models using approximate bayesian computation. *Mathematical biosciences*, 287:42–53.

Lavielle, M. (2014). *Mixed effects models for the population approach: models, tasks, methods and tools*. CRC press.

Lei, J. (2021). *Systems biology*. Springer.

Liepe, J., Kirk, P., Filippi, S., Toni, T., Barnes, C. P., and Stumpf, M. P. (2014). A framework for parameter estimation and model selection from experimental data in systems biology using approximate bayesian computation. *Nature protocols*, 9(2):439–456.

Lin, M., Chen, R., and Liu, J. S. (2013). Lookahead strategies for sequential Monte Carlo. *Statistical Science*, 28(1):69–94.

Lin, M., Chen, R., and Mykland, P. (2010). On generating monte carlo samples of continuous diffusion bridges. *Journal of the American Statistical Association*, 105(490):820–838.

Locke, J. C. and Elowitz, M. B. (2009). Using movies to analyse gene circuit dynamics in single cells. *Nature Reviews Microbiology*, 7(5):383–392.

Lord, R., Koekkoek, R., and Dijk, D. V. (2010). A comparison of biased simulation schemes for stochastic volatility models. *Quantitative Finance*, 10(2):177–194.

Loskot, P., Atitey, K., and Mihaylova, L. (2019). Comprehensive review of models and methods for inferences in bio-chemical reaction networks. *Frontiers in genetics*, 10:549.

Lueckmann, J.-M., Boelts, J., Greenberg, D., Goncalves, P., and Macke, J. (2021). Benchmarking simulation-based inference. In *International conference on artificial intelligence and statistics*, pages 343–351. PMLR.

Malham, S. J. and Wiese, A. (2014). Efficient almost-exact lévy area sampling. *Statistics & Probability Letters*, 88:50–55.

Mao, X. (2006). *Stochastic Differential Equations with Markovian Switching*. Imperial College Press.

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. (2003). Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328.

Martin, G. M., Frazier, D. T., and Robert, C. P. (2024). Approximating Bayes in the 21st century. *Statistical Science*, 39(1):20–45.

Martin, G. M., McCabe, B. P., Frazier, D. T., Maneesoonthorn, W., and Robert, C. P. (2019). Auxiliary likelihood-based approximate bayesian computation in state space models. *Journal of Computational and Graphical Statistics*, 28(3):508–522.

Matsubara, T., Knoblauch, J., Briol, F.-X., and Oates, C. J. (2022). Robust generalised bayesian inference for intractable likelihoods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(3):997–1022.

Maybank, P., Bojak, I., and Everitt, R. G. (2017). Fast approximate bayesian inference for stable differential equation models. *arXiv preprint arXiv:1706.00689*.

McAdams, H. H. and Arkin, A. (1997). Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences*, 94(3):814–819.

McLachlan, R. I. and Quispel, G. R. W. (2002). Splitting methods. *Acta Numerica*, 11:341–434.

Meeds, E., Leenders, R., and Welling, M. (2015). Hamiltonian abc. *arXiv preprint arXiv:1503.01916*.

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

Michaelis, L. and Menten, M. L. (1913). Die kinetik der invertinwirkung. *Biochem. Z.*, 49:333–369.

Miller, B. K., Cole, A., Forré, P., Louppe, G., and Weniger, C. (2021). Truncated marginal neural ratio estimation. *Advances in Neural Information Processing Systems*, 34:129–143.

Møller, J. K. and Madsen, H. (2010). From state dependent diffusion to constant diffusion in stochastic differential equations by the lamperti transform. Technical report, Technical University of Denmark (DTU) Informatics, Lyngby, Denmark.

Morzfeld, M., Tu, X., Atkins, E., and Chorin, A. J. (2012). A random map implementation of implicit filters. *Journal of Computational Physics*, 231(4):2049–2066.

Moyal, J. E. (1949). Stochastic processes and statistical physics. *Journal of the Royal Statistical Society. Series B (Methodological)*, 11(2):150–210.

Ong, V. M., Nott, D. J., Tran, M.-N., Sisson, S. A., and Drovandi, C. C. (2018). Variational Bayes with synthetic likelihood. *Statistics and Computing*, 28:971–988.

Ozaki, T. (1992). A bridge between nonlinear time series models and nonlinear stochastic dynamical systems: a local linearization approach. *Statistica Sinica*, pages 113–135.

Pacchiardi, L., Khoo, S., and Dutta, R. (2024). Generalized Bayesian likelihood-free inference. *Electronic Journal of Statistics*, 18(2):3628–3686.

Papadakis, N., Mémin, É., Cuzol, A., and Gengembre, N. (2010). Data assimilation with the weighted ensemble kalman filter. *Tellus A: Dynamic Meteorology and Oceanography*, 62(5):673–697.

Papamakarios, G. and Murray, I. (2016). Fast ε-free inference of simulation models with Bayesian conditional density estimation. *Advances in Neural Information Processing Systems*, 29.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680.

Papamakarios, G., Sterratt, D., and Murray, I. (2019). Sequential Neural Likelihood: Fast Likelihood-free Inference with Autoregressive Flows. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89, pages 837–848. PMLR.

Park, J. and Ionides, E. L. (2020). Inference on high-dimensional implicit dynamic models using a guided intermediate resampling filter. *Statistics and Computing*, 30(5):1497–1522.

Paulsson, J., Berg, O. G., and Ehrenberg, M. (2000). Stochastic focusing: fluctuation-enhanced sensitivity of intracellular regulation. *Proceedings of the National Academy of Sciences*, 97(13):7148–7153.

Pedersen, A. R. (1995). A new approach to maximum likelihood estimation for stochastic differential equations based on discrete observations. *Scandinavian journal of statistics*, pages 55–71.

Pham, H. and Quenez, M.-C. (2001). Optimal portfolio in partially observed stochastic volatility models. *Annals of Applied Probability*, pages 210–238.

Picchini, U. (2014). Inference for SDE models via approximate Bayesian computation. *Journal of Computational and Graphical Statistics*, 23(4):1080–1100.

Picchini, U. and Forman, J. L. (2016). Accelerating inference for diffusions observed with measurement error and large sample sizes using approximate Bayesian computation. *Journal of Statistical Computation and Simulation*, 86(1):195–213.

Picchini, U., Simola, U., and Corander, J. (2023). Sequentially guided MCMC proposals for synthetic likelihoods and correlated synthetic likelihoods. *Bayesian Analysis*, 18(4):1099–1129.

Picchini, U. and Tamborrino, M. (2024). Guided sequential ABC schemes for intractable Bayesian models. *Bayesian Analysis*, 1(1):1–32.

Pilipovic, P., Samson, A., and Ditlevsen, S. (2024). Parameter estimation in nonlinear multivariate stochastic differential equations based on splitting schemes. *The Annals of Statistics*, 52(2):842–867.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599.

Prangle, D. (2017). Adapting the ABC distance function. *Bayesian Analysis*, 12(1).

Prangle, D. (2018). Summary statistics. In *Handbook of approximate Bayesian computation*, pages 125–152. Chapman and Hall/CRC.

Price, L. F., Drovandi, C. C., Lee, A., and Nott, D. J. (2018). Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27(1):1–11.

Priddle, J. W., Sisson, S. A., Frazier, D. T., Turner, I., and Drovandi, C. (2022). Efficient bayesian synthetic likelihood with whitening transformations. *Journal of Computational and Graphical Statistics*, 31(1):50–63.

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., and Feldman, M. W. (1999). Population growth of human y chromosomes: a study of y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798.

Radev, S. T., Schmitt, M., Pratz, V., Picchini, U., Koethe, U., and Buerkner, P. (2023). JANA: Jointly amortized neural approximation of complex Bayesian models. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 1695–1706. PMLR.

Ratmann, O., Andrieu, C., Wiuf, C., and Richardson, S. (2009). Model criticism based on likelihood-free inference, with an application to protein network evolution. *Proceedings of the National Academy of Sciences*, 106(26):10576–10581.

Ratmann, O., Jørgensen, O., Hinkley, T., Stumpf, M., Richardson, S., and Wiuf, C. (2007). Using likelihood-free inference to compare evolutionary dynamics of the protein networks of h. pylori and p. falciparum. *PLoS Computational Biology*, 3(11):e230.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR.

Roberts, G. O. and Stramer, O. (2001). On inference for partially observed nonlinear diffusion models using the metropolis–hastings algorithm. *Biometrika*, 88(3):603–621.

Rosenbluth, M. N. and Rosenbluth, A. W. (1955). Monte carlo calculation of the average extension of molecular chains. *The Journal of chemical physics*, 23(2):356–359.

Rubin, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, pages 1151–1172.

Rydén, T. and Wiktorsson, M. (2001). On the simulation of iterated ito integrals. *Stochastic processes and their applications*, 91(1):151–168.

Samson, A., Tamborrino, M., and Tubikanec, I. (2025). Inference for the stochastic fitzhugh-nagumo model from real action potential data via approximate bayesian computation. *Computational Statistics & Data Analysis*, 204:108095.

Särkkä, S. and Svensson, L. (2023). *Bayesian filtering and smoothing*, volume 17. Cambridge university press.

Schnoerr, D., Sanguinetti, G., and Grima, R. (2014). The complex chemical Langevin equation. *The Journal of chemical physics*, 141(2).

Schnoerr, D., Sanguinetti, G., and Grima, R. (2017). Approximation and inference methods for stochastic biochemical kinetics—a tutorial review. *Journal of Physics A: Mathematical and Theoretical*, 50(9):093001.

Shahrezaei, V. and Swain, P. S. (2008). Analytical distributions for stochastic gene expression. *Proceedings of the National Academy of Sciences*, 105(45):17256–17261.

Shoji, I. and Ozaki, T. (1998). Estimation for nonlinear stochastic differential equations by a local linearization method. *Stochastic Analysis and Applications*, 16(4):733–752.

Shreve, S. E. et al. (2004). *Stochastic calculus for finance II: Continuous-time models*, volume 11. Springer.

Simola, U., Cisewski-Kehe, J., Gutmann, M. U., and Corander, J. (2021). Adaptive approximate bayesian computation tolerance selection. *Bayesian analysis*, 16(2):397–423.

Sisson, S. A. and Fan, Y. (2011). Likelihood-free mcmc. *Handbook of Markov Chain Monte Carlo*, pages 313–335.

Sisson, S. A., Fan, Y., and Beaumont, M. (2018). *Handbook of approximate Bayesian computation*. CRC Press.

Stathopoulos, V. and Girolami, M. A. (2013). Markov chain Monte Carlo inference for Markov jump processes via the linear noise approximation. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110541.

Szpruch, L. and Higham, D. J. (2010). Comparing hitting time behavior of Markov jump processes and their diffusion approximations. *Multiscale Modeling & Simulation*, 8(2):605–621.

Taghavi, E., Lindsten, F., Svensson, L., and Schön, T. B. (2013). Adaptive stopping for fast particle smoothing. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6293–6297. IEEE.

Tancredi, A. (2019). Approximate bayesian inference for discretely observed continuous-time multi-state models. *Biometrics*, 75(3):966–977.

Tavaré, S. (2018). On the history of ABC. In *Handbook of Approximate Bayesian Computation*, pages 55–69. Chapman and Hall/CRC.

Tavaré, S., Balding, D. J., Griffiths, R. C., and Donnelly, P. (1997). Inferring coalescence times from dna sequence data. *Genetics*, 145(2):505–518.

Tian, T. and Burrage, K. (2006). Stochastic models for regulatory networks of the genetic toggle switch. *Proceedings of the national Academy of Sciences*, 103(22):8372–8377.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. (2009). Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202.

Tukey, J. W. (1962). The future of data analysis. *Ann. Math. Statist.*, 33(1).

van der Meulen, F. and Schauer, M. (2017). Bayesian estimation of discretely observed multi-dimensional diffusion processes using guided proposals. *Electronic Journal of Statistics*, 11:2358–2396.

Van Kampen, N. G. (1992). *Stochastic processes in physics and chemistry*, volume 1. Elsevier.

Van Leeuwen, P. J. (2010). Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society*, 136(653):1991–1999.

Vihola, M. and Franks, J. (2020). On the use of approximate bayesian computation markov chain monte carlo with inflated tolerance and post-correction. *Biometrika*, 107(2):381–395.

Wang, X., Chen, R., and Guo, D. (2002). Delayed-pilot sampling for mixture kalman filter with application in fading channels. *IEEE Transactions on Signal Processing*, 50(2):241–254.

Wang, X., Kelly, R. P., Jenner, A. L., Warne, D. J., and Drovandi, C. (2024). A comprehensive guide to simulation-based inference in computational biology. *arXiv preprint arXiv:2409.19675*.

Weerasinghe, C., Frazier, D. T., Loaiza-Maya, R., and Drovandi, C. (2025). Robustifying approximate bayesian computation. *arXiv preprint arXiv:2504.04733*.

Wegmann, D., Leuenberger, C., and Excoffier, L. (2009). Efficient approximate bayesian computation coupled with markov chain monte carlo without likelihood. *Genetics*, 182(4):1207–1218.

Wiktorsson, M. (2001). Joint characteristic function and simultaneous simulation of iterated itô integrals for multiple independent brownian motions. *The Annals of Applied Probability*, 11(2):470–487.

Wilkie, J. and Wong, Y. M. (2008). Positivity preserving chemical Langevin equations. *Chemical Physics*, 353(1-3):132–138.

Wilkinson, D. J. (2018). *Stochastic modelling for systems biology*. Chapman and Hall/CRC, 3rd edition.

Wilkinson, R. D. (2013). Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Statistical applications in genetics and molecular biology*, 12(2):129–141.

Wiqvist, S., Frellsen, J., and Picchini, U. (2021). Sequential neural posterior and likelihood approximation. *arXiv preprint arXiv:2102.06522*.

Wiqvist, S., Mattei, P.-A., Picchini, U., and Frellsen, J. (2019). Partially exchangeable networks and architectures for learning summary statistics in approximate Bayesian computation. In *International Conference on Machine Learning*, pages 6798–6807. PMLR.

Wood, S. N. (2010). Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104.

Young, J. W., Locke, J. C., Altinok, A., Rosenfeld, N., Bacarian, T., Swain, P. S., Mjolsness, E., and Elowitz, M. B. (2012). Measuring single-cell gene expression dynamics in bacteria using fluorescence time-lapse microscopy. *Nature protocols*, 7(1):80–88.

Zammit-Mangion, A., Sainsbury-Dale, M., and Huser, R. (2024). Neural methods for amortized inference. *Annual Review of Statistics and Its Application*, 12.

Zhang, J. L. and Liu, J. S. (2002). A new sequential importance sampling method and its application to the two-dimensional hydrophobic–hydrophilic model. *The Journal of Chemical Physics*, 117(7):3492–3498.

Zhou, S., Lo, W.-C., Suhalim, J. L., Digman, M. A., Gratton, E., Nie, Q., and Lander, A. D. (2012). Free extracellular diffusion creates the dpp morphogen gradient of the drosophila wing disc. *Current Biology*, 22(8):668–675.

Øksendal, B. (2013). *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media.