Exact Bayesian Inference for Diffusion-based Models

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Statistics (Research)

by Timothée Stumpf-Fétizon
Department of Statistics, University of Warwick

December 2022

Contents

List of Figures

Acknowledgements

I am grateful to my academic supervisors Krzysztof Łatuszyński and Gareth Roberts, for their imaginative and confident guidance.

I would also like to to recognize Omiros Papaspiliopoulos for setting me on this journey, and my fellow travelers Ian Hamilton, Santhosh Narayanan and Simon Gansinger, for their good company in times of upheaval.

Declaration of Authenticity

I hereby declare that this thesis is the result of my own work and research, except where otherwise indicated. This thesis has not been submitted for examination to any institution other than the University of Warwick.

Abstract

We develop methods to carry out Bayesian inference for *diffusion*-based continuous-time models, formulated as *stochastic differential equations* (SDEs). The transition density implied by such SDEs is intractable, which complicates likelihood-based inference from discrete observations. In spite of this obstacle, we seek methods that are *exact* in the sense that they target the correct posterior distribution, in contrast to prevailing discretization approaches.

We begin by discussing the main approaches to likelihood-based inference under intractability, and their application to diffusion-based models. This discussion is followed by a presentation of the fundamental inference algorithms for ordinary *Itō diffusion* inference, of computational difficulties they meet in practice, and of recent improvements motivated by our research on more complex diffusion-based models. These include *Markov switching diffusions* and *stochastic volatility models*, where a latent continuous-time process modifies the dynamics of an observable diffusion process. We follow up by developing *Markov chain Monte Carlo* (MCMC) and *Monte Carlo Expectation Maximization* (MCEM) inference algorithms for the more complex settings, and evaluate them systematically. We close with a discussion of practical hurdles to adoption of exact algorithms, and propose solutions to overcome those hurdles.

1 Introduction

> All will be revealed, *retrospectively.*

Mathematical models for data that evolves randomly and continuously in time have become increasingly popular across different scientific disciplines since the 1940s, when the foundations of *Stochastic calculus* were laid for the analysis of such models. Collectively, these models are known as *diffusion processes*. In many cases, practical diffusion models arise as the limiting case of models originally formulated in discrete time. Examples include models in mathematical finance [19], genetics [113, 86], physics [121] and chemistry [50].

We will be interested in models that can be described, at least for some finite time, as *stochastic differential equations* (SDEs) of form

$$\mathrm{d}V_t = \mu(V_t)\,\mathrm{d}t + \sigma(V_t)\,\mathrm{d}W_t, \qquad (t \in [0, \omega]) \tag{1.1}$$

where W is a Brownian motion on the canonical filtered probability space

$$(\Omega = \mathcal{C}[0,\omega],\ \mathcal{F} = \mathcal{B}(\mathcal{C}[0,\omega]),\ \{\mathcal{F}_t : t \in [0,\omega]\},\ \mathbb{W}), \tag{1.2}$$

with $\mathcal{C}[0,\omega]$ being the set of continuous functions on $[0,\omega]$, $\{\mathcal{F}_t : t \in [0,\omega]\}$ the natural filtration of W, and $\mathbb{W}$ the *Wiener measure* induced by W. V_t takes values in an interval $\mathcal{V} \subseteq \mathbf{R}$ for some *instantaneous drift function* $\mu : \mathcal{V} \to \mathbf{R}$ and *instantaneous volatility function* $\sigma : \mathcal{V} \to [0, \infty)$.

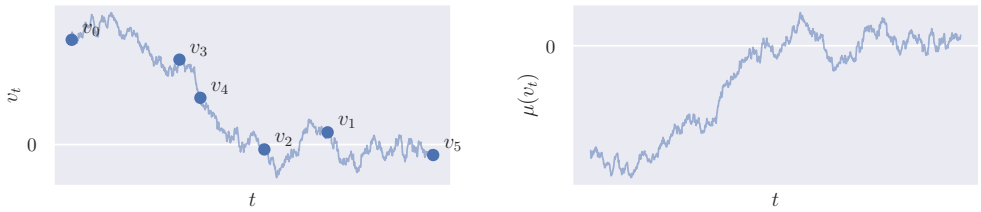


Figure 1.1: Example trajectory from a mean-reverting diffusion process. The process is driven down from its starting value by a negative drift which subsides once it reaches a smaller value. We discretely observe the process at the labeled locations.

An SDE is a concise representation of the stochastic integral equation

$$V_t - V_0 = \int_0^t \mu(V_u)\,\mathrm{d}u + \int_0^t \sigma(V_u)\,\mathrm{d}W_u, \qquad (t \in [0, \omega]) \tag{1.3}$$

but it also provides intuition on why such models are attractive from an applied perspective. A person wishing to describe a stochastic process in continuous time will often find it easiest to reason about it based on its local behavior, with μ describing the strength and direction of deterministic forces affecting the process, and σ describing the strength of the randomness, depending on where the process happens to be. In particular, μ can be used to describe attractive or repulsive regions of $\mathcal{V}$, and induce mean reversion. To strip away the complexities of the continuous time formalism, we can also reason about SDEs in terms of the discrete approximation

$$V_t - V_0 = \epsilon\mu(V_0) + \sqrt{t}\sigma(V_0)(W_t - W_0), \qquad (t \in [0, \omega]) \tag{1.4}$$

known as the *Euler-Maruyama approximation* to the SDE. The SDE may then be seen as the limit of the approximation as $t \to 0$. It is clear that for a fixed step size t, the discrete equation describes a Markov process with a transition density $\pi(v_t|v_0)$. Analogously, under appropriate regularity conditions on μ and σ [96], an SDE defines a continuous-time Markov process with a transition density $\pi(v_t|v_0)$ for any positive t. We call such a process a *diffusion process*. We will take a particular interest in *Markov switching diffusion models*, where we allow for SDEs of form

$$\mathrm{d}V_t = \mu(V_t, Y_t)\,\mathrm{d}t + \sigma(V_t, Y_t)\,\mathrm{d}W_t, \tag{1.5}$$

where Y_t takes a discrete number of values. This allows for exogenous, discrete breaks in μ and σ and therefore in the behavior of the process.

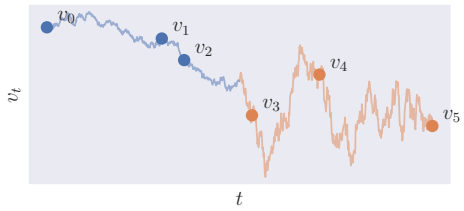


Figure 1.2: Example trajectory from Markov switching diffusion process. The process switches to a different regime at the halfway mark, upon which it exhibits larger volatility. This change in regime could be identified from the observations at the labeled locations.

In scientific practice, both drift and volatility are often parameterized as μ_θ and σ_θ by a vector of free parameters θ. This naturally raises the question of how to infer those parameters based on discretely observed data. Standard statistical methodology relies on evaluating the *likelihood* of the observed data under a given model. Within the Bayesian

framework, given a discrete set of observation times s and the corresponding observation $\{V_s = v_s\}$, the *posterior density* over θ is given by

$$\pi(\theta|v_s) = \frac{\overbrace{\pi(\theta)}^{\text{prior}} \overbrace{\prod_{(\dot{s}\sim\ddot{s})\in s} \pi(v_{\ddot{s}}|v_{\dot{s}}, \theta)}^{\text{likelihood}}}{\underbrace{\int \pi(\theta') \prod_{(\dot{s}\sim\ddot{s})\in s} \pi(v_{\ddot{s}}|v_{\dot{s}}, \theta') \, \mathrm{d}\theta'}_{\text{prior predictive}}}, \tag{1.6}$$

where the notation $(\dot{s} \sim \ddot{s}) \in s$ is to be understood as iterating over neighbouring ordered pairs in s, the transition density $\pi(v_{\ddot{s}}|v_{\dot{s}}, \theta)$ is now parameterized by θ, and the factorization of the likelihood is due to the Markov property. The *prior predictive density* (also known as marginal likelihood or evidence) typically follows from an intractable integration problem with respect to θ, and therefore most of the field of *Bayesian computation* seeks to approximate $\pi(\theta|v_s)$ without requiring evaluations of the denominator. We say that a computational method is *exact* or *computationally consistent* if that method provides arbitrarily accurate answers as its computational budget increases. Discretizing the model as in (1.4) is not an exact method for any given ϵ since the discretization introduces a bias into the posterior distribution. The diffusion context poses a particular challenge to the application of exact computational frameworks because for all but a small class of diffusion processes, the transition density $\pi(v_{\ddot{s}}|v_{\dot{s}}, \theta)$ is implied by the SDE, but not explicitly available for evaluation. Accordingly, if we wish to estimate the parameters of such an intractable diffusion model, we have the option of either extending exact methods appropriately, or of renouncing exact methods entirely, and resorting to discretization schemes such as (1.4). We see two main reasons for persisting with exact methods:

- The magnitude of the bias in approximate methods is difficult to predict. Accordingly, practitioners may have to run an inference algorithm with various choices of step size ϵ, hoping that the inference eventually stabilizes.
- Assuming that both step size and Monte Carlo sample size are increased at an optimal rate, the best approximate methods have an asymptotic mean square error of more than $\mathcal{O}\left[1/\sqrt{\text{computational budget}}\right]$, which is the rate of exact Monte Carlo methods under favorable conditions.

To the extent that the exact methods studied here may be thought of as adaptive, randomized discretization methods, they can also be more parsimonious than methods with fixed step sizes, allocating effort to the most difficult part of the diffusion. Indeed, for many diffusion models, drift or volatility are highly variable on some parts of the trajectory, increasing the error of discretization methods, and less variable in others, where discretization performs better. The exact methods shift their effort to those highly variable parts of the trajectory, while often allocating no additional effort to the stable parts. The downside of the exact methods is that they build on theoretical aspects of diffusions that would not be essential to addressing the question of inference diligently if the transition density could be evaluated. Conversely, static discretization methods

are conceptually simple, and require only limited knowledge of the theory of diffusions to be developed and deployed. This thesis puts exact methods front and center, though it includes approximate methods for the purpose of benchmarking the exact ones, or as a stepping stone to exact methods.

1.1 Summary of Thesis

Overall, the contributions of the thesis fall along two lines:

- On the one hand, the development of more practical, reliable and scalable algorithms for basic Itō diffusion inference by building on novel and existing methodology. In particular, the thesis contains the first algorithms for exact Bayesian inference in Itō diffusions that do not require user-specified tunings, and only a few lines of code for customization by the user.
- On the other hand, the development of algorithms for broader and more challenging classes of diffusion models by leveraging our refinements for Itō diffusions. In particular, the thesis contains the first practical algorithms for exact Bayesian inference in Markov switching diffusions and an algorithm for Bayesian inference in fully continuous time stochastic volatility models.

With some exceptions, Chapters 2-4 of the thesis are mostly background material, while the fifth has original elements that factor into the development of the fully original Chapters 6-8. The structure of the thesis is as follows:

- The second chapter introduces generic approaches to exact inference when the likelihood is intractable, in particular the approach of *data augmentation*, to obtain an extended, tractable model.
- The third chapter provides the necessary theoretical background on diffusion models to apply the intractable likelihood inference principles to diffusions.
- The fourth chapter provides methods that address the infinite dimensional nature of diffusion paths, based on the principle of *retrospective simulation*. These include the family of *exact (simulation) algorithms*.
- The fifth chapter applies the background material to develop model-agnostic, self-tuned and robust algorithms for exact Bayesian diffusion inference.
- The sixth chapter takes insights from the ordinary diffusion case and extends them to build exact Bayesian inference algorithms for Markov switching diffusions.
- The seventh chapter presents an approximate representation of *Stochastic volatility* models as Markov switching diffusions, and leverages the methodology of the sixth chapter to construct a Bayesian inference algorithm.

- The eight chapter addresses some previously challenging aspects of implementing exact simulation and inference algorithms, and shows how to automate them.

1.2 Conventions

Throughout the thesis, these conventions are adhered to:

- Random variables and events are written in uppercase (A), random *variates* and most other deterministic objects in lowercase (a).
- Measures are written in blackboard bold ($\mathbb{A}$). $\mathbb{A}|C$ is the measure $\mathbb{A}$ conditioned on the event C. For a random variable B, $\mathbb{A}|b$ is a shorthand for $\mathbb{A}|\{B = b\}$. For better readability, we write $\mathbb{A}_{|b}[D]$ to evaluate $\mathbb{A}|b$ at the event D.
- For a discrete random variable A, $\pi(a)$ denotes the probability mass function of A at the variate a, i.e. $\Pr[A = a]$. For a continuous random variable B, $\pi(b)$ denotes the probability density function of B at the variate b.
- For a set or function a, $\check{a} = \inf a$ and $\hat{a} = \sup a$.
- For a set or function a, $a^{\downarrow} \leq \check{a}$ and $a^{\uparrow} \geq \hat{a}$ are lower and upper bounds on a, respectively.
- Similarly, when indexing a set with an integer index k, e.g. $a = \left\{a_k : k = 1, ..., \hat{k}\right\}$, $\hat{k}$ refers to the cardinality $|a|$. When using a continuous index t, $\hat{t}$ refers to the upper end of the indexing interval.
- For a time-indexed set $a = \{a_t : t \in [0, \infty)\}$ and a set of times $b \subset [0, \infty)$, a_b denotes $\{a_t : t \in b\}$.
- For three sets a, b, c such that $a \subseteq b$ and a function $f : b \to c$, $f(a)$ denotes the application of f to each element in a.
- As mentioned above, $(\dot{a} \sim \ddot{a}) \in a$ denotes an iteration over the neighbouring pairs in the ordered set a.
- $\mathrm{N}[a; b, c]$ is the Gaussian density function with variate a, mean b and variance c.
- For continuous paths z and intervals $a \in \mathbf{R}$, we may abuse notation by understanding $z \subseteq a$ to mean $\bigcap_t \{z_t \in a\}$, i.e. z is bounded within a.

1.3 Symbols

Though many symbols do not have fixed definitions across different chapters, the following definitions apply throughout the thesis:

- t is a time index. ω is the "end of continuous time".
- V is a generic diffusion process and v a variate of the process. μ is the instantaneous drift, σ the instantaneous volatility. Its state space is denoted $\mathcal{V}$.
- $X = \eta(V)$ is the reduced or *Lamperti transformed* analogue of V and $x = \eta(v)$ is a variate of the process. δ is the instantaneous drift. Its state space is denoted $\mathcal{X}$.
- W is the *Wiener process* and w is a variate of the process.
- s is a set of observation times.
- Θ is a vector of unknown parameters, θ is a variate thereof. $\mathcal{T}$ denotes the corresponding parameter space.
- Y is a discrete space, continuous time *Markov jump process* with state space $\mathcal{Y}$.
- For a set a, $\mathcal{B}(a)$ is the Borel σ-algebra generated by all the open sets in a.

2 MCMC Methods for Intractable Likelihood Models

We proceed with a more careful investigation of how the intractable transition density of diffusion models complicates Bayesian inference, and explore strategies to overcome that difficulty. The discussion applies more broadly to complex, intractable models, and we therefore adopt the generic setting where we infer a vector of unknowns Θ with prior distribution $\pi(\theta)$ on $\mathcal{T} \subseteq \mathbf{R}^d$ from an observational event $\{A = a\}$ with model $\pi(a|\theta)$, yielding the posterior density

$$\pi(\theta|a) = \frac{\pi(a|\theta)\pi(\theta)}{\underbrace{\int \pi(a|\theta')\pi(\theta')\,\mathrm{d}\theta'}_{\pi(a)}} \tag{2.1}$$

with respect to the Lebesgue measure. In most cases, the integral $\pi(a)$ in the denominator is not analytically solvable, precluding the exact evaluation of posterior summaries, generally expressed as

$$\mathrm{E}\left[f(\Theta)|a\right] = \int f(\theta)\frac{\pi(a|\theta)\pi(\theta)}{\int \pi(a|\theta')\pi(\theta')\,\mathrm{d}\theta'}\,\mathrm{d}\theta, \tag{2.2}$$

for integrable test functions f with respect to the posterior measure. This basic difficulty motivates the field of *Bayesian computation*, which seeks to reliably approximate the intractable integrals. The essential insights from the theory of numerical integration carry over to this setting: quadrature methods work well for low-dimensional unknowns, but suffer from the *curse of dimensionality*, an exponential slowdown with the dimension of Θ. The main approach to Bayesian computation is the Monte Carlo method, where samples $(\theta^{(1)}, \ldots, \theta^{(\hat{k})})$ are generated from the posterior, and $\mathrm{E}\left[f(\Theta)|a\right]$ is approximated by its unbiased estimator $\hat{k}^{-1}\sum_{k=1}^{\hat{k}} f(\theta^{(k)})$. The variance of the estimator is in principle invariant in the dimension of Θ, which explains the popularity of Monte Carlo methods in solving high-dimensional Bayesian inference problems. The reader may consult [104] for the bird's-eye view on Monte Carlo methods.

In this thesis, we strongly emphasize *Markov Chain Monte Carlo* (MCMC) algorithms, which are briefly introduced in Section 2.1. MCMC substitutes the *global* problem of normalization - how likely is an outcome relative to the *entirety* of outcomes - for a *local* problem which merely requires knowledge of the relative likelihood of *any two* outcomes. It offers great flexibility in exploiting this locality, and in fortuitous cases

delivers on the CLT's "promise" to lift the curse of dimensionality. The flexibility of the framework is both a feature and a bug - it allows for various strategies to adapt to individual problems, but it will often fail unless these degrees of freedom are exploited well. Decades of practice and theory have resulted in much knowledge of how to best adapt the framework to difficult sampling problems, and adopting the framework allows us to benefit from that knowledge in addressing our specific problem. The purpose of this chapter is to provide an overview of the possibilities that MCMC offers, and to justify the higher-order choices that we make in the light of those possibilities.

The introduction serves to emphasize the problems that arise in the MCMC setting when the transition density is unavailable. Sections 2.2 and 2.3 introduce workarounds to the intractable likelihood problem. Section 2.4 discusses how to assess the computational efficiency of such workarounds. Finally, Section 2.5 draws conclusions for the specific diffusion inference problem and sets out the strategy going forward.

2.1 A Very Short Introduction to Markov Chain Monte Carlo

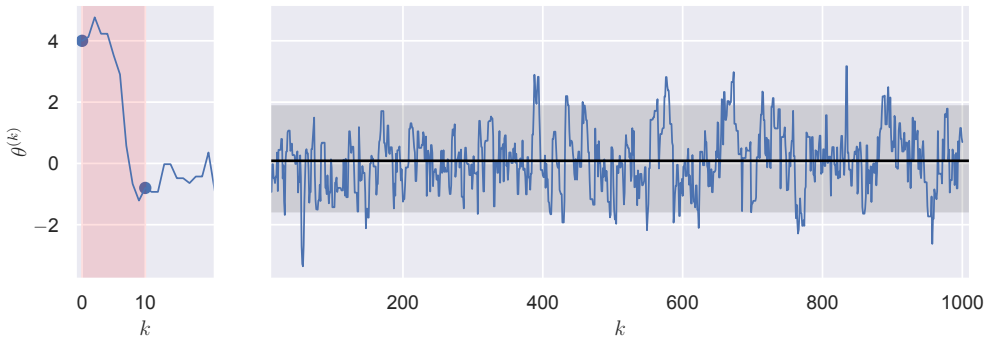


Figure 2.1: Variate of a Markov chain with stationary distribution $\mathrm{N}[0, 1]$, started from $\theta^{(0)} = 4$. The black line shows the ergodic average and the dark shaded area the empirical 90% interval which estimates the 90% credible interval. The first 10 iterations in the red shaded area are not included in the computation of the statistics to reduce the bias from the initialization (*burn-in*).

Since direct sampling methods are usually only available for analytically tractable posteriors, the development of indirect posterior sampling methods is the core challenge of Bayesian computation. Nonetheless, the definitions and methods of this chapter can be applied regardless of the specific objective and notation of Bayesian statistics, so we will consider the general task of sampling from any distribution $\pi(\theta)$ on $\mathcal{T}$. One approach to indirect sampling, known as Markov Chain Monte Carlo, is to construct a tractable Markov kernel $P(\theta, \mathrm{d}\theta^\dagger)$ with limiting distribution $\pi(\theta)$. The Markov chain $(\theta^{(1)}, \ldots, \theta^{(\hat{k})})$ is then simulated for $\hat{k}$ steps, an example of which is shown in Figure 2.1,

and the expectation w.r.t. $\pi(\theta)$ is approximated by the *ergodic average*

$$\frac{1}{\hat{k}} \sum_{k=1}^{\hat{k}} f(\theta^{(k)}). \qquad (\mathrm{E}\left[f(\Theta)\right] < \infty) \tag{2.3}$$

There are various requirements for a "good" Markov chain - simulation of $P(\theta, \mathrm{d}\theta^\dagger)$ should be easy, but it should also quickly "forget its past". Equivalently, the k-transition kernel $P^k(\theta, \vartheta)$ should quickly approach $\Pr\left[\Theta \in \vartheta\right]$ for any starting value θ and any measurable set $\vartheta \in \mathcal{B}(\mathcal{T})$. To make this notion precise, we may apply various concepts from the theory of Markov chains, see e.g. [91]. One popular metric of proximity between the k-kernel and the stationary distribution is the *total variation distance*, which in this instance is given by

$$\left|P^k(\theta, \cdot) - \pi\right|_{\mathrm{TV}} = \sup_{\vartheta \in \mathcal{B}(\mathcal{T})} \left|P^k(\theta, \vartheta) - \pi(\vartheta)\right|, \tag{2.4}$$

i.e. the distance is low if the k-step Markov chain and the stationary distribution assigns similar probabilities to all Borel sets. We call the kernel *geometrically ergodic* if this distance eventually decays exponentially fast, i.e.

$$\left|P^k(\theta, \cdot) - \pi\right|_{\mathrm{TV}} = \mathcal{O}(\gamma^{-k}). \qquad (\gamma \in (0,1), \quad k \geq c, \quad c < \infty) \tag{2.5}$$

If a geometric bound can be obtained for $\sup_{\theta \in \mathcal{T}} \left|P^k(\theta, \cdot) - \pi\right|_{\mathrm{TV}}$, the Markov chain is said to be *uniformly ergodic*. If on top of being geometrically ergodic P is *reversible* or in *detailed balance*, i.e.

$$\pi(\mathrm{d}\theta) P(\theta, \mathrm{d}\theta^\dagger) = \pi(\mathrm{d}\theta^\dagger) P(\theta^\dagger, \mathrm{d}\theta), \qquad (\theta, \theta^\dagger \in \mathcal{T}) \tag{2.6}$$

then the ergodic average satisfies the *Markov chain central limit theorem* (MCCLT).

Theorem 1 (Markov chain central limit theorem (e.g. [91]))**.** *Let P be a geometrically ergodic Markov kernel on $\mathcal{T}$, and $\left\{\Theta^{(k)}\right\}_k$ a Markov chain following P. Then, for square-integrable functions such that $\mathrm{E}\left[f^2(\Theta)\right] < \infty$, the ergodic averages are asymptotically normally distributed, i.e.*

$$\sqrt{\hat{k}} \left(\hat{k}^{-1} \sum_{k=1}^{\hat{k}} f(\Theta^{(k)}) - \mathrm{E}\left[f(\Theta)\right] \right) \Rightarrow \mathrm{N}\left[0, \sigma_f^2\right] \qquad (\mathrm{E}\left[f^2(\Theta)\right] < \infty) \tag{2.7}$$

for some finite asymptotic variance σ_f^2.

See Figure 2.2 for an illustration of the MCCLT in action. Since the variance of a Monte Carlo estimate under independent sampling is $\hat{k}^{-1} \operatorname{Var}\left[f(\Theta)\right]$, the ratio $\sigma_f^2 / \operatorname{Var}\left[f(\Theta)\right]$ captures the efficiency of the algorithm in estimating the expectation of f. Ideally, σ_f^2 does not increase exponentially as the dimension of Θ increases, since that would return us to the curse of dimensionality. Good MCMC algorithms are often domain-specific,

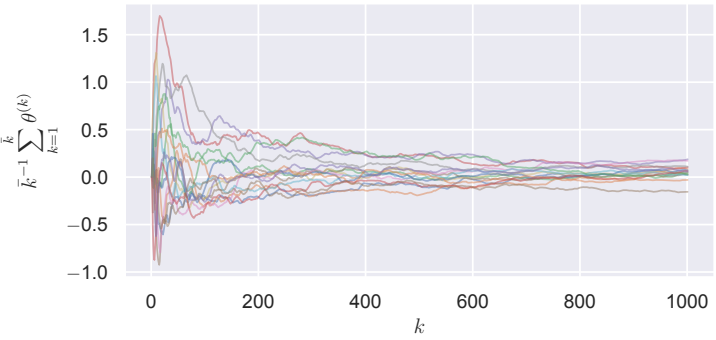


Figure 2.2: Ergodic averages of 16 variates of a Markov chain with stationary distribution $\mathrm{N}[0,1]$, started from $\theta^{(0)} = 4$. The range of the ergodic averages narrows with the runtime.

but abide by a set of principles shaped by a large body of theory and practice, see e.g. [22] for a comprehensive volume and [54] for a summary of the start of the art.

An important reason for the popularity of MCMC is that it is often easier to target a high-dimensional distribution by a sequence of simple, local moves, as opposed to designing a one-shot proposal, as in rejection sampling or importance sampling, which usually suffers from the curse of dimensionality. Even local computations are difficult in the intractable likelihood context, since they typically consist of evaluating the odds

$$\frac{\pi(\theta^\dagger)}{\pi(\theta)} \tag{2.8}$$

for some pair of values $(\theta, \theta^\dagger)$. Therefore, we cannot immediately apply standard MCMC techniques to target $\pi(\theta)$. Various strategies exist to surmount this problem, some more conventional and others more exotic. Part of the goal of this thesis is to investigate the appeal of the more exotic techniques. We continue by elaborating on the importance of the odds for MCMC simulation, and follow up with a discussion of techniques that avoid their direct evaluation. This will set the stage for the more specific case of inference for diffusion models.

2.1.1 Accept-Reject MCMC

While there are many ways of constructing kernels $P(\theta, \mathrm{d}\theta^\dagger)$ with limiting distribution $\pi(\theta)$, the most generic recipe is to first generate a *candidate value* from some simple *proposal kernel* $Q(\theta, \mathrm{d}\theta^\dagger) = \kappa(\theta^\dagger|\theta)\,\mathrm{d}\theta^\dagger$. This value is then accepted with probability

$$\alpha_\beta(\theta, \theta^\dagger) = \beta\left(\frac{\pi(\theta^\dagger)}{\pi(\theta)}\frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\right) \tag{2.9}$$

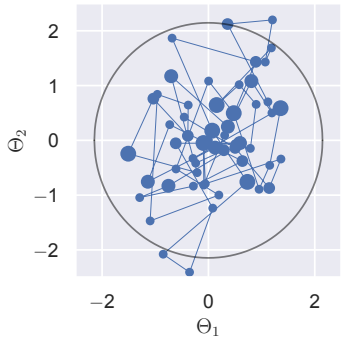


Figure 2.3: Trajectory of a Metropolis Markov chain exploring a bivariate standard Gaussian distribution. The size of the marker corresponds to the number of rejections at the location. The black circle contains a 90% credibility region.

for an appropriate *balancing function* $\beta : [0, \infty] \to [0, 1]$, and rejected otherwise, such that the Markov chain remains at θ. Notice that any intractable normalizing constant in $\pi(\theta)$ cancels out, so the acceptance probability can be computed without knowledge of the normalizing constant. The corresponding Markov kernel is

$$P(\theta, \mathrm{d}\theta^\dagger) = \alpha_\beta(\theta, \theta^\dagger) Q(\theta, \mathrm{d}\theta^\dagger) + \delta_\theta(\mathrm{d}\theta^\dagger) \int (1 - \alpha_\beta(\theta, \theta^\dagger)) Q(\theta, \mathrm{d}\theta^\dagger), \tag{2.10}$$

where δ_θ is the point mass at θ. We assume that P is *irreducible*, i.e. any part of the support of $\pi(\theta)$ can be reached with positive probability by the resulting Markov chain. This is easily ensured by choosing Q such that it support overlaps with π. In addition, P has to be *aperiodic*, which roughly corresponds to the stronger condition that the Markov chain can reach any part of the support of $\pi(\theta)$ at any iteration. Furthermore, if β is chosen such that P is reversible, the Markov chain induced by P has $\pi(\theta)$ as its stationary and limiting distribution. The reader may consult [119] for an extensive technical discussion of those aspects. Figure 2.3 provides an illustration of an accept-reject Markov chain. For our purposes, two choices of β are relevant: The standard *Metropolis-Hastings* kernel, and the *Barker* kernel, which has specific virtues in the intractable likelihood setting. The reader may also refer to [120] for a more extensive discussion of balancing functions, and to [62] for an alternative view on balancing functions, in which they serve to construct a continuous-time Markov chain.

Algorithm 1 Accept-reject MCMC targeting $\pi(\theta)$ with proposal density $\kappa(\theta^\dagger|\theta)$, balancing function β.

repeat
  $\theta^\dagger \sim \kappa(\theta^\dagger|\theta)$
  $u \sim \text{Uniform}\,[0,1]$
  $\alpha_\beta(\theta,\theta^\dagger) \leftarrow \beta\left(\frac{\pi(\theta^\dagger)}{\pi(\theta)}\frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\right)$
  if $\alpha_\beta(\theta,\theta^\dagger) \leq u$ **then**
    $\theta \leftarrow \theta^\dagger$
until ∞

2.1.2 Metropolis-Hastings Kernel

The Metropolis-Hastings algorithm, which originates with [89] and [58], is by far the most popular accept-reject algorithm. It consists of accepting the proposal $\theta^\dagger$ with probability

$$\alpha_{\text{MH}}(\theta,\theta^\dagger) = \min\left[1, \frac{\pi(\theta^\dagger)}{\pi(\theta)}\frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\right]. \tag{2.11}$$

For a given proposal kernel Q and candidate $\theta^\dagger$, α_{MH} is the largest acceptance probability which preserves reversibility. [99, 118] showed that maximizing the acceptance probability results in the optimal reversible algorithm, in the following sense.

Theorem 2 (Peskun ordering [99, 118])**.** *Let $P(\theta, \mathrm{d}\theta^\dagger)$ be an accept-reject Markov kernel targeting $\pi(\theta)$ with proposal kernel $Q(\theta, \mathrm{d}\theta^\dagger) = \kappa(\theta^\dagger|\theta)\,\mathrm{d}\theta^\dagger$ and acceptance probability*

$$\alpha_\beta(\theta,\theta^\dagger) = \beta\left(\frac{\pi(\theta^\dagger)}{\pi(\theta)}\frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\right). \qquad (\beta : [0,\infty] \to [0,1]) \tag{2.12}$$

Suppose Q and β are chosen such that P is irreducible, aperiodic and reversible. Then, $\beta(b) = \min[1,b]$ minimizes the asymptotic variance of the ergodic average $\hat{k}^{-1}\sum_{k=1}^{\hat{k}} f(\Theta^{(k)})$ for any square-integrable f.

The Peskun ordering justifies the enduring dominance of the Metropolis-Hastings choice. Nonetheless, the implication that the Metropolis kernel dominates any other assumes that the acceptance decision is reached with the same computational effort for all choices.

2.1.3 Barker Kernel

The Barker algorithm, which originates with [9], consists of accepting $\theta^\dagger$ with probability

$$\begin{aligned}\alpha_{\mathrm{B}}(\theta, \theta^\dagger) &= \frac{\pi(\theta^\dagger)\kappa(\theta|\theta^\dagger)}{\pi(\theta^\dagger)\kappa(\theta, \theta^\dagger) + \pi(\theta)\kappa(\theta^\dagger, \theta)} \\ &= \left(\frac{\pi(\theta^\dagger)}{\pi(\theta)}\frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\right) / \left(1 - \frac{\pi(\theta^\dagger)}{\pi(\theta)}\frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\right).\end{aligned} \tag{2.13}$$

It is easy to see that $\alpha_{\mathrm{B}} \leq \alpha_{\mathrm{MH}}$, thus Barker's algorithm is suboptimal in the Peskun sense. Nevertheless, the difference in acceptance probability is at most within a factor of 2:

$$2^{-1}\alpha_{\mathrm{MH}}(\theta, \theta^\dagger) \leq \alpha_{\mathrm{B}}(\theta, \theta^\dagger) \leq \alpha_{\mathrm{MH}}(\theta, \theta^\dagger). \tag{2.14}$$

As shown by [80], that implies a bound on the relative inefficiency of the Barker algorithm versus the Metropolis algorithm, given by the following theorem.

Theorem 3 (Barker variance bound [80])**.** *Suppose that the test function f is square-integrable with respect to $\pi(\theta)$, and that the Metropolis algorithm with proposal Q admits a Monte Carlo CLT for $\mathrm{E}[f(\Theta)]$, with asymptotic variance σ^2_{MH}. Then, Barker's algorithm with the same proposal admits a Monte Carlo CLT as well, with asymptotic variance σ^2_B that satisfies*

$$\sigma^2_{MH} \leq \sigma^2_B \leq 2\sigma^2_{MH} + \mathrm{Var}[\Theta]. \tag{2.15}$$

Hence, it is guaranteed not to be "substantially worse", and Section 2.3.1 shows how the intractable likelihood setting can contravene the preference of the Peskun ordering for the MH algorithm.

2.2 Augmented MCMC

For the rest of this section, we return to the specific task of targeting from a Bayesian posterior $\pi(\theta|a)$. Having introduced the framework of accept-reject MCMC, it is clear that we cannot compute the acceptance probability when $\pi(a|\theta)$ is intractable. A time-tested workaround is to find an unbiased estimator of the likelihood, i.e. a random variable B and an extended, tractable model $\pi(b, a|\theta)$ such that

$$\pi(a|\theta) = \int \pi(a, b|\theta)\mathbb{Q}(\mathrm{d}b), \tag{2.16}$$

where $\mathbb{Q}$ is an appropriate dominating measure. The random variable B is variably known as the *missing data*, *latent variable* or *auxiliary variable*, depending on the specific statistical context. We may then devise an algorithm that targets

$$\pi(\theta, b|a) \propto \pi(a|b, \theta)\pi(b|\theta)\pi(\theta). \tag{2.17}$$

The key insight is that the algorithm indirectly targets the marginal $\pi(a|\theta)$ as well: For a posterior sample $\left\{(\theta^{(1)}, b^{(1)}), \ldots, (\theta^{(\hat{k})}, b^{(\hat{k})})\right\}$, the marginal chain over Θ is distributed according to $\pi(\theta|a)$. This approach is pervasive in computational statistics and often referred to as *(missing) data augmentation*.

Example 1 (Exponential mixture of Weibulls)**.** *Suppose that*

$$A|b \sim \text{Weibull}\left[b, k\right], \quad B|\theta \sim \text{Exp}\left[\theta\right]. \qquad (k, \theta > 0) \tag{2.18}$$

Then the marginal distribution $\pi(a|\theta) = \int \text{Weibull}\left[a; b, k\right] \text{Exp}\left[b; \theta\right] \mathrm{d}\theta$ *is an intractable integral, but the joint* $\pi(a, b|\theta) = \text{Weibull}\left[a; b, k\right] \text{Exp}\left[b; \theta\right]$ *can easily be evaluated.*

We will generally refer to an algorithm targeting the extended posterior $\pi(\theta, b|a)$ as an *augmented algorithm*, and to an algorithm targeting $\pi(\theta|a)$ as a *marginal algorithm*. Two generic frameworks to design augmented algorithms are known as *Gibbs sampling* and *Pseudo-marginal MCMC*, which we explore in the following sections. A common theme with augmented algorithms is that they generate estimates with larger asymptotic variance than the hypothetical marginal chain, though exceptions exist, see e.g. [117]. Alternatively, in the case of Bayesian inference for diffusion models, it is possible to target $\pi(\theta|a)$ directly, but at a computational cost per MCMC iteration that is larger than for an algorithm targeting $\pi(\theta, b|a)$. This potentially induces a tradeoff between the computational cost per iteration of the algorithm, and the statistical efficiency of carrying out such an iteration. This tradeoff is one of the key methodological subjects of investigation of this thesis. Due to the centrality of the subject, we proceed with an abstract presentation of the main techniques for constructing augmented and marginal algorithms in the intractable likelihood setting. Their properties will motivate key choices in the design of algorithms for Bayesian diffusion inference.

2.2.1 Gibbs Sampling and Model Parameterization

One of the key paradigms in MCMC, known as *Gibbs sampling* [46, 24], consists of partitioning the target space, and updating the elements of that partition in sequence, conditional on all other elements. In the instance of targeting $\pi(\theta, b|a)$, we partition (Θ, B) into its elements Θ and B. Given the starting values $(\theta^{(k)}, b^{(k)})$, we obtain the update by way of

$$\Theta^{(k+1)} \sim \pi(\theta|b^{(k)}, a) \propto \pi(a|b^{(k)}, \theta)\pi(\theta), \tag{2.19}$$

$$B^{(k+1)} \sim \pi(b|\theta^{(k+1)}, a) \propto \pi(a|b, \theta^{(k+1)})\pi(b|\theta^{(k+1)}). \tag{2.20}$$

The distributions $\pi(\theta|b, a)$ and $\pi(b|\theta, a)$ are called the *full conditional* distributions. The corresponding transition kernel has $\pi(\theta, b|a)$ as its stationary and limiting distribution [44]. This applies for any random or deterministic ordering of full conditional updates

(*random* vs. *deterministic scan*), as long as all the full conditionals are updated at every iteration of the algorithm [115]. For many models, sampling from those full conditionals is easy or even trivial - therefore, Gibbs sampling is an example of the divide-and-conquer principle in action.

Algorithm 2 Deterministic scan Gibbs sampler with B-Θ blocking for posterior density $\pi(\theta, b|a)$, initial value (θ, b).

repeat
  $\theta \sim \pi(\theta|b, a)$
  $b \sim \pi(b|\theta, a)$
until ∞

Example 2 (Linear hierarchical model in centered parameterization)**.** *For the following "centered" hierarchical model*

$$A|b \sim \mathrm{N}\left[b, \sigma^2\right], \quad B|\theta \sim \mathrm{N}\left[\theta, \tau^2\right], \quad \pi(\theta) \propto 1. \tag{2.21}$$

a valid Gibbs sampler consists of the updates

$$B|a, \theta \sim \mathrm{N}\left[(\sigma^{-2}a + \tau^{-2}\theta)/(\sigma^{-2} + \tau^{-2}), 1/(\sigma^{-2} + \tau^{-2})\right], \tag{2.22}$$

$$\Theta|a, b \sim \mathrm{N}\left[b, \tau^2\right]. \tag{2.23}$$

In other instances, even the full conditionals $\pi(\theta|b, a)$ and $\pi(b|\theta, a)$ cannot be sampled directly. Even so, we may update each of those by way of an accept-reject step (with respective proposals $\kappa(\theta^\dagger|\theta, b)$ and $\kappa(b^\dagger|b, \theta)$) that is invariant for the conditional model. Such a procedure still has the correct limiting distribution. Indeed, a Metropolis-within-Gibbs proposal can be seen as a conventional Metropolis proposal that only changes one coordinate at a time. If that coordinate is chosen randomly, the Metropolis-within-Gibbs proposal still fulfills all the requirements laid out in Section 2.1.1. Iterating a MwG update gives a full conditional update in the limit, therefore it is asymptotically equivalent to a pure Gibbs update. Conversely, the pure Gibbs update, e.g. to $\pi(\theta|b, a)$, is a Metropolis-within-Gibbs update with proposal $\kappa(\theta|b) = \pi(\theta|b, a)$ and acceptance probability 1:

$$\begin{aligned} \min\left[1, \frac{\pi(a, \theta^\dagger|b)}{\pi(a, \theta|b)} \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\right] &= \min\left[1, \frac{\pi(a, \theta^\dagger|b)}{\pi(a, \theta|b)} \frac{\pi(\theta|b, a)}{\pi(\theta^\dagger|b, a)}\right] \\ &= \min\left[1, \frac{\pi(\theta^\dagger|b, a)}{\pi(\theta|b, a)} \frac{\pi(\theta|b, a)}{\pi(\theta^\dagger|b, a)}\right] = 1 \end{aligned} \tag{2.24}$$

Metropolis-within-Gibbs samplers are particularly attractive in computational terms when we find ourselves in the conditional independence setting where a and b are $\hat{\imath}$-dimensional, and $\pi(a, b|\theta) = \prod_{i=1}^{\hat{\imath}} \pi(a_i|b_i, \theta)\pi(b_i|\theta)$. Then, the latent variable update factorizes:

$$\pi(b|a, \theta) = \prod_{i=1}^{\hat{\imath}} \pi(b_i|a_i, \theta) \tag{2.25}$$

Hence, the Metropolis-within-Gibbs updates to B_i can be carried out independently. The efficiency of the full update to B does not suffer from the usual curse of dimensionality that affects accept-reject updates, and little is lost from doing dependent rather than independent Gibbs updates!

Just as data augmentation, Gibbs sampling is a tremendously useful technique, and the two are often complimentary. Nevertheless, Gibbs samplers are known to exhibit poor performance when the elements of the partition exhibit strong posterior dependence. In the augmentation context, B and Θ are highly dependent whenever the missing data B is substantially more informative about Θ than A - the effect of the unknown dominates the effect of the known. In the terminology of [82, 97], Gibbs samplers for augmented posteriors fail when the *fraction of missing information* is very large. Figure 2.4 gives an example of lowered efficiency in the presence of a large fraction of missing information in the hierarchical linear model context.

Definition 1 (Bayesian fraction of missing information [82, 97])**.** *Consider a joint model $\pi(a, b, \theta)$ and a square-integrable test function $f : \mathcal{T} \to \mathbf{R}$. We define the* Bayesian fraction of missing information *by*

$$\gamma_f = 1 - \mathrm{E}\left[\frac{\mathrm{Var}\left[f(\Theta)|a,b\right]}{\mathrm{Var}\left[f(\Theta)|a\right]}|a\right], \qquad (\mathrm{Var}\left[f(\Theta)|a\right] < \infty) \tag{2.26}$$

and note that the fraction is 1 if $\pi(\theta|a, b)$ is singular. The maximal correlation coefficient *is defined by*

$$\gamma = \sup_{f:\mathrm{Var}[f(\Theta)|a]<\infty} \gamma_f, \tag{2.27}$$

and corresponds to the geometric rate of convergence in (2.5)*, assuming that geometric ergodicity holds.*

This effect is particularly strong in the Bayesian diffusion inference context, where the fraction can be 1 [108]! One remedy is to reparameterize the model by way of a map $C = f(B, \Theta)$, such that C and Θ are a priori independent, i.e. $\pi(c, \theta) = \pi(c)\pi(\theta)$. (C, Θ) is known as a *noncentered parameterization* (NCP) of the model, as opposed to the *centered parameterization* (CP) (B, Θ) [43, 97]. Gibbs samplers that target the noncentered posterior typically perform well when the fraction of missing information is large.

Example 3 (Linear hierarchical model in noncentered parameterization)**.** *Consider the linear hierarchical model defined in Example 2, and define $C = B - \Theta$. Then, an equivalent formulation of the model in noncentered parameterization is given by*

$$A|c, \theta \sim \mathrm{N}\left[\theta + c, \sigma^2\right], \quad C \sim \mathrm{N}\left[0, \tau^2\right], \quad \pi(\theta) \propto 1. \tag{2.28}$$

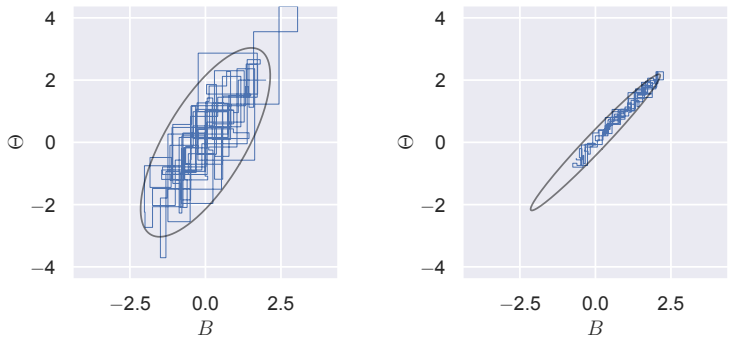


Figure 2.4: 100 iterations of the centered Gibbs sampler of Example 2 with $a = 0$, $\sigma = 1$ and $\tau = 1$ (left) or $\tau = 1/5$ (right), initialized at $(2, 2)$. The right panel shows a run of the noncentered Gibbs sampler for $\tau = 1/5$ of Example 3 in the original space (B, Θ). Mixing is degraded by the increase in the fraction of missing information from left to middle. The black circles contain 90% credibility regions.

The corresponding noncentered Gibbs sampler consists of the updates

$$C|a, \theta \sim \mathrm{N}\left[\sigma^{-2}(a-\theta)/(\sigma^{-2}+\tau^{-2}), \sigma^{-2}/(\sigma^{-2}+\tau^{-2})\right], \tag{2.29}$$

$$\Theta|a, c \sim \mathrm{N}\left[a-c, \sigma^2\right]. \tag{2.30}$$

In particular, this algorithm has a geometric rate of convergence bounded away from 1 as $\gamma \to 1$.

Figure 2.5: Graph of the hierarchical linear model in CP (left) and NCP (right).

Figure 2.6 shows how the noncentered Gibbs sampler outperforms the centered Gibbs sampler in a large fraction of missing information setting. In fact, the settings where centered and noncentered Gibbs samplers perform well are somewhat complimentary, particularly for the hierarchical linear model. Nonetheless, there are instances where posterior dependence within both (B, Θ) and (C, Θ) is large, and attempts have been made to generate synergies between the strategies [126]. Even so, in some instances no parameterization achieves good performance, and Gibbs sampling is not a panacea.

2.2.2 Pseudo-Marginal MCMC

In instances where no parameterization of the model results in a satisfactory Gibbs sampler, an algorithm that jointly updates (B, Θ) can exhibit better performance. Such

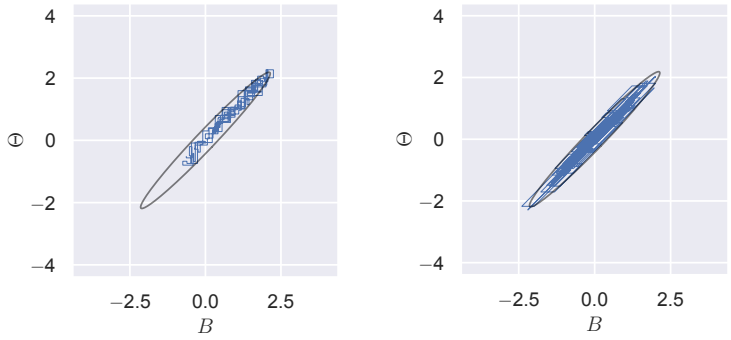


Figure 2.6: 100 iterations of the centered (left) and noncentered (right) Gibbs sampler of Examples 2 and 3 with $a = 0$, $\sigma = 1$ and $\tau = 1/5$, initialized at $(2, 2)$. The noncentered run is shown in the original space (B, Θ). The black circles contain 90% credibility regions.

algorithms are often referred to as *pseudo-marginal algorithms* [11, 4], in that they represent an approximation to marginal algorithms. A candidate pair $(b^\dagger, \theta^\dagger)$ is generated according to the proposal kernel $Q(\{b, \theta\}, \{\mathrm{d}b^\dagger, \mathrm{d}\theta^\dagger\}) = \kappa(b^\dagger, \theta^\dagger|\theta, b)\mathbb{Q}(\mathrm{d}b^\dagger)\,\mathrm{d}\theta^\dagger$ and accepted with probability

$$\alpha_\beta(\{b, \theta\}, \{b^\dagger, \theta^\dagger\}) = \beta\left(\frac{\pi(a|b^\dagger, \theta^\dagger)}{\pi(a|b, \theta)}\frac{\pi(b^\dagger|\theta^\dagger)}{\pi(b|\theta)}\frac{\pi(\theta^\dagger)}{\pi(\theta)}\frac{\kappa(\theta, b|b^\dagger, \theta^\dagger)}{\kappa(b^\dagger, \theta^\dagger|\theta, b)}\right), \tag{2.31}$$

where β could correspond either to a Metropolis or Barker balancing function. This is merely accept-reject MCMC on an extended spate space, therefore, the resulting Markov chain has limiting distribution $\pi(b, \theta|a)$. The main difference lies in the fact that B is typically high-dimensional, matching A, while Θ is usually of low and fixed dimension.

Algorithm 3 Accept-reject pseudo-marginal MCMC for posterior density $\pi(b, \theta|a)$, proposal density $\kappa(b^\dagger, \theta^\dagger|\theta, b)$, balancing function β, initial value θ.

repeat
  $b^\dagger, \theta^\dagger \sim \kappa(b^\dagger, \theta^\dagger|\theta, b)$
  $u \sim \text{Uniform}\,[0, 1]$
  $\alpha_\beta(\{b, \theta\}, \{b^\dagger, \theta^\dagger\}) = \beta\left(\frac{\pi(a|b^\dagger, \theta^\dagger)}{\pi(a|b, \theta)}\frac{\pi(b^\dagger|\theta^\dagger)}{\pi(b|\theta)}\frac{\pi(\theta^\dagger)}{\pi(\theta)}\frac{\kappa(\theta, b|b^\dagger, \theta^\dagger)}{\kappa(b^\dagger, \theta^\dagger|\theta, b)}\right)$
  if $\alpha_\beta(\{b, \theta\}, \{b^\dagger, \theta^\dagger\}) \leq u$ **then**
    $b, \theta \leftarrow b^\dagger, \theta^\dagger$
until ∞

Where $\pi(b|\theta)$ is tractable, a standard choice is to set the proposal kernel $Q(\{b, \theta\}, \{\mathrm{d}b^\dagger, \mathrm{d}\theta^\dagger\}) = \pi(b^\dagger|\theta^\dagger)\kappa(\theta^\dagger|\theta)\mathbb{Q}(\mathrm{d}b^\dagger)\,\mathrm{d}\theta^\dagger$, which is an independence proposal for B according to its prior

distribution. The simplified acceptance probability is

$$\alpha_\beta(\{b,\theta\},\{b^\dagger,\theta^\dagger\}) = \beta\left(\frac{\pi(a|b^\dagger,\theta^\dagger)}{\pi(a|b,\theta)}\frac{\pi(\theta^\dagger)}{\pi(\theta)}\frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\right). \tag{2.32}$$

which corresponds to the acceptance probability in the marginal algorithm after substituting $\pi(a|b,\theta)$ for $\pi(a|\theta)$. In this instance, the relative performance of the pseudo-marginal algorithm relative to the marginal algorithm depends on

$$\log \mathrm{Var}\left[\pi(a|B,\theta)|a,\theta\right] = \log \int (\pi(a|b,\theta) - \pi(a|\theta))^2 \pi(b|\theta)\mathbb{Q}(\mathrm{d}b), \tag{2.33}$$

keeping in mind that $\pi(a|B,\theta)$ is an unbiased estimator of $\pi(a|\theta)$. The lower the variance, the closer the performance of the two algorithms, and we recover the marginal algorithm when the variance is 0. One way of improving the estimator is to simply sample more copies of B and to average over the estimates, but it incurs the cost of additional likelihood evaluations. [112, 33] give guidance on optimizing the trade-off between estimator variance and cost of estimate evaluation.

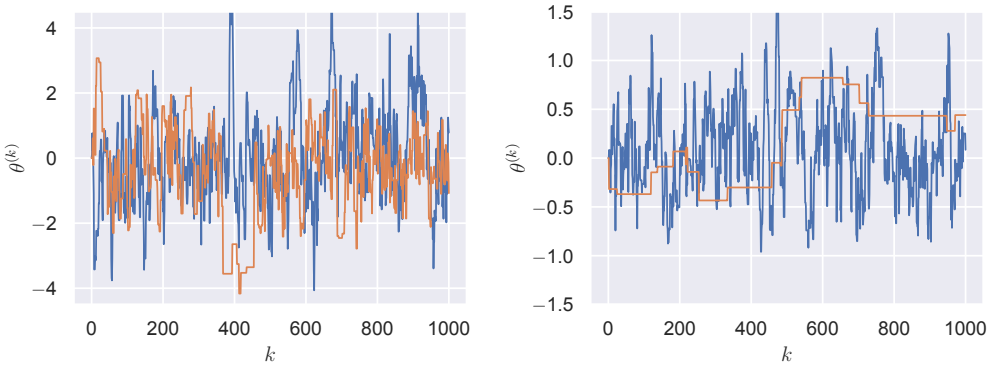


Figure 2.7: Contrasting a marginal (blue) and a pseudo-marginal (orange) trace plot for the model $A_i|b_i \sim \mathrm{N}[b_i,1]$, $B_i|\theta \sim \mathrm{N}[\theta,1]$, $\pi(\theta) \propto 1$, with number of data points $\hat{\imath} = 1$ (left) and $\hat{\imath} = 10$. Both samplers use the same proposal for Θ. The relative performance of the pseudo-marginal algorithm quickly degrades with $\hat{\imath}$.

While pseudo-marginal joint proposals can in some instances solve the Gibbs sampler's dependent update problem, the approach suffers from two shortcomings. The first is the practical challenge of designing or automatically tuning good joint proposals. The second is that in the conditional independence setting where $\pi(a,b|\theta) = \prod_i \pi(a_i|b_i,\theta)\pi(b_i|\theta)$, the log variance increases at least linearly in $\hat{\imath}$ [12]. This is most easily seen when $\pi(b^\dagger|\theta,b) = \prod_i \pi(b_i^\dagger|\theta)$, in which case

$$\log \mathrm{Var}\left[\pi(a|B,\theta)|a,\theta\right] \geq \sum_{i=1}^{\hat{\imath}} \log \mathrm{Var}\left[\pi(a_i|B_i,\theta)|a_i,\theta\right]. \tag{2.34}$$

In this instance, at least $\hat{\imath}$ samples are required to stabilize the log variance of the estimator, and $\hat{\imath}^2$ likelihood evaluations have to be performed. See Figure 2.7 for an illustration. This is somewhat mitigated by recent extensions, such as [30].

2.3 Bernoulli Factory MCMC

Having considered the main augmented MCMC approaches and their general and individual shortcomings, we face the question of whether we can forgo augmentation entirely. Up to now, we have assumed that the acceptance decision is reached by first computing the acceptance probability α_β, and then reaching a decision according to the event $\{U < \alpha_\beta : U \sim \text{Uniform}\,[0,1]\}$. Therefore, α_β is just a means to an end, and we could avoid its computation if we had another way of simulating a coin of probability α_β.

The alternative that we pursue here is to construct coins that provably have probability of heads α_β, without requiring explicit knowledge of α_β. This is made possible by the application of *Bernoulli factories*. A 2-parameter Bernoulli factory is an algorithm which takes as an input two coin flip generators with probability of heads p_1 and p_2 and generates coin flips with probability $f(p_1, p_2)$, without requiring explicit knowledge of the arguments. The Bernoulli factory problem has been investigated in its own right by e.g. [94, 79], but we are especially interested in applying it to MCMC balancing functions. Suppose, then, that the acceptance probability for an MCMC algorithm can be written as

$$\alpha_\beta(\theta, \theta^\dagger) = \beta\left(\frac{\pi(a,\theta^\dagger)}{\pi(a,\theta)}\frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\right) = \beta\left(\frac{c_1 p_1}{c_2 p_2}\right), \qquad (c_1, c_2 > 0, \quad p_1, p_2 \in [0,1]) \quad (2.35)$$

where c_1, p_1, c_2 and p_2 are functions of θ and $\theta^\dagger$, and that we possess a Bernoulli factory with $f(p_1, p_2) = \alpha_\beta(\theta, \theta^\dagger)$. Then, we could simulate the acceptance decision by generating p_1- and p_2-flips, and applying the $f(p_1, p_2)$-factory to those flips. The flips can be obtained by way of unbiased estimators $\bar{p}_1, \bar{p}_2 \in [0,1]$, simulating $\{U < \bar{p}_1 : U \sim \text{Uniform}\,[0,1]\}$ or $\{U < \bar{p}_2 : U \sim \text{Uniform}\,[0,1]\}$ respectively. While data augmentation may be used to construct the unbiased estimators, the Bernoulli factory MCMC algorithm still has the advantageous statistical properties of the marginal algorithm.

As discussed by [7, 79], there is no general solution to the factory problem for the Metropolis balancing function. Conversely, [52] *do* develop a Bernoulli factory for the Barker balancing function. Since the relative efficiency shortfall of Barker against Metropolis is bounded, while the potential cost of augmentation is unbounded, a marginal Barker algorithm could easily prevail over a pseudo-marginal Metropolis algorithm. Therefore, we will now investigate the Barker balancing function factory, which [52] dub the *2-coin algorithm*, in more detail.

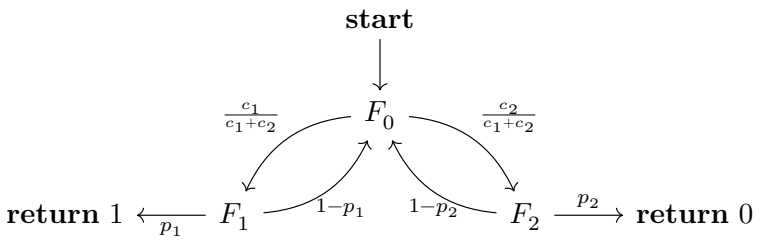


Figure 2.8: Probability flow diagram of the vanilla 2-coin algorithm. Nodes (F_0, F_1, F_2) refer to coin flips, edges give the probabilities of moving to the corresponding node.

2.3.1 2-Coin Barker Algorithm

We will factorize the Barker acceptance odds to match the form

$$\frac{\pi(a,\theta^\dagger)}{\pi(a,\theta)}\frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)} = \frac{c_1(\theta,\theta^\dagger)p_1(\theta,\theta^\dagger)}{c_2(\theta,\theta^\dagger)p_2(\theta,\theta^\dagger)}. \qquad (c_1,c_2>0, \quad p_1,p_2\in[0,1]) \tag{2.36}$$

Notice that this factorization is usually not unique, and that we typically suppress the notational dependence of $p_{1,2}$ and $c_{1,2}$ on $(\theta,\theta^\dagger)$. The Bernoulli factory that generates coins with those odds is known as the *2-coin algorithm*, originally proposed by [52]. Figure 2.8 illustrates the steps of the algorithm.

Critically, the number of iterations until the algorithm terminates is a geometric random variable with expectation

$$\frac{c_1+c_2}{c_1p_1+c_2p_2}, \tag{2.37}$$

where higher coin probabilities p_1 and p_2 result in faster termination. Therefore, it is crucial to use a factorization of the acceptance odds that maximizes p_1 and p_2 while still providing unbiased estimators $\bar{p}_1$ and $\bar{p}_2$.

Example 4 (Weibull mixture likelihood [122])**.** *Suppose that*

$$A|b\sim\text{Weibull}\left[b,k\right], \quad B|\theta\sim\mathbb{Q}_\theta, \quad \pi(\theta)\propto 1. \qquad (k,\theta>0) \tag{2.38}$$

for some measure $\mathbb{Q}_\theta(\mathrm{d}b)$*. Then*

$$\pi(\theta|a)\propto\int\text{Weibull}\left[a;b,k\right]\mathbb{Q}_\theta(\mathrm{d}b)=\mathrm{E}_{\mathbb{Q}_\theta}\left[\text{Weibull}\left[a;B,k\right]\right]. \tag{2.39}$$

Moreover, using the bound $\text{Weibull}\left[a;b,k\right]\le k/(ea)\overset{\text{def}}{=}z$*, we write the acceptance odds as*

$$\frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\frac{\pi(\theta^\dagger|a)}{\pi(\theta|a)} = \frac{\overbrace{\kappa(\theta|\theta^\dagger)}^{c_1}}{\underbrace{\kappa(\theta^\dagger|\theta)}_{c_2}}\frac{\overbrace{z^{-1}\pi(\theta^\dagger|a)}^{p_1}}{\underbrace{z^{-1}\pi(\theta|a)}_{p_2}}, \tag{2.40}$$

and obtain coins of probability $z^{-1}\pi(\theta|a)$ *by way of the event probability*

$$\Pr\left[z^{-1}\,\text{Weibull}\left[a;B,k\right] \leq U|\theta\right] = z^{-1}\pi(\theta|a), \quad B|\theta \sim \mathbb{Q}_\theta, \quad U \sim \text{Unif}\left[0,1\right] \tag{2.41}$$

2.3.2 Portkey Barker Algorithm

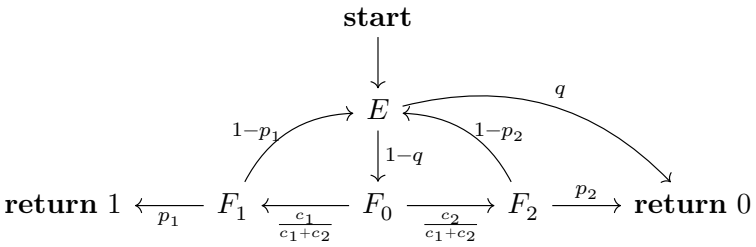


Figure 2.9: Probability flow diagram of the portkey 2-coin algorithm. Nodes (E, F_0, F_1, F_2) refer to coin flips, edges give the probabilities of moving to the corresponding node.

Even if the 2-coin factorization has been optimally designed, it is likely that p_1 or p_2 will be very small in some parts of the state space. In such cases, it is often preferable to terminate the algorithm early and reject the proposal $\theta^\dagger$. [122] propose a variant of the vanilla 2-coin Barker algorithm that incorporates such an "escape hatch", which they call the *Portkey Barker algorithm*. As displayed in Figure 2.9, at every iteration of the 2-coin algorithm, it flips an additional coin with probability of heads q, and exits if that coin comes up heads, thereby rejecting the proposal. The resulting acceptance probability is

$$\begin{aligned} &\frac{\pi(\theta^\dagger|a)\kappa(\theta|\theta^\dagger)}{\pi(\theta^\dagger|a)\kappa(\theta|\theta^\dagger) + \pi(\theta|a)\kappa(\theta^\dagger|\theta) + (c_1+c_2)(q^{-1}-1)} \\ &\quad \leq \frac{\pi(\theta^\dagger|a)\kappa(\theta|\theta^\dagger)}{\pi(\theta^\dagger|a)\kappa(\theta|\theta^\dagger) + \pi(\theta|a)\kappa(\theta^\dagger|\theta)}, \end{aligned} \tag{2.42}$$

which implies a statistically less efficient algorithm. The mean number of iterations until the algorithm terminates is again a geometric random variable with expectation

$$\left(q + (1-q)\frac{c_1 p_1 + c_2 p_2}{c_1 + c_2}\right)^{-1}, \tag{2.43}$$

which for nonzero q is bounded above regardless of p_1 and p_2. In practice, p_1 and p_2 tend to be small when the acceptance probability according to the vanilla algorithm would have been low. Therefore, even though the Portkey version is statistically less efficient, it can avoid long 2-coin run times that often result in rejections anyway.

2.4 Assessing MCMC Performance

Since we have loosely discussed the "efficiency" of various MCMC frameworks, it is worth elaborating on that notion. Two aspects enter into the efficiency of an algorithm: The cost of executing an iteration, and the degree of dependence of the Markov chain. Under Theorem 1, the error variance of the $\hat{k}$-sample ergodic average in estimating $\mathrm{E}\left[f(\Theta)\right]$ for square-integrable f is $\sigma_f^2/\hat{k}$, whereas it is $\mathrm{Var}\left[f(\Theta)\right]/\hat{k}$ for a Monte Carlo estimator based on independent samples. Therefore, we define the notion of *effective sample size* (ESS) by

$$\underbrace{(\text{effective sample size})}_{\text{ESS}} = \hat{k}\frac{\mathrm{Var}\left[f(\Theta)\right]}{\sigma_f^2}, \tag{2.44}$$

where $\mathrm{Var}\left[f(\Theta)\right]/\sigma_f^2$ is the effective sampling rate per iteration. This presumes that the Markov chain admits a Monte Carlo central limit theorem, e.g. by way of geometric ergodicity - see [70, 107] for a discussion of such conditions. In practice, the theorem is typically taken for granted, in the hope that irregular estimates indicate when it fails to hold. Notice that the ESS depends on the specific function f - an algorithm can have high effective sampling rate for some functions and low rates for others. Since we are interested in comparing the performance of MCMC algorithms with different, or even random runtimes per iteration, we adopt the *average runtime per effective sample* metric:

$$\underbrace{(\text{avg. time per eff. sample})}_{\text{T/ES}} = \underbrace{(\text{avg. time per iteration})}_{\text{T/I}} \times \frac{\sigma_f^2}{\mathrm{Var}\left[f(\Theta)\right]}. \tag{2.45}$$

This metric depends on both hardware and implementation, therefore it only has external validity if the implementations of all algorithms under consideration give similar attention to exploiting the available hardware. There are other, possibly conceptually superior notions of performance, but this one has the benefit of being amenable to estimation from empirical MCMC output. As it happens, all the quantities in (2.45) are typically highly intractable - and even more so for an intractable likelihood algorithm, and we have to resort to Monte Carlo estimation to assess the quality of our Monte Carlo estimate.

T/I is usually assessed by measuring and averaging the CPU wall time at each iteration. The asymptotic variance can be estimated by reference to its representation in terms of the autocovariance of the Markov chain:

$$\sigma_f^2 = \mathrm{Var}\left[f(\Theta)\right] + 2\sum_{k=1}^{\infty}\mathrm{Cov}_P\left[f(\Theta^{(0)}), f(\Theta^{(k)})\right], \tag{2.46}$$

where the covariance operator is applied with respect to the Markov chain P. The most straightforward approach to estimation then consists of truncating the infinite sum, and plugging in the empirical covariance of the Markov chain. The truncation is necessary to obtain a computable estimator, but it induces a bias-variance trade-off. Heuristics for

navigating that tradeoff were proposed e.g. by [116], whose recommendations we follow here. A more recent alternative for asymptotic variance estimation is presented in [123]. The reader may also consult [47] for a classic reference.

Note that estimation of σ_f^2 is substantially more difficult than the expectation $\mathrm{E}[f(\Theta)]$ itself, so expecting an MCMC algorithm to provide good estimates of its own variance may be seen as well in excess of its original task, and therefore somewhat incoherent. Nevertheless, maybe due to the lack of convincing alternatives, estimation of σ_f^2 for select functions f continues to be the standard approach to validating MCMC runs, and we adhere to that standard when assessing simulation output in Chapters 5, 6 and 7.

2.5 Discussion

Having introduced many complementary and alternative approaches to address the intractable likelihood problem, we need to draw some conclusions for our diffusion inference strategy. The general theme will be that some degree of augmentation is unavoidable in this setting - the question is how much of it is necessary, and whether the Bernoulli factory MCMC approach presents a favorable tradeoff in our setting, which we investigate in Chapters 5 and 6. We will follow the principle of maximally exploiting conditional independence through Gibbs sampling since it affords us more hope of developing algorithms that scale favorably (close to linearly) with the amount of input data.

There are important aspects of MCMC theory and practice that factor into the algorithms that we develop in this thesis, but which are tangential to the intractable likelihood setting and therefore not discussed in this chapter. One such topic is the *optimal scaling* of conditional proposals of for $\kappa(\theta^\dagger|\theta)$. Particularly conclusive results on optimal proposals exist for many Metropolis-Hastings algorithms on $\mathbf{R}^d$, see e.g. [106] for the Metropolis algorithm and [1] for the Barker algorithm. Notice that those do not take into account the possiblity that iteration time may depend on the proposal. Such results raise the question of how to implement optimal proposals, which is addressed in the *adaptive MCMC* literature, see e.g. [5].

3 Data Augmentation for Stochastic Differential Equations

In order to apply the data augmentation approach of Chapter 2 to diffusion models, we require a better understanding of their properties, and the ability to simulate their trajectories. It will be sufficient to understand *time invariant Itō diffusions* with SDE representation

$$\mathrm{d}V_t = \mu(V_t)\,\mathrm{d}t + \sigma(V_t)\,\mathrm{d}W_t, \qquad (t \in [0, \omega]) \tag{3.1}$$

where W is a *Wiener process*. Time invariance means that μ and σ do not depend on t, other than through V_t. Throughout the thesis, we require that μ and σ are measurable functions that almost surely satisfy

$$\int_0^\omega \mu(V_t)\,\mathrm{d}t < \infty, \quad \int_0^\omega \sigma(V_t)\,\mathrm{d}t < \infty. \tag{3.2}$$

In this chapter, we suppress dependence of μ and σ on any parameters θ, which we assume to be known and fixed for now. They will be reintroduced later on.

The development of a data augmentation scheme is first step towards applying the intractable likelihood methods from Chapter 2. We begin in Section 3.1 by providing conditions under which SDEs give rise to well-defined stochastic processes. Section 3.2 introduces some properties of SDEs that the derivation of the augmentation scheme in the following Section 3.3 relies on. Section 3.4 discusses variations of that augmentation scheme. Finally, Section 3.5 gives a brief account of approximation schemes for SDEs.

3.1 Diffusion Processes as SDE Solutions

To use an SDE as a statistical model, we first have to establish that an SDE implies a well-defined stochastic process. Preferably, we establish that the SDE has a *strong and unique solution*, i.e. for a given Wiener process W, almost every variate w implies a unique solution v. The following theorem provides for existence of a strong solution when μ and σ are sufficiently smooth.

Theorem 4 (Strong solution of stochastic differential equations [96])**.** *Let μ and σ be* Lipschitz-continuous*, i.e.*

$$|\mu(a) - \mu(b)| + |\sigma(a) - \sigma(b)| \leq c\,|a - b|\,, \tag{3.3}$$

for some constant c and any $a, b \in \mathbf{R}$. Then, the SDE

$$\mathrm{d}V_t = \mu(V_t)\,\mathrm{d}t + \sigma(V_t)\,\mathrm{d}W_t \qquad (t \in [0, \omega]) \tag{3.4}$$

has a strong and unique t-continuous solution for every initial condition $\{V_0 = v_0\}$. Moreover, V is adapted to the natural filtration *induced by W.*

Example 5 (Lipschitz continuity for Ornstein-Uhlenbeck SDEs)**.** *Consider the OU SDE with $\mu(a) = -\beta a$, $\sigma(a) = \sigma$, $\beta, \sigma > 0$. Then,*

$$|\mu(a) - \mu(b)| + |\sigma(a) - \sigma(b)| = |\beta|\,|a - b|\,. \tag{3.5}$$

Thus all Ornstein-Uhlenbeck SDEs have Lipschitz-continuous drift and volatility functions and have strong and unique solutions.

Notice that for some SDEs, V may be confined with probability 1 to a set $\mathcal{V} \subset \mathbf{R}$. Some models are not Lipschitz-continuous, but they can still be shown to imply unique and continuous solutions once statements are restricted to $\mathcal{V}$.

An SDE with a unique solution defines a *diffusion process* with Markovian transition density $\pi(v_{\dot\tau}|v_{\dot\tau})$ for any $0 < \dot\tau < \ddot\tau < \infty$, which fully characterizes the diffusion process. Without loss of generality due to the time invariance of Itō diffusions, we merely need consider the case $\pi(v_\omega|v_0)$. In some cases, it is possible to solve the SDE, and obtain the explicit transition density. These solution methods are highly specific, as exemplified by the solution of the OU process.

Example 6 (Solution of the OU SDE by variation of parameters)**.** *Consider an OU process with SDE $\mathrm{d}V_t = -\beta V_t\,\mathrm{d}t + \sigma\,\mathrm{d}W_t$. Apply* Itō's formula*, introduced in Section 3.2.3, with $f(a,t) = ae^{\beta t}$ and $X_t = f(V_t, t)$:*

$$\mathrm{d}X_t = \partial_t f(V_t, t)\,\mathrm{d}t + \partial_{V_t} f(V_t, t)\,\mathrm{d}V_t = e^{\beta t}\sigma\,\mathrm{d}W_t \tag{3.6}$$

Integrating from 0 to ω, obtain

$$X_\omega = X_0 + \int_0^\omega e^{\beta t}\sigma\,\mathrm{d}W_t \quad \Leftrightarrow \quad V_\omega = e^{-\beta\omega}V_0 + \sigma\int_0^\omega e^{\beta(t-\omega)}\,\mathrm{d}W_t, \tag{3.7}$$

and observe that the stochastic integral with deterministic integrand is a Gaussian random variable. Furthermore, by the 0-mean and isometry properties of those integrals, we find that it has moments

$$\mathrm{E}\,[V_\omega|V_0] = e^{-\beta\omega}V_0 + \sigma\,\mathrm{E}\left[\int_0^\omega e^{\beta(t-\omega)}\,\mathrm{d}W_t\right] = e^{-\beta\omega}V_0 \tag{3.8}$$

$$\mathrm{Var}\,[V_\omega|V_0] = \sigma^2\,\mathrm{E}\left[\left(\int_0^\omega e^{\beta(t-\omega)}\,\mathrm{d}W_t\right)^2\right] = \sigma^2\int_0^\omega e^{2\beta(t-\omega)}\,\mathrm{d}t = \frac{\sigma^2}{2\beta}(1 - e^{-2\beta\omega}). \tag{3.9}$$

Accordingly,

$$V_\omega|V_0 \sim \mathrm{N}\left[e^{-\beta\omega}V_0, \frac{\sigma^2}{2\beta}(1 - e^{-2\beta\omega})\right]. \tag{3.10}$$

Techniques for solving SDEs do not generalize, and therefore $\pi(v_\omega|v_0)$ is not available for almost all models. Nonetheless, through data augmentation we can find an augmented density $\pi(v_{(0,\omega]}|v_0)$ for the *complete path* $V_{(0,\omega]}$.

3.2 Theory and Properties of Itō Diffusions

The purpose of this section is to present some theory of SDEs that allow us to express their law in terms of a more simple process. In particular, we will find that under mild conditions, V is equivalent to a process X with constant volatility. In addition, we will state conditions under which the support of a diffusion is nested within the support of another diffusion with different drift, or even without drift. This will allow us to formulate the density $\pi(v_{(0,\omega]}|v_0)$ with respect to a simple dominating measure.

3.2.1 Markov Property and Likelihood

From a statistical perspective, the *Markov property* is potentially the most useful property of diffusion processes, because it allows for the factorization of its likelihood and its characterization by a transition law. We have been referring to the Markov property informally as a form of memorylessness, but due to its importance we include a more formal statement here. Recall that V is adapted to the filtration $\{\mathcal{F}_t\}_{t\geq 0}$. The (weak) Markov property implies that the information provided by the past about the future, represented by the filtration, is entirely reflected in the present:

$$\Pr\left[V_{\ddot{\tau}} \in E|\mathcal{F}_{\dot{\tau}}\right] = \Pr\left[V_{\ddot{\tau}} \in E|V_{\dot{\tau}}\right]. \qquad (0 < \dot{\tau} < \ddot{\tau} < \infty, \quad E \in \mathcal{F}) \tag{3.11}$$

The strong Markov property strengthens this statement to the case where $\dot{\tau}$ is replaced with a random *stopping time* $\dot{T}$. The fact that the Markov property applies to diffusion processes justifies our focus on its transition law.

Theorem 5 (Strong Markov property for diffusions [96])**.** *Let V be a diffusion process whose driving Brownian motion induces the natural filtration $\{\mathcal{F}_t\}_{t\geq 0}$. Then, the strong Markov property holds, i.e.*

$$\Pr\left[V_{\ddot{\tau}} \in E|\mathcal{F}_{\dot{T}}\right] = \Pr\left[V_{\ddot{\tau}} \in E|V_{\dot{T}}\right] \qquad (\dot{T} < \ddot{\tau}, \quad E \in \mathcal{F}). \tag{3.12}$$

for stopping times $\dot{T}$.

Corollary 1 (Diffusion likelihood factorization)**.** *As a direct consequence, the transition density $\pi(v_{\ddot{\tau}}|v_{\dot{\tau}})$ fully characterizes V. Moreover, for any partition τ of $[0,\omega]$, the likelihood $\pi(v_{\tau\setminus\{0\}}|v_0)$ factorizes to*

$$\pi(v_{\tau\setminus\{0\}}|v_0) = \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(v_{\ddot{\tau}}|v_{\dot{\tau}}). \tag{3.13}$$

Therefore, it is sufficient to get a handle on the individual transition density terms $\pi(v_{\ddot{\tau}}|v_{\dot{\tau}})$ in order to evaluate the whole likelihood.

3.2.2 Quadratic Variation

A critical property of a diffusion is that its *quadratic variation* is deterministic. The elementary definition of quadratic variation for a stochastic process V and a partition τ of $[0, t]$ is

$$\langle V \rangle_t = \lim_{\text{mesh}[\tau]\to 0} \sum_{(\dot{\tau}\sim\ddot{\tau})\in\tau} (V_{\ddot{\tau}} - V_{\dot{\tau}})^2, \qquad (t \in [0, \omega]) \tag{3.14}$$

where $\text{mesh}\,[\tau] = \max_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \ddot{\tau} - \dot{\tau}$. In the case of diffusions, the resulting expression is particularly simple.

Theorem 6 (Quadratic variation of diffusion processes [96])**.** *Let V be a diffusion process with SDE $\mathrm{d}V_t = \mu(V_t)\,\mathrm{d}t + \sigma(V_t)\,\mathrm{d}W_t$. Then the limit* (3.14) *converges in probability to*

$$\langle V \rangle_t = \int_0^t \sigma^2(V_u)\,\mathrm{d}u. \qquad (t \in [0, \omega]) \tag{3.15}$$

The same fact may also be expressed in SDE form by $\mathrm{d}\langle V \rangle_t = \sigma^2(V_t)\,\mathrm{d}t$. This deterministic relationship is of particular importance for statistical inference, because for *any* $t \in (0, \omega]$, a path $v_{[0,t]}$ with quadratic variation $\langle v \rangle_t$ is only compatible with certain volatility coefficients σ. Therefore, knowledge of $v_{[0,t]}$ restricts the possible inferences for σ.

3.2.3 Itō's Formula and Closure under Transformation

As seen in Example 6, we are frequently given a diffusion process V and wish to translate statements about V to $f(V)$. This need is addressed by one of the essential results of stochastic calculus, variably known as *Itō's formula* or *Itō's lemma*. A plethora of versions of this theorem exist for various forms of Itō processes. For our purposes, it is sufficient to state a result that applies to time-invariant Itō diffusions and time-invariant transformations f.

Theorem 7 (Itō's formula for diffusion processes [96])**.** *Let V be a diffusion process with SDE $\mathrm{d}V_t = \mu(V_t)\,\mathrm{d}t + \sigma(V_t)\,\mathrm{d}W_t$, and $f : \mathbf{R} \to \mathbf{R}$ be twice differentiable. Then,*

$$\begin{aligned} \mathrm{d}f(V_t) &= f'(V_t)\,\mathrm{d}V_t + 2^{-1} f''(V_t)\,\mathrm{d}\langle V \rangle_t \\ &= f'(V_t)(\mu(V_t)\,\mathrm{d}t + \sigma(V_t)\,\mathrm{d}W_t) + 2^{-1} f''(V_t)\sigma^2(V_t)\,\mathrm{d}t \\ &= (f'(V_t)\mu(V_t) + 2^{-1} f''(V_t)\sigma(V_t))\,\mathrm{d}t + f'(V_t)\sigma^2(V_t)\,\mathrm{d}W_t. \end{aligned} \tag{3.16}$$

Furthermore, the new SDE must have as its unique solution the diffusion process $f(V)$.

Example 7 (Squared Bessel process)**.** *Set* $f(a) = a^2$*,* $R = f(W)$*. Then,*

$$\mathrm{d}R_t = 2W_t\,\mathrm{d}W_t + \mathrm{d}t = \mathrm{d}t + 2\sqrt{R_t}\,\mathrm{d}W_t, \tag{3.17}$$

which is the Squared Bessel process*.*

Therefore, Itō's formula shows that the class of diffusion processes is closed under twice differentiable transformations f, and provides us with an algebra of diffusions. An apparent limitation of the formula is that it requires f to be twice differentiable everywhere, which is not the case of many important transformations such as $f(a) = \sqrt{a}$. This is usually bypassed by first demonstrating that V is almost surely confined to a set $\mathcal{V} \subset \mathbf{R}$, whereupon it is valid to apply Itō's formula if f is twice differentiable on $\mathcal{V}$.

3.2.4 Change of Volatility and the Lamperti Transform

A critical property of many diffusion processes that is critical for the ability to simulate sample paths, among other things, is that they can be mapped 1-to-1 to a diffusion process with constant volatility. A diffusion process is said to be *reducible* if there exists a transformation $\eta : \mathcal{V} \to \mathcal{X}$ such that

$$\mathrm{d}\eta(V_t) = \delta \circ \eta(V_t)\,\mathrm{d}t + \mathrm{d}W_t. \tag{3.18}$$

for some drift function $\delta : \mathcal{X} \to \mathbf{R}$, i.e. the *reduced process* $X = \eta(V)$ has unit volatility coefficient. This transformation is often referred to as the *Lamperti transform*, but early applications precede that naming [29].

From Itō's formula, we know that the transformation $\eta(V)$ of a diffusion process V has unit volatility coefficient precisely when

$$\eta'(a)\sigma(a) = 1 \quad \Rightarrow \quad \eta(a) = \int_{v^*}^{a} \frac{\mathrm{d}b}{\sigma(b)}. \qquad (a, v^* \in \mathcal{V}) \tag{3.19}$$

Notice that v^* is merely a translation of X, and could bet set such that $\eta(v^*) = 0$. After fixing some v^*, we obtain a bijective transformation, and we denote its inverse $\eta^{-1} : \mathcal{X} \to \mathcal{V}$. Assuming σ' is continuously differentiable, we insert η back into Itō's formula, and obtain

$$\delta(a) = \left(\frac{\mu}{\sigma} - \frac{\sigma'}{2}\right) \circ \eta^{-1}(a). \qquad (a \in \mathcal{X}) \tag{3.20}$$

Theorem 8 (Lamperti transform [29])**.** *Let* V *be a diffusion process with support* $\mathcal{V}$ *and continuously differentiable volatility function* σ *on the support, and define the Lamperti transform*

$$\eta(a) = \int_{v^*}^{a} \frac{\mathrm{d}b}{\sigma(b)}. \qquad (a, v^* \in \mathcal{V}). \tag{3.21}$$

Then $X = \eta(V)$ *has constant volatility, and SDE representation*

$$\mathrm{d}X_t = \delta(X_t)\,\mathrm{d}t + \mathrm{d}W_t. \tag{3.22}$$

Example 8 (Lamperti transform for Ornstein-Uhlenbeck processes)**.** *Consider the* OU $[\beta, \sigma]$ *process with* $\mu(a) = -\beta a$, $\sigma(a) = \sigma$, $\beta, \sigma > 0$. *Then*

$$\eta(a) = \int_0^a \frac{\mathrm{d}b}{\sigma} = \frac{a}{\sigma}, \quad \delta(a) = -\beta a. \tag{3.23}$$

Hence, the OU $[\beta, \sigma]$ *process is reducible to the* OU $[\beta, 1]$ *process.*

Conveniently, the transition density of V implies the transition density for X, and vice versa, by the change of variable formula:

$$\pi(v_\omega | v_0) = |\eta'(v_\omega)| \, \pi(x_\omega | x_0). \tag{3.24}$$

Notice that for the Lamperti transform to be useful in practice, we require the integral in (3.19) to be available in closed form. Moreover, while the reducibility requirement is trivial in one dimension, it is much more restrictive in multiple dimensions. [2] gives sufficient conditions for the existence of the transformation in multiple dimensions.

3.2.5 Change of Drift and the Girsanov Theorem

While there is no tractable transformation that modifies drift with the same ease as in the volatility case, we can apply a relationship between the law of two diffusions with different drift and identical volatility. This relationship is known as *Girsanov's theorem*, and we will apply a version of the result that applies to Itō diffusions.

Theorem 9 (Girsanov's theorem [96])**.** *Suppose that the diffusion processes* V *and* $\tilde{V}$ *solve the SDEs*

$$\mathrm{d}V_t = \mu(V_t)\,\mathrm{d}t + \sigma(V_t)\,\mathrm{d}W_t, \qquad (V_0 = v_0) \tag{3.25}$$

$$\mathrm{d}\tilde{V}_t = \tilde{\mu}(\tilde{V}_t)\,\mathrm{d}t + \sigma(\tilde{V}_t)\,\mathrm{d}W_t. \qquad (\tilde{V}_0 = \tilde{v}_0) \tag{3.26}$$

and that the probability measure $\mathbb{V}$ *is induced by* V*. Define the scaled drift differential*

$$\gamma(a) = \frac{\tilde{\mu}(a) - \mu(a)}{\sigma(a)}, \qquad (a \in \mathcal{V}) \tag{3.27}$$

and assume that it is square-integrable. Furthermore, let the Doleans-Dade-exponential

$$\exp\left[\int_0^t \gamma(V_u)\,\mathrm{d}W_u - \frac{1}{2}\int_0^t \gamma^2(V_u)\,\mathrm{d}u\right] \qquad (t \in [0, \omega]) \tag{3.28}$$

be a Martingale under $\mathbb{V}$*. A sufficient condition is* Novikov's condition*, given by*

$$\mathrm{E}\left[\exp\left[\int_0^\omega \gamma^2(V_t)\,\mathrm{d}t\right] | v_0\right] < \infty. \tag{3.29}$$

Then, we may define

$$\tilde{W}_t = W_t - \int_0^t \gamma(V_u)\,\mathrm{d}u, \qquad (t \in [0, \omega]) \tag{3.30}$$

and a new probability measure $\tilde{\mathbb{V}}$ *under which* $\tilde{W}_t$ *is a Brownian motion. Therefore,* V *can be represented by the SDE*

$$\mathrm{d}V_t = \tilde{\mu}(V_t)\,\mathrm{d}t + \sigma(V_t)\,\mathrm{d}\tilde{W}_t. \tag{3.31}$$

Furthermore, $\tilde{\mathbb{V}}$ *is absolutely continuous with respect to* $\mathbb{V}$*, and the* Radon-Nikodym-derivative *(RND) of* $\tilde{\mathbb{V}}$ *with respect to* $\mathbb{V}$ *is given by the Doleans-Dade-exponential:*

$$\begin{aligned} \frac{\mathrm{d}\tilde{\mathbb{V}}}{\mathrm{d}\mathbb{V}}(v_{(0,\omega]}) &= \exp\left[\int_0^\omega \gamma(v_t)\,\mathrm{d}W_t - \frac{1}{2}\int_0^\omega \gamma^2(v_t)\,\mathrm{d}t\right] \\ &= \exp\left[\int_0^\omega \gamma(v_t)\,\mathrm{d}\tilde{W}_t + \frac{1}{2}\int_0^\omega \gamma^2(v_t)\,\mathrm{d}t\right], \end{aligned} \tag{3.32}$$

that is, the DDE is the density of the path $x_{(0,\omega]}$ *under* $\tilde{\mathbb{V}}$ *with respect to* $\mathbb{V}$*.*

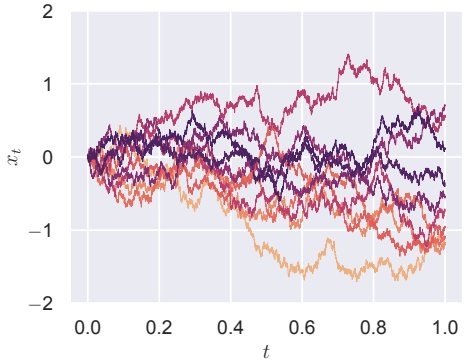


Figure 3.1: Illustration of the Girsanov theorem in action. The plotted paths were sampled from the Wiener measure, and colored according to the Radon-Nikodym derivative of the measure induced by the OU process $\mathrm{d}X_t = -X_t\,\mathrm{d}t + \mathrm{d}W_t$ against the Wiener measure. Since the OU process reverts to 0, paths that deviate farther from 0 have lower RND.

Therefore, if $x_{(0,\omega]}$ has positive support under the diffusion measure $\tilde{\mathbb{V}}$ associated with $\tilde{\mu}$, it also has positive support under $\mathbb{V}$ associated with μ. In particular, we may use the special case $\mu = 0$, and express the density of V under $\tilde{\mathbb{V}}$ with respect to a driftless measure. Figure 3.1 illustrates such an application of the Girsanov theorem.

3.3 Complete Transition Density

With Girsanov's theorem in hand, we can obtain the density of $V_{(0,\omega]}$ with respect to a tractable product measure. Since we will have to customize the result to various settings, we provide a full derivation here, mimicking [29, 18]. To do so, we first need to reduce V to align its volatility with W. Setting $X = \eta(V)$, we obtain

$$\mathrm{d}X_t = \delta(X_t)\,\mathrm{d}t + \mathrm{d}W_t, \tag{3.33}$$

$$\delta(a) = \left(\frac{\mu}{\sigma} - \frac{\sigma'}{2}\right) \circ \eta^{-1}(a), \qquad (a \in \mathcal{X}) \tag{3.34}$$

and define $\mathbb{X}$ as the measure induced induced by X. Let $\mathbb{W}$ be the measure under which X is a Brownian motion, and assume that

$$\exp\left[\int_0^\omega \delta(v_t)\,\mathrm{d}W_t - \frac{1}{2}\int_0^\omega \delta^2(v_t)\,\mathrm{d}t\right] \tag{3.35}$$

is a Martingale. Then, by Theorem 9 $\mathbb{X}|x_0$ is absolutely continuous with respect to $\mathbb{W}|x_0$, and the Radon-Nikodym-derivative is given by

$$\begin{aligned}\frac{\mathrm{d}\mathbb{X}|x_0}{\mathrm{d}\mathbb{W}|x_0}(x_{(0,\omega]}) &= \exp\left[\int_0^\omega \delta(X_t)\,\mathrm{d}W_t + \frac{1}{2}\int_0^\omega \delta^2(X_t)\,\mathrm{d}t\right] \\ &= \exp\left[\int_0^\omega \delta(X_t)\,\mathrm{d}X_t - \frac{1}{2}\int_0^\omega \delta^2(X_t)\,\mathrm{d}t\right].\end{aligned} \tag{3.36}$$

For this expression to be practically useful, we also require that δ be continuously differentiable on $\mathcal{X}$. It is sufficient to require that μ be once and σ be twice continuously differentiable on $\mathcal{V}$. Also define the integrated drift

$$\Delta(a) = \int \delta(a)\,\mathrm{d}a. \tag{3.37}$$

Then, we can remove the stochastic integral by applying Itō's lemma to $\Delta(X_t)$:

$$\begin{aligned}&\mathrm{d}\Delta(X_t) = \delta(X_t)\,\mathrm{d}X_t + 2^{-1}\delta'(X_t)\,\mathrm{d}t \\ \Leftrightarrow\quad &\delta(X_t)\,\mathrm{d}X_t = \mathrm{d}\Delta(X_t) - 2^{-1}\delta'(X_t)\,\mathrm{d}t \\ \Leftrightarrow\quad &\int_0^\omega \delta(X_t)\,\mathrm{d}X_t = \Delta(X_\omega) - \Delta(x_0) - \int_0^\omega 2^{-1}\delta'(X_t)\,\mathrm{d}t,\end{aligned} \tag{3.38}$$

and obtain the simplified RND

$$\frac{\mathrm{d}\mathbb{X}|x_0}{\mathrm{d}\mathbb{W}|x_0}(x_{(0,\omega]}) = \exp\left[\Delta(x_\omega) - \Delta(x_0) - \int_0^\omega \varphi(x_t)\,\mathrm{d}t\right], \tag{3.39}$$

$$\varphi(a) = 2^{-1}(\delta^2 + \delta')(a). \tag{3.40}$$

We now need to change the dominating measure to $\mathbb{W}|x_{\{0,\omega\}} \times \text{Leb}$, such that the Lebesgue-dominated density $\pi(x_\omega|x_0)$ becomes the marginal. By the definition of conditional probability, we note that

$$\frac{\mathrm{d}\mathbb{X}|x_0}{\mathrm{d}\mathbb{X}|x_{\{0,\omega\}}}(x_{(0,\omega]}) = \pi(x_\omega|x_0), \qquad \frac{\mathrm{d}\mathbb{W}|x_0}{\mathrm{d}\mathbb{W}|x_{\{0,\omega\}}}(x_{(0,\omega]}) = \mathrm{N}\left[x_\omega; x_0, \omega\right], \tag{3.41}$$

and accordingly,

$$\begin{aligned}\pi(x_\omega|x_0)\frac{\mathrm{d}\mathbb{X}|x_{\{0,\omega\}}}{\mathrm{d}\mathbb{W}|x_{\{0,\omega\}}}(x_{(0,\omega)}) &= \mathrm{N}\left[x_\omega; x_0, \omega\right]\frac{\mathrm{d}\mathbb{X}|x_0}{\mathrm{d}\mathbb{W}|x_0}(x_{(0,\omega]}) \\ &= \mathrm{N}\left[x_\omega; x_0, \omega\right]\exp\left[\Delta(x_\omega) - \Delta(x_0) - \int_0^\omega \varphi(x_t)\,\mathrm{d}t\right].\end{aligned} \tag{3.42}$$

In what follows, we define the *complete transition density*

$$\pi(x_{(0,\omega]}|x_0) = \pi(x_\omega|x_0)\frac{\mathrm{d}\mathbb{X}|x_{\{0,\omega\}}}{\mathrm{d}\mathbb{W}|x_{\{0,\omega\}}}(x_{(0,\omega)}) \tag{3.43}$$

as the density of $x_{(0,\omega]}$ with respect to $\mathbb{W}|x_{\{0,\omega\}} \times \text{Leb}$, satisfying

$$\pi(x_\omega|x_0) = \int \pi(x_{(0,\omega]}|x_0)\mathbb{W}_{|x_{\{0,\omega\}}}(\mathrm{d}x_{(0,\omega)}), \tag{3.44}$$

as needed. We obtain the complete transition density $\pi(x_{(0,\omega)}, v_\omega|x_0)$ of the original process by a simple change of variables:

$$\pi(x_{(0,\omega)}, v_\omega|v_0) = |\eta'(v_\omega)|\,\pi(x_{(0,\omega]}|x_0) \tag{3.45}$$

Theorem 10 (Complete transition density of reduced diffusions [29, 18])**.** *Let X be a diffusion process with support $\mathcal{X}$, SDE representation $\mathrm{d}X_t = \delta(X_t)\,\mathrm{d}t + \mathrm{d}W_t$ and induced measure $\mathbb{X}$, meeting the following standing assumptions:*

- *δ is continuously differentiable.*
- *$\mathrm{d}\mathbb{X}|x_0 / \mathrm{d}\mathbb{W}|x_0$ exists, $\mathbb{W}$ being the measure under which X is a Brownian motion. A sufficient condition is* Novikov's condition*, given by*

$$\mathrm{E}\left[\exp\left[\int_0^\omega \delta^2(X_t)\,\mathrm{d}t\right]|x_0\right] < \infty. \qquad (x_0 \in \mathcal{X}, \quad 0 < \omega < \infty) \tag{3.46}$$

Then, there is a density $\pi(x_{(0,\omega)}, x_\omega|x_0)$ with respect to $\mathbb{W}|x_{\{0,\omega\}} \times \text{Leb}$ given by

$$\pi(x_{(0,\omega]}|x_0) = \mathrm{N}\left[x_\omega; x_0, \omega\right]\exp\left[\Delta(x_\omega) - \Delta(x_0) - \int_0^\omega \varphi(x_t)\,\mathrm{d}t\right], \tag{3.47}$$

$$\varphi(a) = 2^{-1}(\delta^2 + \delta')(a), \tag{3.48}$$

$$\Delta(a) = \int \delta(a)\,\mathrm{d}a, \tag{3.49}$$

which satisfies

$$\pi(x_\omega|x_0) = \int \pi(x_{(0,\omega)}, v_\omega|v_0)\mathbb{W}_{|x_{\{0,\omega\}}}(\mathrm{d}x_{(0,\omega)}). \tag{3.50}$$

Corollary 2 (Complete transition density)**.** *Let V be a diffusion process with support $\mathcal{V}$, SDE representation $\mathrm{d}V_t = \mu(V_t)\,\mathrm{d}t + \sigma(V_t)\,\mathrm{d}W_t$ and Lamperti transform η. Assume $X = \eta(V)$ fulfills the standing assumptions of Theorem 10, with necessary conditions including μ once and σ twice continuously differentiable. Moreover, let $\mathbb{W}$ be the measure under which X is a Brownian motion. Then, there is a density $\pi(x_{(0,\omega)}, v_\omega|v_0)$ with respect to $\mathbb{W}|(X_{\{0,\omega\}} = \eta(v_{\{0,\omega\}})) \times \mathrm{Leb}$ given by*

$$\pi(x_{(0,\omega)}, v_\omega|v_0) = |\eta'(v_\omega)|\,\mathrm{N}\left[\eta(v_\omega); \eta(v_0), \omega\right] e^{\Delta\circ\eta(v_\omega) - \Delta\circ\eta(v_0) - \int_0^\omega \varphi(x_t)\,\mathrm{d}t} \tag{3.51}$$

which satisfies

$$\pi(v_\omega|v_0) = \int \pi(x_{(0,\omega)}, v_\omega|v_0)\mathbb{W}_{|X_{\{0,\omega\}}=\eta(v_{\{0,\omega\}})}(\mathrm{d}x_{(0,\omega)}). \tag{3.52}$$

Naturally, the Markov property transfers to the augmented setting, so analogously to Corollary 1, for any partition τ of $[0, \omega]$, the complete likelihood factorizes into a product of complete transition densities:

$$\pi(x_{(0,\omega]}|x_0) = \prod_{(\dot\tau\sim\ddot\tau)\in\tau} \pi(x_{(\dot\tau,\ddot\tau]}|x_{\dot\tau}). \tag{3.53}$$

The complete transition density may be exploited for transition density estimation and diffusion sample path simulation, and is at the heart of all the exact inference techniques in this thesis. We further explore those applications in the following Chapter 4. Since the complete transition density incorporates the intractable functional $\exp\left[-\int_0^\omega \varphi(x_t)\,\mathrm{d}t\right]$, further methodological preliminaries are necessary to arrive at an inference method. These take the form of methods to either simulate from $\pi(v_\omega|v_0)$ or estimate it without bias.

3.4 Alternative Dominating Measures

As an aside, notice that the Girsanov theorem allows us to construct transition densities with respect to other tractable dominating measures, subject to obeying the requirements of the theorem. The Wiener dominating measure is the best choice for a general theory, in that it is absolutely continuous with a large class of both transient and mean-reverting measures, and in that its simulation is particularly well understood. Nevertheless, it can be preferable to express the complete transition density with respect to a measure which has identical support to the target measure. In particular, if $\mathcal{X} \in (0, \infty)$, such as for the *Bessel-3* process with SDE $\mathrm{d}X_t = X_t^{-1}\,\mathrm{d}t + \mathrm{d}W_t$, it is usually more elegant to express the complete transition density with respect to said process. In addition, Bessel bridges can

be constructed in terms of Brownian bridges, so it is possible to adapt the retrospective simulation methods of Chapter 4.

Suppose that $\mathrm{d}X_t = \delta(X_t)\,\mathrm{d}t + \mathrm{d}W_t$ such that $\mathcal{X} \in (0, \infty)$, and assume that $\tilde{\delta}(a) = \delta(a) - a^{-1}$ fulfills the conditions of Theorem 9. Moreover, let $\mathbb{R}$ be the measure under which X is the Bessel process and $\mathrm{B3}\,[x_\omega; x_0, \omega]$ be the associated transition density. Then, the complete transition density with respect to $\mathbb{R}|x_{\{0,\omega\}} \times \mathrm{Leb}$ is given by [17] as

$$\pi(x_{(0,\omega]}|x_0) = \frac{x_\omega}{x_0}\,\mathrm{B3}\,[x_\omega; x_0, \omega] \exp\left[\tilde{\Delta}(x_\omega) - \tilde{\Delta}(x_0) - \int_0^\omega \tilde{\varphi}(x_t)\,\mathrm{d}t\right], \tag{3.54}$$

$$\tilde{\varphi}(a) = 2^{-1}(\tilde{\delta}^2 + \tilde{\delta}')(a), \tag{3.55}$$

$$\tilde{\Delta}(a) = \int \tilde{\delta}(a)\,\mathrm{d}a. \tag{3.56}$$

With some moderate complications, this result could be applied in a similar way as Theorem 10 to construct exact Bayesian inference algorithms, though we do not further pursue this idea here. We also note that [68] investigated the setting of Wright-Fisher diffusions with support on $(0, 1)$, for which they propose an appropriate and tractable dominating measure with bounded support.

3.5 Approximate Simulation and Estimation

The tools developed in the previous section also allow us to present some more details of approximate approaches to addressing the intractable transition density problem. As pointed out in the introduction, a discrete time approximation to V is given by the linearization

$$V_t - V_0 = t\mu(V_0) + \sqrt{t}\sigma(V_0)(W_t - W_0). \qquad (t \in [0, \omega]) \tag{3.57}$$

This is the *Euler-Maruyama approximation* to V. t is the *step size* of the approximation, with smaller step-sizes resulting in a better approximation. This scheme converges both weakly and strongly to V as $t \to 0$ [76]. A higher-order method with better strong convergence rates is given by

$$V_t - V_0 = t\mu(V_0) + \sqrt{t}\sigma(V_0)(W_t - W_0) + \frac{1}{2}\sigma(V_0)\sigma'(V_0)((W_t - W_0)^2 - t), \tag{3.58}$$

which is called the *Milstein approximation* to V. We follow the common practice [40, 114, 3] of approximating the reduced process $X = \eta(V)$ rather than V:

$$X_t - X_0 = t\delta(X_0) + \sqrt{t}(W_t - W_0). \tag{3.59}$$

In this instance, $\sigma' = 0$, and we get the higher-order Milstein approximation for free. Moreover, approximating X sometimes preserves the support of V when approximation of V would not. An example of that is the *logistic growth diffusion* $\mathrm{d}V_t = V_t(\beta(1 -$

$V_t/\kappa)\,\mathrm{d}t + \rho V_t)\,\mathrm{d}W_t$. That diffusion has support on $(0, \infty]$, which could be violated by its discretized version. Conversely, $X = -\rho^{-1}\log V$ has support on $\mathbf{R}$, and any discretized simulation is appropriately mapped back to $(0, \infty]$.

In either case, forward simulation is trivial since only Brownian motion need be simulated, while bridge simulation is more challenging, see e.g. [21]. The approximate transition density is given by

$$\bar{\pi}(x_\omega|x_0) = \mathrm{N}\left[x_\omega; x_0 + \omega\delta(x_0), \omega\right]. \tag{3.60}$$

We apply this scheme to design approximate inference algorithms in Chapters 5 and 6. We can also transfer the logic of data augmentation to the discrete approximation, imputing a skeleton of observations at times in $(0, \omega)$. As the skeleton is refined, the approximation becomes arbitrarily precise.

Theorem 11 (Consistency of Euler approximation)**.** *Let τ be a partition of $[0, \omega]$, and X be a diffusion process with transition density $\pi(x_\omega|x_0)$ and approximate transition density $\bar{\pi}(x_{\ddot{\tau}}|x_{\dot{\tau}})$ as in* (3.60)*. Suppose we fill in τ such that* $\mathrm{mesh}\left[\tau\right] \to 0$*. Then, by the weak convergence property of the Euler scheme, $\bar{\pi}(x_{\tau\setminus\{0\}}|x_0)$ has as its marginal a distribution that converges to $\pi(x_\omega|x_0)$:*

$$\pi(x_\omega|x_0) = \lim_{\mathrm{mesh}[\tau]\to 0} \int \mathrm{d}x_{\tau\setminus\{0,\omega\}} \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \bar{\pi}(x_{\ddot{\tau}}|x_{\dot{\tau}}) \tag{3.61}$$

This property lies at the heart of many simulated maximum likelihood and approximate MCMC algorithms.

4 Retrospective Simulation and Estimation

Following the principle of data augmentation, we have obtained by Theorem 10 an explicit probability model for a diffusion path $X_{(0,\omega]}$ with SDE $\mathrm{d}X_t = \delta(X_t)\,\mathrm{d}t + \mathrm{d}W_t$:

$$\pi(x_{(0,\omega]}|x_0) = \mathrm{N}\left[x_\omega; x_0, \omega\right] \exp\left[\Delta(x_\omega) - \Delta(x_0) - \int_0^\omega \varphi(x_t)\,\mathrm{d}t\right], \tag{4.1}$$

$$\varphi(a) = 2^{-1}(\delta^2 + \delta')(a), \tag{4.2}$$

$$\Delta(a) = \int \delta(a)\,\mathrm{d}a, \tag{4.3}$$

where $\pi(x_{(0,\omega]}|x_0)$ is a density with respect to $\mathbb{W}|x_{\{0,\omega\}} \times \mathrm{Leb}$, and $\mathbb{W}$ is the measure under which X is a Brownian motion. We have assumed that δ is continuously differentiable on the support $\mathcal{X}$ and that it satisfies all requirements of Theorem 10. Throughout this chapter, we shall also assume that φ has a uniform lower bound on $\mathcal{X}$:

$$-\infty < \inf_{a \in \mathcal{X}} \varphi(x). \tag{4.4}$$

That assumption is required in order to implement the sample path simulation algorithm in Section 4.1, though it may be relaxed if the goal is merely to carry out Poisson coin simulation or Poisson estimation such as in Subsections 4.1.1 and 4.2.1, which is all that is required to implement the inference methods presented in the following chapters. In that instance, it is sufficient to obtain a bound on φ for some subset of $\mathcal{X}$.

The main impediment to applying this construction to either path simulation or inference lies in the functional $\exp\left[-\int_0^\omega \varphi(x_t)\,\mathrm{d}t\right]$. This is an integral over a nondifferentiable path, preventing exact evaluation. We will address that obstacle with the strategy of *retrospective simulation*, first proposed in [18], which consists of using unbiased estimators of the intractable functional that use finite information, and leaving almost all of the sample paths $x_{(0,\omega]}$ indeterminate until that information is required. The information is then simulated ex post.

We distinguish the case of sample path simulation according to $\pi(x_{(0,\omega]}|x_0)$, and unbiased estimation thereof. Those tasks are intimately related and required to put the methods of Chapter 2 into practice. We address simulation in Section 4.1 and density estimation in Section 4.2. Both raise the need to simulate Brownian bridges given lower or two-sided bounds on the sample path. This is enabled by the EA2 and EA3 algorithms pioneered by [16, 15], presented in Sections 4.3 and 4.4 respectively. We also note the various refinements of those methods discussed in [102], some of which factor into our presentation, as well an extension to jump diffusions.

4.1 Sample Path Simulation

In this section, we consider both forward simulation of $X_{(0,\omega]} \sim \pi(x_{(0,\omega]}|x_0)$ and bridge simulation of $X_{(0,\omega)} \sim \pi(x_{(0,\omega)}|x_{\{0,\omega\}})$, where

$$\pi(x_{(0,\omega)}|x_{\{0,\omega\}}) \stackrel{\text{def}}{=} \frac{\mathrm{d}\mathbb{X}|x_{\{0,\omega\}}}{\mathrm{d}\mathbb{W}|x_{\{0,\omega\}}}(x_{(0,\omega)}) \propto \exp\left[-\int_0^\omega \varphi(x_t)\,\mathrm{d}t\right]. \tag{4.5}$$

We will relate the problems of bridge simulation and forward simulation by way of the construction of *biased Brownian motion*. This will allow us to address both problems under a common rejection sampling framework.

Proposition 1 (Biased Brownian motion [18])**.** *Let $e^{\Delta(x_\omega)-(x_\omega-x_0)^2/(2\omega)}$ be integrable, thus normalizable to a density $\kappa(x_\omega|x_0)$, and assume that it has support on $\mathbf{R}$. Recall that $\mathbb{X}$ denotes the diffusion measure induced by X, and $\mathbb{W}$ the measure under which X is a Brownian motion. We construct the* biased Wiener measure $\tilde{\mathbb{W}}|x_0$ *such that $X_\omega \sim \kappa(x_\omega|x_0)$ and $X_{(0,\omega)}|x_\omega \sim \mathbb{W}|x_{\{0,\omega\}}$, i.e. the bridge dynamics are unchanged from the Wiener measure. Then,*

$$\frac{\mathrm{d}\mathbb{W}|x_0}{\mathrm{d}\tilde{\mathbb{W}}|x_0}(x_{(0,\omega]}) = \frac{\mathrm{N}\left[x_\omega; x_0, \omega\right]}{\kappa(x_\omega|x_0)} \propto \frac{e^{-(x_\omega-x_0)^2/(2\omega)}}{e^{\Delta(x_\omega)-(x_\omega-x_0)^2/(2\omega)}} \propto \exp\left[-\Delta(x_\omega)\right], \tag{4.6}$$

$$\frac{\mathrm{d}\mathbb{X}|x_0}{\mathrm{d}\tilde{\mathbb{W}}|x_0}(x_{(0,\omega]}) = \frac{\mathrm{d}\mathbb{X}|x_0}{\mathrm{d}\mathbb{W}|x_0}(x_{(0,\omega]})\frac{\mathrm{d}\mathbb{W}|x_0}{\mathrm{d}\tilde{\mathbb{W}}|x_0}(x_{(0,\omega]}) \propto \exp\left[-\int_0^\omega \varphi(x_t)\,\mathrm{d}t\right]. \tag{4.7}$$

Therefore, up to normalizing constants, the forward measure $\mathbb{X}|x_0$ has the same density with respect to the forward measure $\tilde{\mathbb{W}}|x_0$ as the bridge measure $\mathbb{X}|x_{\{0,\omega\}}$ has with respect to the bridge measure $\mathbb{W}|x_{\{0,\omega\}}$.

Assuming we solve the 1-dimensional sampling problem from $\kappa(x_\omega|x_0)$, it is then easy to generate samples from $\tilde{\mathbb{W}}|x_0$. If we take a rejection sampling view, we accept forward proposals $x_{(0,\omega]}$ from $\tilde{\mathbb{W}}|x_0$ with the same probability as bridge proposals $x_{(0,\omega)}$ from $\mathbb{W}|x_{\{0,\omega\}}$. If φ is lower bounded on $\mathcal{X}$ by $\varphi^\downarrow$, $\mathrm{d}\mathbb{X}|x_{\{0,\omega\}}/\,\mathrm{d}\mathbb{W}|x_{\{0,\omega\}}$ is bounded above by $e^{-\varphi^\downarrow\omega}$ (up to normalizing constants), and the acceptance probability of the corresponding rejection sampler is given by

$$\exp\left[\int_0^\omega (\varphi^\downarrow - \varphi(x_t))\,\mathrm{d}t\right] \leq 1. \tag{4.8}$$

Accordingly, both forward sampling and bridge sampling are largely reducible to flipping a coin with probability $\exp\left[\int_0^\omega (f^\downarrow - f_t)\,\mathrm{d}t\right]$ for some lower bounded path f. We address that task in the following Section 4.1.1.

4.1.1 Poisson Coin

This section introduces the *Poisson coin* algorithm of [18], which lies at the heart of the exact algorithms for diffusion simulation. It addresses the task of simulating coins with probability $\exp\left[\int_0^\omega (f^\downarrow - f_t)\,\mathrm{d}t\right]$ for $f^\downarrow \leq f_t$, which is intractable for nondifferentiable f. Notice that we will need an upper bound $f^\uparrow \geq f_t$ to implement the Poisson coin algorithm. The essential insight is that if we can construct a tractable event E such that $\Pr[E] = p$, evaluation of p is not necessary to flipping p-coins if we can assess E. In what follows, we require the notion of a *Poisson process*, which may be constructed as follows:

Definition 2 (Homogeneous Poisson Process)**.** *Consider the point process Ψ on $\mathbf{R}^d$. Suppose that for every bounded set $B \in \mathbf{R}^d$,*

$$|\Psi \cap B| \sim \mathrm{Pois}\left[\lambda \times \mathrm{Vol}\left[B\right]\right], \tag{4.9}$$

where λ is called the rate of the Poisson process. Therefore, the expected number of points in B is proportional to the volume of B. Moreover, assume that for a collection of disjoint sets $\{B_i\}$, the cardinalities $|\Psi \cap B_i|$ are independent. Then, Ψ is a (homogeneuous) rate λ Poisson process. Notice that we may apply this definition recursively, i.e. $\Psi \cap B$ is again a Poisson process, but on B rather than $\mathbf{R}^d$. We use the shorthand $\mathrm{PP}\left[B, \lambda\right]$ for a rate λ Poisson process on B. Finally, notice the property that given $|\Psi \cap B|$, the elements of $\Psi \cap B$ are distributed uniformly and independently on B.

We further define the epigraph of $t \mapsto f_t - f^\downarrow$ as

$$\mathrm{epi}\left[f_t - f^\downarrow\right] = \left\{(t, a) \in [0, \omega] \times [0, \infty) : a \leq f_t - f^\downarrow\right\}, \tag{4.10}$$

and notice that is has area $\int_0^\omega (f^\downarrow - f_t)\,\mathrm{d}t$. Furthermore, let $\Psi \sim \mathrm{PP}\left[[0, \omega] \times [0, f^\uparrow - f^\downarrow], 1\right]$, and notice that since $\mathrm{epi}\left[f_t - f^\downarrow\right] \subset [0, \omega] \times [0, f^\uparrow - f^\downarrow]$, the intersection $\mathrm{epi}\left[f_t - f^\downarrow\right] \cap \Psi$ is a unit rate Poisson process on the epigraph. Finally, we note that the distribution of the cardinality of a Poisson process follows a Poisson law with rate equal to the measure of its domain. Therefore,

$$\left|\mathrm{epi}\left[f_t - f^\downarrow\right] \cap \Psi\right| \sim \mathrm{Pois}\left[\int_0^\omega (f_t - f^\downarrow)\,\mathrm{d}t\right], \tag{4.11}$$

$$\Pr\left[\left|\mathrm{epi}\left[f_t - f^\downarrow\right] \cap \Psi\right| = 0\right] = \exp\left[\int_0^\omega (f^\downarrow - f_t)\,\mathrm{d}t\right], \tag{4.12}$$

i.e. $\left\{\left|(\mathrm{epi}\, f_t) \cap \Psi\right| = 0\right\}$ is an event of appropriate probability. We can assess the event by observing that

$$\left\{\left|\mathrm{epi}\left[f_t - f^\downarrow\right] \cap \Psi\right| = 0\right\} = \bigcap_{(T,A)\in\Psi} \left\{A > f_T - f^\downarrow\right\}, \qquad (|\Phi| < \infty \text{ a.s.}), \tag{4.13}$$

so ascertaining the value of the event merely requires evaluating f at a finite number of locations.

Theorem 12 (Poisson coin [18]). *Let f be a bounded, continuous function mapping $[0,\omega] \mapsto [f^\downarrow, f^\uparrow]$, and $\Psi \sim \mathrm{PP}\left[[0,\omega] \times [0, f^\uparrow - f^\downarrow], 1\right]$. Then*

$$\Pr\left[\left|\mathrm{epi}\left[f_t - f^\downarrow\right] \cap \Psi\right| = 0\right] = \exp\left[\int_0^\omega (f^\downarrow - f_t)\,\mathrm{d}t\right]. \tag{4.14}$$

Corollary 3 (Complexity of Poisson coin). *The average cost of simulating a variate from Ψ is $\omega(f^\uparrow - f^\downarrow)$, while the acceptance probability is $\mathcal{O}(e^{\omega(f^\downarrow - f^\uparrow)})$, which gives an upper bounded on the expected runtime until the first acceptance of $\mathcal{O}(e^{\omega(f^\uparrow - f^\downarrow)})$.*

4.1.2 Exact Algorithm

We now apply the Poisson coin algorithm to the rejection sampling problem for diffusion bridges $X_{(0,\omega)}$. For a given bridge path $x_{(0,\omega)}$, the Poisson coin heads event corresponds to

$$\left\{\left|\mathrm{epi}\left[\varphi - \varphi^\downarrow\right] \cap \Psi\right| = 0\right\} = \bigcap_{(T,\Phi)\in\Psi} \left\{\Phi > \varphi(x_T) - \varphi^\downarrow\right\}. \tag{4.15}$$

For models where $\varphi(x)$ is uniformly bounded above in $\mathcal{X}$ by $\varphi^\uparrow$, simulation of Φ can proceed without any inspection of $x_{(0,\omega)}$. We class this class of models $\mathcal{D}_1$.

Definition 3 ($\mathcal{D}_1$-class). *Let X be a diffusion process meeting the standing assumptions of Theorem 10, and*

$$\varphi(a) \leq \sup_{a\in\mathcal{X}} \varphi(a) < \infty. \qquad (a \in \mathcal{X}) \tag{4.16}$$

We call this upper bound $\varphi^\uparrow$, and say that $X \in \mathcal{D}_1$.

Example 9 (Tanh process). *Let X be the Tanh process with SDE $\mathrm{d}X_t = -\beta \tanh\left[\mu - X_t\right]\mathrm{d}t + \mathrm{d}W_t$, $\beta > 0$, inducing subexponential reversion to the stationary mean μ. Then,*

$$\begin{aligned}
\varphi(a) &= 2^{-1}(\beta^2 \tanh\left[\mu - a\right]^2 + \beta(1 - \tanh\left[\mu - a\right]^2)) \qquad (a \in \mathbf{R}) \\
&= 2^{-1}((\beta^2 - \beta)\tanh\left[\mu - a\right]^2 + \beta) \\
&\leq \beta^2/2.
\end{aligned} \tag{4.17}$$

Therefore, $X \in \mathcal{D}_1$.

Given $\{\Psi = \psi\}$, the Poisson coin is then simulated by retrospectively sampling the skeleton $X_{\{t:(t,\phi)\in\psi\}} \sim \mathbb{W}|x_{\{0,\omega\}}$, and then accepting if $\bigcap_{(t,\phi)\in\psi} \left\{\phi > \varphi(x_t) - \varphi^\downarrow\right\}$. See Figure 4.1 for an illustration of this algorithm in action. This algorithm is the original exact algorithm first proposed by [18]. Notice that once x_τ has been accepted as a skeleton from $\mathbb{X}|x_{\{0,\omega\}}$, the path can be filled in at any finite subset of $(0,\omega)$ according to $\mathbb{W}|x_{\tau\cup\{0,\omega\}}$.

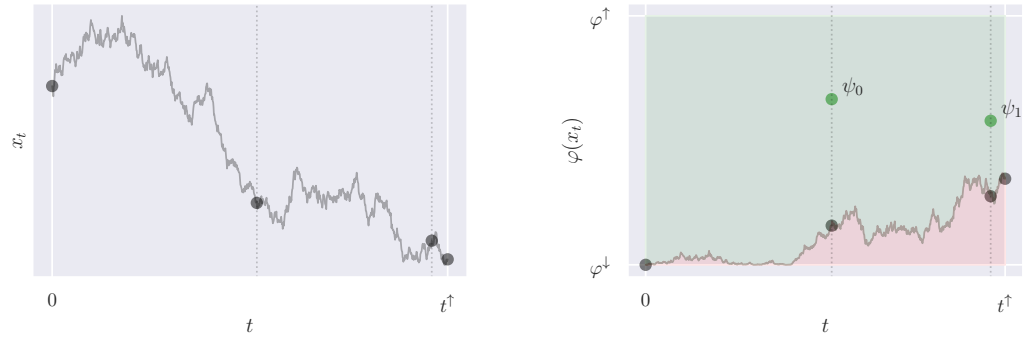


Figure 4.1: Illustration of the exact algorithm. The left panel shows the Brownian bridge skeleton and an implicit sample of the full path $x_{[0,\omega]}$. The right panel shows the corresponding integrand path and a sample of the Poisson process Ψ. epi $\varphi(x_{[0,\omega]})$ is shaded red. Since none of the points fall into epi $\varphi(x_{[0,\omega]})$, the skeleton is accepted.

Algorithm 4 Bridge sampling $X_{(0,\omega)} \sim \pi(x_{(0,\omega)}|x_{\{0,\omega\}})$ for $X \in \mathcal{D}_1$ [18].

repeat
  $\psi \sim \text{PP}\,[[0,\omega] \times [0, \varphi^\uparrow - \varphi^\downarrow], 1]$
  $x_{\{t:(t,\phi)\in\psi\}} \sim \mathbb{W}|x_{\{0,\omega\}}$
until $\bigcap_{(t,\phi)\in\psi} \{\phi > \varphi(x_t) - \varphi^\downarrow\}$

Algorithm 5 Forward sampling $X_{(0,\omega]} \sim \pi(x_{(0,\omega]}|x_0)$ for $X \in \mathcal{D}_1$ [18].

repeat
  $x_\omega \sim \kappa(x_\omega|x_0)$
  $\psi \sim \text{PP}\,[[0,\omega] \times [0, \varphi^\uparrow - \varphi^\downarrow], 1]$
  $x_{\{t:(t,\phi)\in\psi\}} \sim \mathbb{W}|x_{\{0,\omega\}}$
until $\bigcap_{(t,\phi)\in\psi} \{\phi > \varphi(x_t) - \varphi^\downarrow\}$

While this procedure is strikingly elegant, the requirement of a uniform upper bound $\varphi^\uparrow$ is very restrictive, and violated by such standard models as the Vasicek process with SDE $\beta(\mu - V_t)\,\mathrm{d}t + \sigma\,\mathrm{d}W_t$. If we wish to port the logic of the EA algorithm to larger model classes, we must generate additional finite-dimensional information Ξ about $X_{(0,\omega)}$. Given $\{\Xi = \xi\}$, the support of the conditional must be restricted to $\mathcal{X}_\xi \subset \mathcal{X}$ such that φ is upper bounded on that subset, i.e.

$$\mathbb{W}_{|x_{\{0,\omega\}},\xi}[X_{(0,\omega)} \in \mathcal{X}_\xi] = 1, \quad \sup_{a\in\mathcal{X}_\xi} \varphi(a) < \infty. \tag{4.18}$$

We then define

$$\varphi_\xi{}^\uparrow = \sup_{a \in \mathcal{X}_\xi} \varphi(a), \tag{4.19}$$

though of course $\varphi_\xi{}^\uparrow$ may be loose in practice, if the supremum is too difficult to evaluate. To obtain a practical EA algorithm, we further require that:

- Ξ be finite dimensional.
- we can simulate Ξ according to $\mathbb{W}|x_{\{0,\omega\}}$.
- we can simulate a skeleton X_τ according to $\mathbb{W}|(x_{\{0,\omega\}}, \xi)$.

We can then operate an exact algorithm that first generates ξ, computes $\varphi_\xi{}^\uparrow$, generates the Poisson process, and finally the skeleton x_τ. We will distinguish the classes $\mathcal{D}_1 \subset \mathcal{D}_2 \subset \mathcal{D}_3$, each with an associated exact algorithm. Sections 4.3 and 4.4 describe how to carry out those tasks for $\mathcal{D}_2$ and $\mathcal{D}_3$ respectively.

Algorithm 6 Bridge sampling $X_{(0,\omega)} \sim \pi(x_{(0,\omega)}|x_{\{0,\omega\}})$ for $X \notin \mathcal{D}_1$ [16, 15].

repeat
  $\xi \sim \mathbb{W}|x_{\{0,\omega\}}$
  $\psi \sim \mathrm{PP}\left[[0,\omega] \times [0, \varphi_\xi{}^\uparrow - \varphi^\downarrow], 1\right]$
  $x_{\{t:(t,\phi)\in\psi\}} \sim \mathbb{W}|(x_{\{0,\omega\}}, \xi)$
until $\bigcap_{(t,\phi)\in\psi} \{\phi > \varphi(x_t) - \varphi^\downarrow\}$

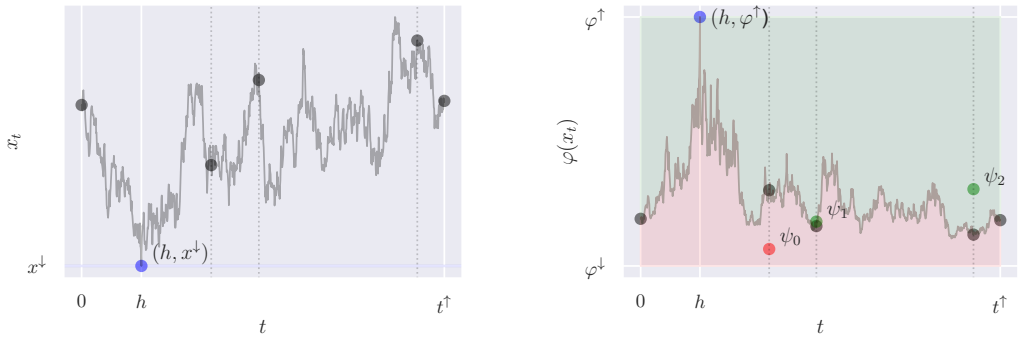


Figure 4.2: Illustration of the Poisson coin algorithm in the EA2 setting. The left panel shows the Brownian bridge skeleton and an implicit sample of the full path $x_{[0,\omega]}$. The right panel shows the corresponding integrand path and a sample of the Poisson process Ψ. epi $\varphi(x_{[0,\omega]})$ is shaded red. Since some of the points fall into epi $\varphi(x_{[0,\omega]})$, the skeleton is rejected. Both panels show the location of the bridge minimum/integrand maximum in blue.

Definition 4 ($\mathcal{D}_2$-class)**.** *Let X be a diffusion process meeting the standing assumptions of Theorem 10, and*

$$\limsup_{a\to\infty}\varphi(a)<\infty \quad \text{or} \quad \limsup_{a\to-\infty}\varphi(a)<\infty. \tag{4.20}$$

Without loss of generality, we focus on the former case, which implies that

$$\sup_{x^{\downarrow}\le a}\varphi(a)<\infty, \qquad (-\infty<x^{\downarrow}) \tag{4.21}$$

i.e. $\varphi(a)$ is bounded above on $[x^{\downarrow},\infty)$. We then say that $X\in\mathcal{D}_2$.

Example 10 (Bessel Process)**.** *Let X be the Bessel process with SDE $\mathrm{d}X_t=\frac{\nu-1}{2X_t}\,\mathrm{d}t+\mathrm{d}W_t$, $\nu>3$. Then,*

$$\begin{aligned}\varphi(a)&=\frac{a^2}{2}\left(\left(\frac{\nu-1}{2}\right)^2+\frac{1-\nu}{2}\right) \qquad (a\in[0,\infty))\\ &=\left(\frac{\nu^2}{8}-\frac{\nu}{2}+\frac{3}{8}\right)/a^2,\end{aligned} \tag{4.22}$$

which is unbounded on $[0,\infty)$, but bounded on $[x^{\downarrow},\infty)$ for a lower bound $x^{\downarrow}$ on X. Thus, $X\in\mathcal{D}_2$. Notice that for $\nu\in(2,3)$ the symmetrical case applies, and φ is bounded on $(-\infty,u]$ for an upper bound u.

Definition 5 ($\mathcal{D}_3$-class)**.** *Let X be a diffusion process meeting the standing assumptions of Theorem 10, and*

$$\sup_{x^{\downarrow}\le a\le x^{\uparrow}}\varphi(a)<\infty, \qquad (-\infty<x^{\downarrow}<x^{\uparrow}<\infty) \tag{4.23}$$

i.e. $\varphi(a)$ is bounded above on $[x^{\downarrow},x^{\uparrow}]$. We then say that $X\in\mathcal{D}_3$.

Example 11 (Vasicek process)**.** *Let X be the Vasicek process with SDE $\mathrm{d}X_t=-\beta(\mu-X_t)\,\mathrm{d}t+\mathrm{d}W_t$, $\beta>0$. Then,*

$$\varphi(a)=2^{-1}(\beta^2(\mu-a)^2+\beta), \qquad (a\in\mathbf{R}) \tag{4.24}$$

which is unbounded on $\mathbf{R}$, but bounded on $[\mu,\varphi(x^{\downarrow})\vee\varphi(x^{\uparrow})]$ for bounds $(x^{\downarrow},x^{\uparrow})$ on X. Thus, $X\in\mathcal{D}_3$.

See Figure 4.2 for an illustration of the exact algorithm for $\mathcal{D}_2$. The successive classes require heavier computational machinery to simulate the skeleton, but they also allow for tighter bounds ${\varphi_\xi}^{\uparrow}$, which reduces the size of the Poisson process and the associated bridge skeleton. We also note that in the case $\mathcal{D}_3$, we can obtain a bound

$${\varphi_\xi}^{\downarrow}=\inf_{a\in\mathcal{X}_\xi}\varphi(a), \tag{4.25}$$

even when $\varphi^{\downarrow}=-\infty$. Such a bound can be applied in the Poisson coin algorithm of Section 4.1.1, and the Poisson estimator of Section 4.2.1.

4.1.3 Batch EA

A rule of thumb when applying EA algorithms is that rejections are more expensive than acceptances. This is due to a generally positive correlation between ${\varphi_\xi}^\uparrow$ and $\int_0^\omega \varphi(x_t)\,\mathrm{d}t$. In addition, if $\int_0^\omega \varphi(x_t)\,\mathrm{d}t$ is large, many rejections might precede an acceptance. As a consequence, it is usually desirable to trade off much faster rejections for slightly slower acceptances.

One option consists of partitioning Ψ vertically into n independent Poisson processes:

$$\Psi = \bigcup_{i=1}^{n} \Psi_i, \quad \Psi_i \sim \mathrm{PP}\left[[0,\omega] \times [((i-1)/n)({\varphi_\xi}^\uparrow - \varphi^\downarrow), (i/n)({\varphi_\xi}^\uparrow - \varphi^\downarrow)], 1\right]. \tag{4.26}$$

We then sequentially evaluate

$$\bigcap_{(t,\phi)\in\psi_i} \left\{\phi > \varphi(x_t) - \varphi^\downarrow\right\}, \qquad (i = 1, \dots, n) \tag{4.27}$$

and reject as soon as the first event evaluates to false. Since we're simulating Ψ from the bottom, where most of the epigraph lies, we are likely to reject early on in the loop. In the n-limit of infinitely small partitions, we may even simulate Ψ pointwise from the bottom up, by first ascertaining the size of the Poisson process

$$|\Psi| \sim \mathrm{Pois}\left[\omega({\varphi_\xi}^\uparrow - \varphi^\downarrow)\right], \tag{4.28}$$

and then, exploiting the properties of Uniform distribution order statistics, sequentially sampling the point process according to

$$t_j \sim \mathrm{Unif}\left[0,\omega\right], \quad \frac{\varphi_j - \varphi_{j-1}}{{\varphi_\xi}^\uparrow - \varphi^\downarrow} \sim \mathrm{Beta}\left[1, |\psi|\right]. \qquad (\varphi_0 = 0, \quad j = 1, \dots, |\psi|) \tag{4.29}$$

We apply this technique to all Bernoulli MCMC algorithms in Chapters 5 and 6, and since the time to generate a rejection is essentially constant in $\omega({\varphi_\xi}^\uparrow - \varphi^\downarrow)$, the average iteration time is substantially lowered.

4.2 Transition Density Estimation

The natural companion problem to simulating from $\pi(x_{(0,\omega]}|x_0)$ is to estimate the complete transition density itself, without bias. Such a result is a stepping stone in deriving the auxiliary algorithms in Chapters 5 and 6. In fact, given an unbiased estimator of $\exp\left[-\int_0^\omega \varphi(x_t)\,\mathrm{d}t\right]$, it directly follows that we can estimate $\pi(x_{(0,\omega]}|x_0)$ without bias as well. We present such an estimator in Section 4.2.1. Section 4.2.2 shows how to apply the result to obtain a model $\pi(\psi, x_{(0,\omega]}|x_0)$ with marginal $\pi(x_{(0,\omega]}|x_0)$.

4.2.1 Poisson Estimator

The *Poisson estimator*, originally proposed in [17] and extended in [39], is an unbiased estimator of functionals of form $\exp\left[\int_0^\omega (f^\downarrow - f_t)\,\mathrm{d}t\right]$ for $f^\downarrow \leq f_t \leq f^\uparrow$. Our jumping off point is the following representation of the exponentiated path integral as an expectation over a Poisson variate.

Lemma 1 (Poisson representation of exponentiated path integral [17])**.** *Let $N \sim \mathrm{Pois}\,[\beta\omega]$, $\beta > 0$. The exponentiated path integral has the following probabilistic representation:*

$$\exp\left[\int_0^\omega (f^\downarrow - f_t)\,\mathrm{d}t\right] = e^{(\beta + f^\downarrow - f^\uparrow)\omega}\,\mathrm{E}\left[\left(\int_0^\omega \frac{f^\uparrow - f_t}{\beta\omega}\,\mathrm{d}t\right)^N\right] \tag{4.30}$$

We may interpret the integrand in the expectation as an estimator that picks a random term from the Taylor expansion of the exponentiated path integral.

Proof. We Taylor expand the expression, and rewrite it as an expectation with respect to N:

$$\begin{aligned}
\exp\left[\int_0^\omega (f^\downarrow - f_t)\,\mathrm{d}t\right] &= e^{(f^\downarrow - f^\uparrow)\omega}\exp\left[\beta\omega\int_0^\omega \frac{f^\uparrow - f_t}{\beta\omega}\,\mathrm{d}t\right] \\
&= e^{(f^\downarrow - f^\uparrow)\omega}\sum_{i=0}^\infty \frac{(\beta\omega)^i}{i!}\left(\int_0^\omega \frac{f^\uparrow - f_t}{\beta\omega}\,\mathrm{d}t\right)^i \\
&= e^{(\beta + f^\downarrow - f^\uparrow)\omega}\sum_{i=0}^\infty \mathrm{Pois}\,[i;\beta\omega]\left(\int_0^\omega \frac{f^\uparrow - f_t}{\beta\omega}\,\mathrm{d}t\right)^i \\
&= e^{(\beta + f^\downarrow - f^\uparrow)\omega}\,\mathrm{E}\left[\left(\int_0^\omega \frac{f^\uparrow - f_t}{\beta\omega}\,\mathrm{d}t\right)^N\right],
\end{aligned} \tag{4.31}$$

□

We now proceed to rewriting the representation as an integral with respect to a Poisson process. Since the path integrals are equivalent to an evaluation of the integrand at a random time, we obtain

$$\begin{aligned}
\exp\left[\int_0^\omega (f^\downarrow - f_t)\,\mathrm{d}t\right] &= e^{(\beta + f^\downarrow - f^\uparrow)\omega}\,\mathrm{E}\left[\left(\frac{f^\uparrow - \omega^{-1}\int_0^\omega f_t\,\mathrm{d}t}{\beta}\right)^N\right] \\
&= e^{(\beta + f^\downarrow - f^\uparrow)\omega}\,\mathrm{E}\left[\prod_{i=1}^N \frac{f^\uparrow - \mathrm{E}\left[f_{T_i}\right]}{\beta}\right] \qquad (T_i \sim \mathrm{Unif}\,[0,\omega]) \\
&= e^{(\beta + f^\downarrow - f^\uparrow)\omega}\,\mathrm{E}\left[\prod_{T\in\Psi} \frac{f^\uparrow - f_T}{\beta}\right]. \qquad (\Psi \sim \mathrm{PP}\,[[0,\omega],\beta])
\end{aligned} \tag{4.32}$$

Accordingly,

$$\bar{P}_\beta = e^{(\beta+f^\downarrow-f^\uparrow)\omega} \prod_{T\in\Psi} \frac{f^\uparrow - f_T}{\beta}, \quad \Psi \sim \mathrm{PP}\left[[0,\omega],\beta\right], \tag{4.33}$$

is an unbiased estimator of the path functional. A natural tuning of β is to set it to $f^\uparrow - f^\downarrow$. [39] call this the *Generalized Poisson estimator-1*:

$$\bar{P}_{f^\uparrow-f^\downarrow} = \prod_{T\in\Psi} \frac{f^\uparrow - f_T}{f^\uparrow - f^\downarrow} \tag{4.34}$$

This is the formulation that we will typically abide by because it guarantees $\bar{P}_{f^\uparrow-f^\downarrow} \in [0,1]$, i.e. the estimate is itself a probability. We refer the reader to [39] for a discussion of estimation variance and alternative tunings of β.

Theorem 13 (Generalized Poisson estimator [17])**.** *Let f be a bounded, continuous function mapping $[0,\omega] \mapsto [f^\downarrow, f^\uparrow]$, and $\Psi \sim \mathrm{PP}\left[[0,\omega], f^\uparrow - f^\downarrow\right]$. Then,*

$$\exp\left[\int_0^\omega (f^\downarrow - f_t)\,\mathrm{d}t\right] = \mathrm{E}\left[\prod_{T\in\Psi} \frac{f^\uparrow - f_T}{f^\uparrow - f^\downarrow}\right], \tag{4.35}$$

and $\prod_{T\in\Psi} \frac{f^\uparrow - f_T}{f^\uparrow - f^\downarrow} \in [0,1]$ is an unbiased estimator of $\exp\left[\int_0^\omega (f^\downarrow - f_t)\,\mathrm{d}t\right]$. Furthermore, its second moment is given by

$$\exp\left[\int_0^\omega \frac{(f^\uparrow - f_t)^2}{f^\uparrow - f^\downarrow}\,\mathrm{d}t - (f^\uparrow - f^\downarrow)\omega\right]. \tag{4.36}$$

Corollary 4 (Complexity of Poisson coin algorithm)**.** *Poisson estimation has cost $\mathcal{O}(\omega(f^\uparrow - f^\downarrow))$, which is more expensive than a rejection with batch EA, identical in cost to an acceptance, and cheaper than simulating to the first acceptance.*

4.2.2 Auxiliary Transition Density

As seen in [17], an immediate application of the Poisson estimator is to estimate the complete transition density without bias. Let $\Psi \sim \mathrm{PP}\left[[0,\omega], \varphi^\uparrow - \varphi^\downarrow\right]$, $\mathbb{P}$ its induced measure, and define the *auxiliary transition density*

$$\pi(\psi, x_{(0,\omega]}|x_0) = \mathrm{N}\left[x_\omega; x_0, \omega\right] e^{\Delta(x_\omega)-\Delta(x_0)-\varphi^\downarrow\omega} \prod_{t\in\psi} \frac{\varphi^\uparrow - \varphi(x_t)}{\varphi^\uparrow - \varphi^\downarrow} \tag{4.37}$$

with dominating measure $\mathbb{P} \times \mathbb{W}|x_{\{0,\omega\}} \times \mathrm{Leb}$. Then, by the Poisson estimator given in Theorem 13, we find that the auxiliary transition density has the complete transition

density as its marginal:

$$\begin{aligned}\int \pi(\psi, x_{(0,\omega]}|x_0)\mathbb{P}(\mathrm{d}\psi) &= \mathrm{N}\left[x_\omega; x_0, \omega\right] e^{\Delta(x_\omega)-\Delta(x_0)-\varphi^\downarrow\omega} \int \prod_{t\in\psi} \frac{\varphi^\uparrow - \varphi(x_t)}{\varphi^\uparrow - \varphi^\downarrow}\mathbb{P}(\mathrm{d}\psi) \\ &= \mathrm{N}\left[x_\omega; x_0, \omega\right] e^{\Delta(x_\omega)-\Delta(x_0)-\varphi^\downarrow\omega}\, \mathrm{E}\left[\prod_{T\in\Psi} \frac{\varphi^\uparrow - \varphi(x_T)}{\varphi^\uparrow - \varphi^\downarrow}\right] \qquad (4.38) \\ &= \mathrm{N}\left[x_\omega; x_0, \omega\right] \exp\left[\Delta(x_\omega) - \Delta(x_0) - \int_0^\omega \varphi(x_t)\,\mathrm{d}t\right]\end{aligned}$$

Since $\pi(\psi, x_{(0,\omega]}|x_0)$ can be evaluated from finite information, we can leverage it to construct a conventional auxiliary MCMC algorithm. We follow this approach in the auxiliary algorithms of Chapters 5 and 6.

Theorem 14 (Auxiliary transition density [17])**.** *Let V be a diffusion process with Lamperti transform η and $X = \eta(V)$ having SDE representation $\mathrm{d}X_t = \delta(X_t)\,\mathrm{d}t + \mathrm{d}W_t$. Assume that X fulfills the assumptions of Theorem 10. Moreover, let $\delta^2 + \delta'$ be bounded. Finally, define $\mathbb{W}$ as the measure under which X is a Brownian motion, and $\mathbb{P}$ as the measure induced by the unit rate Poisson process Ψ on $[0, \omega]$. Then, there is a density $\pi(\psi, x_{(0,\omega)}, v_\omega|v_0)$ with respect to $\mathbb{P} \times \mathbb{W}|(X_{\{0,\omega\}} = \eta(v_{\{0,\omega\}})) \times \mathrm{Leb}$, given by*

$$\begin{aligned}\pi(\psi_{(0,\omega)}, x_{(0,\omega)}, v_\omega|v_0) &= |\eta'(v_\omega)|\,\mathrm{N}\left[\eta(v_\omega); \eta(v_0), \omega - 0\right] e^{\Delta\circ\eta(v_\omega)-\Delta\circ\eta(v_0)-\varphi^\downarrow\omega} \\ &\quad\times \prod_{t\in\psi} \frac{\varphi^\uparrow - \varphi(x_t)}{\varphi^\uparrow - \varphi^\downarrow}, \qquad (4.39)\end{aligned}$$

$$\varphi(a) = 2^{-1}(\delta^2 + \delta')(a), \qquad (4.40)$$

$$\Delta(a) = \int \delta(a)\,\mathrm{d}a, \qquad (4.41)$$

which satisfies

$$\pi(v_\omega|v_0) = \int \pi(\psi, x_{(0,\omega)}, v_\omega|v_0)\mathbb{P}(\mathrm{d}\psi)\mathbb{W}_{|X_{\{0,\omega\}}=\eta(v_{\{0,\omega\}})}(\mathrm{d}x_{(0,\omega)}). \qquad (4.42)$$

Corollary 5 (Transition density estimator)**.**

$$\pi(\Psi, X_{(0,\omega)}, v_\omega|v_0), \quad \Psi \sim \mathbb{P}, \quad X_{(0,\omega)} \sim \mathbb{W}|(X_{\{0,\omega\}} = \eta(v_{\{0,\omega\}})) \qquad (4.43)$$

is an unbiased estimator of $\pi(v_\omega|v_0)$, only requiring evaluations of the Brownian bridge $X_{(0,\omega)}$ at a finite number of times given by the Poisson estimator.

This is analogous to the Exact algorithm, and carries the same implication for extending the estimator to models in $\mathcal{D}_2$ or $\mathcal{D}_3$ for which φ is not uniformly bounded on $\mathbf{R}$. As before, we will appropriately condition $\mathbb{W}|(X_{\{\dot\tau,\ddot\tau\}} = \eta(v_{\{\dot\tau,\ddot\tau\}}))$ on information $\{\Xi = \xi\}$, given which φ is almost surely bounded at ${\varphi_\xi}^\downarrow$ and ${\varphi_\xi}^\uparrow$, and interpolate $X_{(\dot\tau,\ddot\tau)}$ at the required times such that ξ is conserved. Sections 4.3 and 4.4 describe how to carry out those tasks for $\mathcal{D}_2$ and $\mathcal{D}_3$, respectively.

4.3 Simulation of Lower Bounded Brownian Bridges (EA2)

In this section, we consider the class of processes for which

$$\sup_{x^{\downarrow} \leq a} \varphi(a) < \infty, \qquad (-\infty < x^{\downarrow}) \tag{4.44}$$

i.e. processes for which we can upper bound φ given a lower bound on the sample path. W is a Brownian motion, and $\mathbb{W}$ is its induced measure. The direct approach is to simulate the minimum $\check{W} = \min W_{[0,\omega]}$ of a Brownian bridge $W_{(0,\omega)} \sim \mathbb{W}|w_{\{0,\omega\}}$. As a matter of fact, we will devise a sampler for the joint law of the minimum $\check{W}$ and its first passage time, i.e.

$$H = \min_{t \in (0,\omega)} \left\{ t : W_t = \check{W} \right\}. \tag{4.45}$$

Therefore, the full conditioning set is given by

$$\Xi = (H, \check{W}), \tag{4.46}$$

and once a sample $(h, \check{w})$ as been obtained,

$$\sup_{t \in [0,\omega]} \varphi(W_t) = \sup_{h \leq a} \varphi(a) < \infty \tag{4.47}$$

almost surely. In addition, we need to be able to simulate skeletons from the lower bounded Brownian bridge measure $\mathbb{W}|(w_{\{0,\omega\}}, h, \check{w})$. Section 4.3.1 addresses the first and 4.3.2 the second task. For brevity, we introduce the notation $\mathbb{B} = \mathbb{W}|w_{\{0,\omega\}}$ for the Bridge measure and $\mathbb{L} = \mathbb{W}|(w_{\{0,\omega\}}, h, \check{w})$ for the lower bounded bridge measure.

4.3.1 Simulating the Brownian Bridge Minimum

We directly state the relevant result on the joint density of $(H, \check{W})$.

Proposition 2 (Joint density of Brownian bridge minimum $(H, \check{W})$ [71])**.** *Let W be a $w_0 \to w_\omega$ Brownian bridge. Then the minimum $\check{W}$ and its hitting time $H = \min_{t \in (0,\omega)} \left\{ t : W_t = \check{W} \right\}$ follow the density*

$$\pi(h, \check{w}|w_{\{0,\omega\}}) \propto \frac{(\check{w} - w_0)(\check{w} - w_\omega)}{h^3(\omega - h)^3} \exp\left[-\frac{(\check{w} - w_0)^2}{2h} - \frac{(\check{w} - w_\omega)^2}{2(\omega - h)} \right], \qquad (h \in (0,\omega), \quad \check{w} < w_0 \wedge w_\omega) \tag{4.48}$$

and $\check{W}$ has marginal CDF

$$\mathbb{B}[\check{W} \leq \check{w}] = \exp\left[-2\omega(\check{w} - w_0)(\check{w} - w_\omega) \right]. \qquad (\check{w} < w_0 \wedge w_\omega) \tag{4.49}$$

Since the marginal CDF is easily inverted, we obtain a sample from $\pi(\breve{w}|w_{\{0,\omega\}})$ by way of the inverse transform method.

Algorithm 7 Sampling $\breve{W} \sim \mathbb{B}$ [71, Chapter 2].

$u \sim \text{Uniform}\,[0,1]$
$\breve{w} \leftarrow \left((w_0 + w_\omega) - \sqrt{(w_0 - w_\omega)^2 - 2\omega \log u}\right)/2$

Sampling H given $\breve{w}$ is more involved, but [31, Chapter IV] provides an algorithm. Define the Wald or Inverse Gaussian distribution by way of the following density:

$$\pi(a|\mu,\lambda) = \sqrt{\frac{\lambda}{2\pi a^3}} \exp\left[-\frac{\lambda(a-\mu)^2}{2\mu^2 a}\right] \qquad (a > 0) \tag{4.50}$$

On the basis of that definition, a sample from $\pi(h|w_{\{0,\omega\}}, \breve{w})$ may be obtained by the algorithm below.

Algorithm 8 Sampling $H \sim \mathbb{B}|\breve{w}$ [31, Chapter IV].

$u \sim \text{Uniform}\,[0,1]$
$c_1 \leftarrow (w_\omega - \breve{w})^2/(2\omega), \quad c_2 \leftarrow (\breve{w} - w_0)^2/(2\omega), \quad c_3 \leftarrow \sqrt{c_1/c_2}$
if $u < 1/(1+c_3)$ **then**
  $a \sim \text{Wald}\,[c_3, 2c_1]$
  $h \leftarrow \omega/(1+a)$
else
  $a \sim \text{Wald}\,[c_3^{-1}, 2c_2]$
  $h \leftarrow \omega/(1+a^{-1})$

4.3.2 Filling in the Lower Bounded Bridge

Next, we consider the task of sampling the lower bounded bridge at a finite set of times. Suppose, for brevity of notation, that only $w_{\{0,\omega\}}$ have been revealed, and $W_{(0,\omega)}$ follows $\mathbb{L}$, conditional on the minimum. Given $(h, \breve{w})$, the key result due to [6] is that the trajectories either side of the minimum can be represented as independent bridges of the Bessel-3 processes. The Bessel-3 process solves the SDE

$$\mathrm{d}R_t = R_t^{-1}\,\mathrm{d}t + \mathrm{d}W_t, \tag{4.51}$$

and we use the symbol $\mathbb{R}$ to refer to its induced measure. Notice that a Bessel process can be represented as a Wiener process conditioned on remaining positive.

Proposition 3 (Bessel decomposition of Brownian bridges [6])**.** *Let W be a $w_0 \to w_\omega$ Brownian bridge. Given its minimum and associated hitting time $(h, \breve{w})$, $W_{(0,h)}$ and*

$W_{(h,\omega)}$ are independent, and their transformations

$$\overleftarrow{R}_u = W_{h-u} - \check{w}, \qquad (u \in (0,h)) \tag{4.52}$$

$$\vec{R}_u = W_{h+u} - \check{w}, \qquad (u \in (0,\omega-h)) \tag{4.53}$$

are independent $0 \to w_0 - \check{w}$ and $0 \to w_\omega - \check{w}$ Bessel-3 bridges, respectively. Conversely,

$$W_t = \begin{cases} \check{w} + \overleftarrow{R}_{h-t} & (t \in (0,h)) \\ \check{w} + \vec{R}_{t-h} & (t \in (h,\omega)) \end{cases} \tag{4.54}$$

follows $\mathbb{L}$.

Algorithm 9 Sampling $W_t \sim \mathbb{L}$ on $t \in (0,\omega)$[6].

$(u, \hat{u}, r_{\hat{u}}) \leftarrow \begin{cases} (h-t, h, w_0 - \check{w}) & (t \in (0,h)) \\ (t-h, \omega-h, w_\omega - \check{w}) & (t \in (h,\omega)) \end{cases}$
$r_u \sim \mathbb{R}|r_{\hat{u}}$ according to (ALG 10)
$w_t \leftarrow r_u + \check{w}$

The Bessel process may also be constructed by taking the Euclidean norm of a 3-dimensional Brownian motion vector, i.e.

$$W^{(n)} \sim \mathbb{W}|(W^{(n)}=0) \quad \Rightarrow \quad R = \sqrt{\sum_{i=1}^{3}(W^{(n)})^2} \sim \mathbb{R}|(R_0=0). \tag{4.55}$$

In a similar way, [13] show how to construct its bridges starting from 0 by adding up multiple Brownian bridges. Given the Bessel bridge simulator 10, we sample skeletons from $W_{(0,\omega)} \sim \mathbb{L}$ according to Algorithm 9.

Algorithm 10 Sampling $R_u \sim \mathbb{R}|(R_0=0, R_{\hat{u}}=r_{\hat{u}})$ on $u \in (0,\hat{u})$ [13].

$b_u^{(n)} \sim \mathbb{W}|(B_0^{(n)} = B_{\hat{u}}^{(n)} = 0) \quad (n=1,2,3)$
$r_u \leftarrow \sqrt{(b_u^{(1)})^2 + (b_u^{(2)})^2 + (b_u^{(3)} + r_{\hat{u}} u/\hat{u})^2}$

We now move on to the setting where in addition to $(h,\check{w})$, $\{W_\tau = w_\tau\}$ has been revealed at a set of times $\tau \subset [0,\omega]$, and therefore the law at any further times follows $\mathbb{L}|w_\tau$. Without loss of generality, suppose that $\tau = \{\dot{\tau}, \ddot{\tau}\} \subset [h,\omega]$, and that we wish to sample according to the conditional law of $W_{(h,\omega)}|(w_{\{\dot{\tau},\ddot{\tau}\}}, h, \check{w})$. The key observation due to [53] is that since we can represent $W_{(h,\omega)}$ as a set of a set of 3 Brownian bridges $\left\{\vec{B}^{(n)} : n=1,2,3\right\}$ with $\vec{B}_0^{(n)} = \vec{B}_{\omega-h}^{(n)} = 0$, we can represent the conditioning on $\{W_\tau = w_\tau\}$ as an event $\left\{\vec{B}^{(n)}_{\{\dot{\tau}-h,\ddot{\tau}-h\}} = \vec{b}^{(n)}_{\nu-h} : n=1,2,3\right\}$. Therefore, a sample from W_ν

for $\nu \subset \{\dot{\tau}, \ddot{\tau}\}$ can be obtained by way of a sample from $\left\{\vec{B}^{(n)}_{\nu-h} : n = 1,2,3\right\}$, where $\vec{B}^{(n)}_{\nu-h}$ follows the Brownian bridge law $\mathbb{W}|(\vec{B}^{(n)}_{\{\dot{\tau}-h,\ddot{\tau}-h\}} = \vec{b}^{(n)}_{\{\dot{\tau}-h,\ddot{\tau}-h\}})$. Accordingly, the most direct way of revealing $W_{(0,\omega)}$ flexibly and iteratively consists of storing its skeleton in terms of the representation

$$\{h, \check{w}\} \cup \left\{\check{B}^{(n)} : n = 1,2,3\right\} \cup \left\{\vec{B}^{(n)} : n = 1,2,3\right\}, \tag{4.56}$$

revealing the Brownian bridges $\check{B}^{(n)}$ and $\vec{B}^{(n)}$ as needed.

4.4 Simulation of Bounded Brownian Bridges (EA3)

In this section, we consider the larger class of processes for which

$$\sup_{x^{\downarrow} \le a \le x^{\uparrow}} \varphi(a) < \infty, \qquad (-\infty < x^{\downarrow} < x^{\uparrow} < \infty) \tag{4.57}$$

i.e. processes for which we can upper bound φ given a bounded sample path. The direct solution would be to sample both $\check{W}$ and $\hat{W}$ according to $\mathbb{B}$, but the joint distribution is intractable. Instead, the approach proposed by [15] is to partition the Brownian bridge measure into the sum of conditional measures confined to increasing subintervals $\ell_1 \subset \ell_2 \subset \cdots \subset \mathbf{R}$, where $\ell_\infty = \mathbf{R}$. We index those intervals by λ, and use the notation $\{\Lambda \le \lambda\} = \{W \subseteq \ell_\lambda\}$ to indicate that W is confined to the λ-th layer. The decomposition is then given by

$$\mathbb{B}(\mathrm{d}w_{(0,\omega)}) = \sum_{\lambda=1}^{\infty} \mathbb{B}[\Lambda = \lambda]\mathbb{B}_{|\lambda}(\mathrm{d}w_{(0,\omega)}), \tag{4.58}$$

where we note that $\{\Lambda = \lambda\} = \{\ell_{\lambda-1} \subset W \subseteq \ell_\lambda\}$.

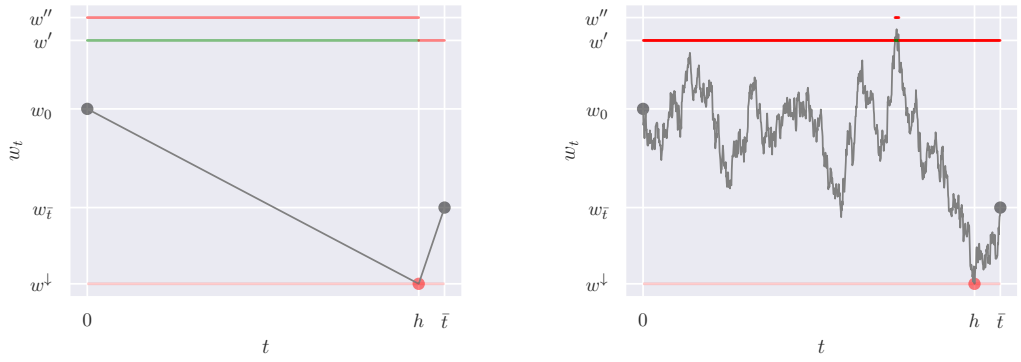


Figure 4.3: Illustration of the minimum skeleton (left), and its refinement after densely interpolating the bridge (right). The green line has to be attained at some point. As we interpolate more finely, we accumulate information on where it is attained.

Because simulation from $\mathbb{B}_\lambda$ turns out to be inefficient, the appropriate conditioning set is more complicated. It consists of the global minimum $(H, \breve{W})$, and a range $[W', W'']$, such that for a sample thereof,

$$\sup_{t\in[0,\omega]} \varphi(W_t) = \sup_{h\le a\le w''} \varphi(a) < \infty. \tag{4.59}$$

There is also a symmetrical case where we record the global maximum and the lower layer. We do not treat that case separately since $\mathbb{B}$ is symmetrical with respect to reflections around the line connecting $(0, w_0)$ and (ω, w_ω). In addition, we will require a set of indicators $\left\{\Gamma_{(0,H)}, \Gamma_{(H,\omega)}\right\}$, showing whether W exceeds W' within the respective time interval. In practice, it is often advantageous to refine the hitting indicators as W is revealed at additional times, such as illustrated in Figure 4.3.

In a first step, we will describe how to initialize Ξ in Section 4.4.2. This will require the flipping of coins with probabilities given as certain kinds of infinite sums, which we discuss in Section 4.4.1. We then describe how to fill in the skeleton given Ξ in Section 4.4.3.

4.4.1 Probabilities as Alternating Cauchy Sequences

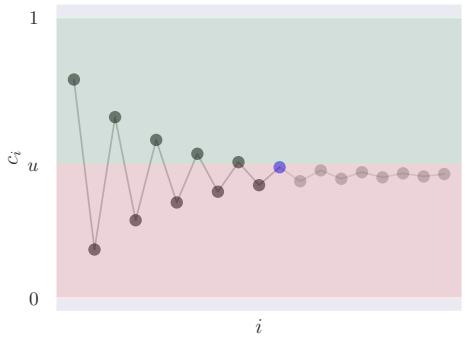


Figure 4.4: Illustration of the alternating Cauchy sequence coin simulation algorithm. The dividing horizontal line is randomly drawn between 0 and 1. If the sequence stabilizes in the green region, the coin comes up heads, and vice versa. The event is determined by the time the sequence reaches the blue element.

An implementation of EA3 requires simulations from coins whose probability is only available as an infinite series, of the form

$$p = \sum_{i=1}^{\infty}(a_i - b_i). \qquad (a_i, b_i > 0, \quad p \in (0,1)) \tag{4.60}$$

Define the sequence c by the relations

$$c_{2i-1} = \sum_{j=1}^{i-1}(a_j - b_j) + a_i, \quad c_{2i} = c_{2i-1} - b_i, \qquad (i = 1, 2, \dots) \tag{4.61}$$

and observe that $\lim_{i\to\infty} c_i = p$. Furthermore, if it satisfies

$$c_{2i} < c_{2i+2} < c_{2i+1} < c_{2i-1} \Rightarrow |c_{i+1} - c_i| < |c_i - c_{i-1}|\,, \qquad (i = 1, 2, \dots) \tag{4.62}$$

it forms an alternating Cauchy sequence. Due to the decaying increments, once two consecutive elements of a sequence fall on a given side of a value, all following elements will as well. See Figure 4.4 for an illustration of the argument. Accordingly, coins of probability p may be simulated according to the event:

$$\{U < p\} = \{\exists i : U < c_{2i-1} \vee c_{2i}\}\,, \quad U \sim \text{Uniform}\,[0, 1]\,. \tag{4.63}$$

The sequences that are used in EA3 converge exponentially fast.

Algorithm 11 Sampling $C \sim \text{Bernoulli}\left[p = \sum_{i=1}^{\infty}(a_i - b_i)\right]$, $a_i, b_i > 0$ [15].

$u \sim \text{Unif}\,[0, 1]$, $i \leftarrow 0$, $d \leftarrow 0$
repeat
  $i \leftarrow i + 1$
  $c \leftarrow d + a_i$
  $d \leftarrow c - b_i$
until $\{c \vee d < u\} \vee \{c \wedge d > u\}$
if $\{c \wedge d < u\}$ **then**
  $c \leftarrow$ `True`
else
  $c \leftarrow$ `False`

4.4.2 Simulating the Brownian Bridge Bounds

We now dispose of the necessary tools simulate Λ. As previously hinted, it will not be sufficient to merely simulate $\{\Lambda = \lambda\}$, in fact, we will need to simulate a richer subset of that event. Nonetheless, we will still $\{\Lambda = \lambda\}$ as a first step, and proceed with the derivation of the algorithm.

Let $\{d_i : i = 1, 2, ...\}$ be a strictly increasing and diverging sequence starting at 0. Moreover, define the increasing sequence of intervals

$$\{\ell_i = [w_0 \wedge w_\omega - d_i, w_0 \vee w_\omega + d_i] : i = 1, 2, ...\}\,, \tag{4.64}$$

covering $\mathbf{R}$. We also define the *layer* i as the smallest index for which W is bounded in ℓ_i:

$$\Lambda = \min\{i : W \subseteq \ell_i\} \tag{4.65}$$

Thus, the CDF of Λ is given by

$$\mathbb{B}[\Lambda \leq \lambda] = \mathbb{B}[W \subseteq \ell_\lambda]. \qquad (\lambda = 1, 2, \dots) \tag{4.66}$$

These containment probabilities are given by [32].

Proposition 4 (Containment probabilities of Brownian bridges [32])**.** *Let W be a $w_0 \to w_\omega$ Brownian bridge. Then, its containment probability is given by the infinite series*

$$\mathbb{B}[W \subseteq [\pm c]] = \sum_{i=1}^{\infty} (a_i - b_i), \qquad (c > |w_0| \vee |w_\omega|) \tag{4.67}$$

$$a_i = \exp\left[-2\omega^{-1}(2ci - c - w_0)(2ci - c - w_\omega)\right], \tag{4.68}$$

$$b_i = \exp\left[-2i\omega^{-1}(4c^2 i + 2c(w_\omega - w_0))\right]. \tag{4.69}$$

In addition, the infinite series is an alternating Cauchy sequence.

Therefore, we may sample from the distribution of Λ by resorting to Algorithm 11.

Algorithm 12 Sampling the layer $\Lambda \sim \mathbb{B}$ [15].

$u \sim \text{Unif}\,[0,1]$, $\lambda \leftarrow 0$
repeat
  $\lambda \leftarrow \lambda + 1$
  $\ell_\lambda = [w_0 \wedge w_\omega - d_\lambda, w_0 \vee w_\omega + d_\lambda]$
  $b \leftarrow \{u < \mathbb{B}[W \subseteq \ell_\lambda]\}$ according to (ALG 11), using the already sampled u
until $b =$ `True`

Given $\{\Lambda = \lambda\}$, the direct approach consists of simulating from $\mathbb{B}|\lambda$ by rejection sampling from $\mathbb{B}$. This is inadvisable since the expected number of attempts N until success corresponds to

$$\mathrm{E}\,[N] = \mathrm{E}\,\mathrm{E}\,[N|\lambda] = \sum_{i=0}^{\infty} \frac{1}{\mathbb{B}[\lambda]} \times \mathbb{B}[\lambda] = \infty, \tag{4.70}$$

which is decidedly unappealing. The better option is to generate a proposal conditioned on its minimum or maximum being located in $\ell_\lambda \setminus \ell_{\lambda-1}$. We use the notation

$$L_\lambda = \left\{\check{W} \in [w_0 \wedge w_\omega - d_\lambda, w_0 \wedge w_\omega - d_{\lambda-1}]\right\}, \tag{4.71}$$

$$U_\lambda = \left\{\hat{W} \in [w_0 \vee w_\omega + d_{\lambda-1}, w_0 \vee w_\omega + d_\lambda]\right\}, \tag{4.72}$$

for the respective events, noting that $L_\lambda \cup U_\lambda \supset \{\Lambda = \lambda\}$. The following proposition establishes that such a proposal results in a valid rejection sampler.

Proposition 5 (Conditioned $\mathbb{B}|\lambda$-proposal [15])**.** *Define the proposal measure*

$$\mathbb{P}_\lambda = \frac{\mathbb{B}|L_\lambda + \mathbb{B}|U_\lambda}{2}. \tag{4.73}$$

$\mathbb{B}|\lambda$ is absolutely continuous with respect to $\mathbb{P}|\lambda$, and proposals from the latter are accepted with probability equal to the Radon-Nikodym derivative

$$\frac{\mathrm{d}\mathbb{B}|\lambda}{\mathrm{d}\mathbb{P}_\lambda}(w_{(0,\omega)}) \propto \frac{1_{\Lambda=\lambda}}{1 + 1_{L_\lambda \cap U_\lambda}} \leq 1. \tag{4.74}$$

Proof.

$$\begin{aligned}
\frac{\mathrm{d}\mathbb{B}|\lambda}{\mathrm{d}\mathbb{P}_\lambda}(w_{(0,\omega)}) &= 2\frac{\mathrm{d}\mathbb{B}|\lambda}{\mathrm{d}(\mathbb{B}|L_\lambda + \mathbb{B}|U_\lambda)}(w_{(0,\omega)}) \\
&= 2\frac{1_{\Lambda=\lambda}/\mathbb{B}[\lambda]}{1_{L_\lambda}/\mathbb{B}[L_\lambda] + 1_{U_\lambda}/\mathbb{B}[U_\lambda]} \\
&= \frac{\mathbb{B}[L_\lambda]}{\mathbb{B}[\lambda]}\frac{1_{\Lambda=\lambda}}{1_{L_\lambda} + 1_{U_\lambda}} \propto \frac{1_{\Lambda=\lambda}}{1 + 1_{L_\lambda \cap U_\lambda}},
\end{aligned} \tag{4.75}$$

where $\mathbb{B}[L_\lambda] = \mathbb{B}[U_\lambda]$ due to the symmetry of the layers, and $1_{L_\lambda} + 1_{U_\lambda} = 1 + 1_{L_\lambda \cap U_\lambda}$ almost surely under $\mathbb{P}_\lambda$. □

Hence, for any proposal from $\mathbb{P}_\lambda$, we need only concern ourselves with the events $\{\Lambda = \lambda\}$ and $L_\lambda \cap U_\lambda$ to accept or reject, and the remainder of the section addresses the simulation of those events. Notice that the expected number of proposals per acceptance is bounded above by

$$\mathrm{E}\,[N|\lambda] = 2\frac{\mathbb{B}[U_\lambda]}{\mathbb{B}[\lambda]} \leq 2\frac{\mathbb{B}[L_\lambda]}{\mathbb{B}[L_\lambda, \lambda]} \leq \frac{2}{\mathbb{B}[\hat{W} < w_0 \vee w_\omega + d_\lambda]}, \tag{4.76}$$

which is small for sufficiently large d_λ. As mentioned before, we may restrict ourselves to proposing from $\mathbb{B}|L_\lambda$ without loss of generality.

To assess the necessary events, we have to ascertain the precise location of $\check{W}$. Given L_λ, $\check{W}$ has the invertible CDF $\mathbb{B}_{|L_\lambda}[\check{W} \leq \check{w}]$, so we may obtain a sample by way of the inverse transform method, just as for Algorithm 7, using the algorithm below.

Algorithm 13 Sampling $\check{W} \sim \mathbb{B}|(\check{W} \in [a, b])$.

$u \sim \mathrm{Uniform}\,[0, 1]$
$\tilde{u} \leftarrow u(e^{-2\omega(b-w_0)(b-w_\omega)} - e^{-2\omega(a-w_0)(a-w_\omega)}) + e^{-2\omega(a-w_0)(a-w_\omega)}$
$\check{w} \leftarrow \left((w_0 + w_\omega) - \sqrt{(w_0 - w_\omega)^2 - 2\omega \log \tilde{u}}\right)/2$

Next, we assess the event $\{\Lambda = \lambda\}$. Recalling that $\mathbb{L} = \mathbb{B}|(h, \check{w})$ for a given minimum proposal $(h, \check{w})$, it has probability

$$\mathbb{L}[\lambda] = \mathbb{L}[\hat{W}_{(0,h)} \in \ell_\lambda]\mathbb{L}[\hat{W}_{(h,\omega)} \in \ell_\lambda], \tag{4.77}$$

where $\mathbb{L}[\hat{W}_{(0,h)} \in \ell_\lambda]$ is the containment probability of a Bessel bridge and given in alternating Cauchy series form in [15].

Algorithm 14 Sampling $\{\Lambda = \lambda\} \sim \mathbb{L}$ [15].

$\ell_\lambda \leftarrow [w_0 \wedge w_\omega - d_\lambda, w_0 \vee w_\omega + d_\lambda]$
$\left\{\hat{W}_{(0,h)} \in \ell_\lambda\right\} \sim \mathbb{L}$ according to (ALG 11), using the series representation in [15]
$\left\{\hat{W}_{(h,\omega)} \in \ell_\lambda\right\} \sim \mathbb{L}$
$\{\Lambda = \lambda\} \leftarrow \left\{\hat{W}_{(0,h)} \in \ell_\lambda\right\} \cup \left\{\hat{W}_{(h,\omega)} \in \ell_\lambda\right\}$

If $\{\Lambda = \lambda\}$ is false, we discard the proposed minimum. Otherwise, we assess U_λ, which is true with probability

$$\mathbb{L}_{|\lambda}[U_\lambda] = 1 - \mathbb{L}_{|\hat{W}_{(0,h)} \in \ell_\lambda}[\hat{W}_{(0,h)} \in \ell_{\lambda-1}]\mathbb{L}_{|\hat{W}_{(h,\omega)} \in \ell_\lambda}[\hat{W}_{(h,\omega)} \in \ell_{\lambda-1}]. \tag{4.78}$$

In keeping with Proposition 5, $(h, \breve{w})$ is accepted with probability $1/2$ if U_λ is true, and probability 1 otherwise. These containment probabilities for upper bounded Bessel bridges are also given in alternating Cauchy series form by [15]. If the proposal is accepted, the hitting indicators follow from the escape events, i.e.

$$\begin{cases} \Gamma_{(0,h)} = 0 & \text{if } \hat{W}_{(0,h)} \in \ell_{\lambda-1} \\ \Gamma_{(h,\omega)} = 1 & \text{otherwise} \end{cases}, \tag{4.79}$$

and likewise for $\Gamma_{(h,\omega)}$.

Algorithm 15 Sampling $U_\lambda \sim \mathbb{L}|\lambda$ [15].

$\ell_\lambda \leftarrow [w_0 \wedge w_\omega - d_\lambda, w_0 \vee w_\omega + d_\lambda]$
$\left\{\hat{W}_{(0,h)} \in \ell_{\lambda-1}\right\} \sim \mathbb{L}|(\hat{W}_{(0,h)} \in \ell_\lambda)$ according to (ALG 11), using the series representation in [15]
$\left\{\hat{W}_{(h,\omega)} \in \ell_{\lambda-1}\right\} \sim \mathbb{L}|(\hat{W}_{(h,\omega)} \in \ell_\lambda)$
$U_\lambda \leftarrow \left\{\hat{W}_{(0,h)} \notin \ell_{\lambda-1}\right\} \cup \left\{\hat{W}_{(h,\omega)} \notin \ell_{\lambda-1}\right\}$

Algorithm 16 Sampling $(H, \breve{W}) \sim \mathbb{B}|(\lambda, L_\lambda)$ [15].

repeat
  $(h, \breve{w}) \sim \mathbb{B}|L_\lambda$ according to (ALG 13)
  $\{\Lambda = \lambda\} \sim \mathbb{L}$ according to (ALG 14)
  if $\{\Lambda = \lambda\}$ **then**
    $U_\lambda \sim \mathbb{L}|\lambda$ according to (ALG 15)
    $u \sim \text{Unif}\,[0, 1]$
until $(\{\Lambda = \lambda\} \wedge (\neg U_\lambda \vee \{u < 0.5\}))$

4.4.3 Filling in the Bounded Bridge

We return to considering the task of sampling the bridge, now bounded on both sides, at a finite set of times. We begin again with the case where only $w_{\{0,\omega\}}$ are known. Define $\mathbb{U} = \mathbb{L}|(\gamma_{(0,h)}, \gamma_{(h,\omega)})$ as the conditional law of $W_{(0,\omega)}$ given ξ. Without loss of generality due to the conditional independence of $W_{(0,h)}$ and $W_{(h,\omega)}$, suppose that we wish to reveal W at $\tau \subset (h,\omega)$. Since $\mathbb{U} \ll \mathbb{L}$, we may obtain samples from $\mathbb{U}$ by way of rejection sampling from $\mathbb{L}$, with Radon-Nikodym derivative

$$\frac{\mathrm{d}\mathbb{U}}{\mathrm{d}\mathbb{L}}(w_{(h,\omega)}) = \frac{\mathrm{d}\mathbb{L}|\gamma_{(h,\omega)}}{\mathrm{d}\mathbb{L}}(w_{(h,\omega)}) \propto 1_{\Gamma_{(h,\omega)}=\gamma_{(h,\omega)}}. \tag{4.80}$$

Accordingly, in analogy with the previous section, we use Algorithm 9 to propose w_τ according to $\mathbb{L}$, and then enforce conformity with $\gamma_{(h,\omega)}$. The corresponding probability $\mathbb{L}_{|w_\tau}[\gamma_{(h,\omega)}]$ is given by

$$\mathbb{L}_{|w_\tau}[\gamma_{(h,\omega)}] = \mathbb{L}_{|w_\tau,\lambda}[\gamma_{(h,\omega)}]\mathbb{L}_{|w_\tau}[\lambda] = \mathbb{L}_{|w_\tau,\lambda}[\gamma_{(h,\omega)}] \prod_{(\dot\tau\sim\ddot\tau)\in\tilde\tau} \mathbb{L}_{|w_{\{\dot\tau,\ddot\tau\}}}[\hat W_{(\dot\tau,\ddot\tau)} \in \ell_\lambda], \tag{4.81}$$

$$\mathbb{L}_{|w_\tau,\lambda}[\gamma_{(h,\omega)}] = \begin{cases} \prod_{(\dot\tau\sim\ddot\tau)\in\tilde\tau} \mathbb{L}_{|w_{\{\dot\tau,\ddot\tau\}},\hat W_{(\dot\tau,\ddot\tau)}\in\ell_\lambda}[\hat W_{(\dot\tau,\ddot\tau)} \in \ell_{\lambda-1}] & (\gamma_{(h,\omega)} = 0) \\ 1 - \prod_{(\dot\tau\sim\ddot\tau)\in\tilde\tau} \mathbb{L}_{|w_{\{\dot\tau,\ddot\tau\}},\hat W_{(\dot\tau,\ddot\tau)}\in\ell_\lambda}[\hat W_{(\dot\tau,\ddot\tau)} \in \ell_{\lambda-1}] & (\gamma_{(h,\omega)} = 1) \end{cases}, \tag{4.82}$$

where $\tilde\tau = \tau \cup \{h,\omega\}$. These probabilities correspond to products of Bessel bridge escape and containment probabilities provided in [15, Section 3.1.2], which are simulated according to Algorithm 11. Also keep in mind that in the $\gamma_{(h,\omega)} = 1$ case, the coin flips according to $\mathbb{L}_{|w_{\{\dot\tau,\ddot\tau\}},\hat W_{(\dot\tau,\ddot\tau)}\in\ell_\lambda}[\hat W_{(\dot\tau,\ddot\tau)} \in \ell_{\lambda-1}]$ refine the indicator $\gamma_{(h,\omega)}$ into a set $\left\{\gamma_{(\dot\tau,\ddot\tau)} : (\dot\tau \sim \ddot\tau) \in \tilde\tau\right\}$, at least one of which has to be 1. It is often preferable to record the refined escape indicators, as that preserves conditional independence of W between the times τ.

If in addition $\{W_\tau = w_\tau\}$ has been revealed at a set of times $\tau \subset [0,\omega]$, we suppose again without loss of generality that $\tau = \{\dot\tau, \ddot\tau\} \subset [h,\omega]$. The sampling algorithm for W_ν with $\nu \subset (\dot\tau, \ddot\tau)$ consists of first generating a proposal w_ν from $\mathbb{L}|w_{\{\dot\tau,\ddot\tau\}}$ as set out in Section 4.3.2, then accepting with probability $\mathbb{L}_{|w_\nu}[\gamma_{(\dot\tau,\ddot\tau)}]$, in the process refining $\gamma_{(\dot\tau,\ddot\tau)}$ into $\left\{\gamma_{(\dot\nu,\ddot\nu)} : (\dot\nu \sim \ddot\nu) \in \nu \cup \{\dot\tau,\ddot\tau\}\right\}$.

4.4.4 Layer Refinement

In the following chapters, a key task will consist of either flipping a coin with probability

$$\exp\left[\int_0^\omega (\varphi_\xi{}^\downarrow - \varphi(x_t))\right] \tag{4.83}$$

for some ξ-dependent bounds $\varphi_\xi{}^\downarrow$ and $\varphi_\xi{}^\uparrow$, or estimating that probability without bias. Other things equal, the computational burden of this task quickly increases in ω. In

particular, as a consequence of Proposition 4, if we wish to keep $\mathbb{B}[X_{(0,\omega)} \in \ell_\lambda]$ constant as ω increases, then $|\ell_\lambda|$ has to increase at rate $\mathcal{O}(\sqrt{\omega})$. Indeed, the sequence of intervals is typically set as

$$\left\{\ell_i = [w_0 \wedge w_\omega - ci\sqrt{\omega}, w_0 \vee w_\omega + ci\sqrt{\omega}] : i = 1, 2, ...\right\}, \tag{4.84}$$

for some $c > 0$. This scaling of ℓ_λ usually implies exponential growth of ${\varphi_\xi}^\uparrow - {\varphi_\xi}^\downarrow$ in ω. The flipside is that the bound would tighten quickly if we could split the time interval $[0, \omega]$ into a few subintervals.

Consider the bridge $X_{(0,\omega)}$ spanning x_0 and x_ω, and suppose that τ is a partition of $[0, \omega]$. A simple divide-and-conquer approach to bound refinement proposed by [53] is to first generate

$$X_{\tau\setminus\{0,\omega\}} \sim \mathbb{W}|x_{\{0,\omega\}}. \tag{4.85}$$

Thereafter, given $\{X_\tau = x_\tau\}$, we sample the local layers $\Lambda_{(\dot\tau,\ddot\tau)}$ according to Section 4.4.2, with $\left|\ell_{(\dot\tau,\ddot\tau)}\right|$ decreasing at rate $\mathcal{O}(\sqrt{\operatorname{mesh}\tau})$. Therefore, the trajectory of a Brownian bridge proposal for X can be bounded with arbitrary accuracy by increasing $|\tau|$, at a linear cost in $|\tau|$. Since the bounds on X are now local, we also obtain the local bounds $\varphi^\downarrow_{(\dot\tau,\ddot\tau)}$ and $\varphi^\uparrow_{(\dot\tau,\ddot\tau)}$ on φ. The Poisson coin algorithm may be applied separately on each interval in τ, and the coin comes up heads if each subproposal on $(\dot\tau, \ddot\tau)$ is accepted. The corresponding probability of heads is

$$\prod_{(\dot\tau\sim\ddot\tau)\in\tau} \exp\left[\int_{\dot\tau}^{\ddot\tau} (\varphi^\downarrow_{(\dot\tau,\ddot\tau)} - \varphi(x_t))\,\mathrm{d}t\right], \tag{4.86}$$

which for many models and moderate $|\tau|$ increases faster than $\mathcal{O}(|\tau|)$, and hence realizes computational savings. The main shortcoming of this method is that τ has to be set in advance, requiring trial and error adjustments until the computional cost has been roughly minimized.

4.5 Discussion

We have introduced methods for estimation of, or rejection sampling according to the probability $\exp\left[\int_0^\omega (\varphi^\downarrow - \varphi(x_t))\,\mathrm{d}t\right]$, which enables diffusion path simulation and complete transition density estimation. Going forward, we will mainly treat those methods as black boxes, merely emphasizing that flipping a coin or estimating the above probability has cost $\mathcal{O}(\omega(\varphi^\uparrow - \varphi^\downarrow))$. Critically, once we introduce dependence on parameters θ, this quantity can be arbitrarily large, which raises concerns for the use of these tools within an MCMC algorithm on a possibly unbounded space $\mathcal{T}$. Worse still, if we wish to use the Poisson coin algorithm within the 2-coin algorithm of Section 2.3.1, the runtime could be exponentially large in $\omega(\varphi^\uparrow - \varphi^\downarrow)$.

When designing exact MCMC algorithms, we will devise various mitigation schemes that prevent expensive simulations from arising too often. In the case of the 2-coin

algorithm, we have already introduced the Portkey 2-coin algorithm, which has bounded expected runtime for given Poisson coin probabilities, though it remains unbounded once we introduce parameters θ. Complementarily, we have discussed the novel batch EA algorithm in Section 4.1.3, which allows for quick simulation of tail tosses within the 2-coin algorithm.

5 Exact Inference for Itō Diffusion Models

In this chapter, we finally return to the overarching questions raised in the introduction: we consider SDEs that involve a vector of unknown scalars θ, and the natural task of estimating those parameters based on a sequence of observations. We define a *parameterized Itō diffusion model* by way of the SDE

$$\mathrm{d}V_t = \mu_\theta(V_t)\,\mathrm{d}t + \sigma_\theta(V_t)\,\mathrm{d}W_t, \qquad (V_0 = v_0) \tag{5.1}$$

where $\mu_\theta : \mathbf{R} \to \mathbf{R}$ and $\sigma_\theta : \mathbf{R} \to (0, \infty)$ are valid drift and volatility functions for any choice of θ in some set $\mathcal{T}$, and W is a standard Brownian motion. If μ and σ are Lipschitz-continuous, the SDE is solved by a unique stochastic diffusion process, though broader conditions exist, see Section 3.1. Where necessary, we further restrict $\mathcal{T}$ such that the SDE has a unique solution for all elements of $\mathcal{T}$. For each θ, the SDE implies a transition density $\pi(v_t|v_0, \theta)$, which we assume to be intractable. We will mostly work within a Bayesian inference setting, where we treat the parameters as a random variable, denoted Θ, and assign some prior density $\pi(\theta)$ with support on $\mathcal{T}$. Given a set of observation times s and corresponding observations v_s, the posterior density of Θ is given by

$$\pi(\theta|v_s) = \frac{\pi(\theta)\prod_{(\dot{s}\sim\ddot{s})\in s}\pi(v_{\ddot{s}}|v_{\dot{s}}, \theta)}{\int \pi(\theta')\prod_{(\dot{s}\sim\ddot{s})\in s}\pi(v_{\ddot{s}}|v_{\dot{s}}, \theta')\,\mathrm{d}\theta'}. \tag{5.2}$$

This is an example of the twice intractable inference problem. On the one hand, the denominator is typically not available, as is common in Bayesian models. Therefore, expectations of test functions f with respect to the posterior, denoted $\mathrm{E}\left[f(\Theta)|v_s\right]$, cannot be derived analytically. On the other hand, the transition density $\pi(v_t|v_0, \theta)$ is unavailable as well, so we cannot evaluate the numerator either! This puts our task outside of the standard Bayesian computational setting, which assumes that the numerator can be evaluated, and we find ourselves in the setting of Chapter 2. Likelihood-based estimation has usually relied on an Euler approximation to the transition density, as laid out in Section 3.5. Bayesian examples of such approximate procedures are found e.g. in [69], [37], [35] and [108], while Maximum likelihood estimation dominates in Econometrics, as seen for example in [29], [98], [109], [34]. Higher-order expansions of the transition density have also been proposed, e.g. in [3].

Nonetheless, we need not give up on the benefits of the Bayesian approach and its computational methods. On the contrary, by leveraging the techniques introduced in Chapters 2, 3, and 4, we can provide an orthodox numerical treatment of the Bayesian inference problem. In a first step, in Section 5.1 we apply the hidden data augmentation strategy

discussed in Section 3.3. This is a natural starting point to any of the intractable likelihood inference strategies discussed in Chapter 2. It results in an explicit joint model $\pi(v, \theta)$ involving the entire diffusion path. Due to the Markov property and the resulting conditional independence structure, the natural strategy for that extended posterior is to carry out Gibbs sampling. To execute a Gibbs sampling approach successfully, we need to be mindful of posterior dependence of model variables, an issue previously discussed in Section 2.2.1. We do so in Section 5.1.2 by changing variables to obtain the model's noncentered parameterization. We proceed to addressing the infinite dimensionality of V in accordance with the retrospective sampling approach of Chapter 4, allowing us to implement the infinite dimensional algorithm based on finite information only. One option is to carry out further data augmentation, which we do in Section 5.1.3 based on the Poisson estimator methodology of Section 4.2. Based on those developments, we present two MCMC algorithms in Sections 5.2 and 5.3 that target $\pi(\theta|v_s)$, one of which more conventionally relies on the additional auxiliary variables, while the other implements the Bernoulli factory MCMC strategy introduced in Section 2.3. In what follows, we refer to those algorithms as the *auxiliary* and the *marginal* algorithm, respectively. The auxiliary algorithm presented here is closely related to the one originally proposed by [111], while the marginal algorithm improves on the methodology of [52]. We note that in investigating the more complicated problem of inference for jump diffusions, [53] have recently made various other improvements over the latter paper, many of which apply to other diffusion settings.

We will also present a method for approximate posterior sampling based on Euler discretization, based on the work of [108] on data augmentation for diffusion inference. Finally, in Section 5.5 we present an EM algorithm for exact MAP inference, derived from the treatment of maximum likelihood estimation for diffusions given in [17]. Notice that exact maximum likelihood estimation was also investigated from a pure simulated likelihood angle in [14].

The choice between the auxiliary and the marginal algorithm is determined by a complicated tradeoff. Following the logic of Section 2.2, the marginal algorithm is likely to be more efficient statistically since it has fewer latent variables. Nevertheless, it also has the statistical disadvantage pointed out in Section 2.1 of relying on the suboptimal Barker acceptance procedure, rather than the Metropolis procedure used by the auxiliary algorithm. Furthermore, due to the phenomena described in Section 2.3, acceptance decisions in the marginal algorithm are typically more expensive to carry out. We expect each single iteration of the marginal algorithm to be more beneficial, while the iteration time favors the auxiliary algorithm. The net effect may depend on the specific setting at hand, and is experimentally investigated in the simulation studies of Section 5.8.1.

In the broader context of the thesis, the chapter serves two main purposes. Firstly, many concepts that are required for inference in more complicated models are more easily introduced in the plain Itō diffusion setting. Once they have been properly motivated and explored in the more simple setting, the treatment of the complicated setting will be a natural extension. Secondly, recent advances in the development of inference

algorithms for those complicated models also allow for more efficient inference in the Itō diffusion context. Thirdly, we systematically explore the tradeoff between the auxiliary and marginal algorithm in this more simple setting.

Remark 1 (Unbounded iteration time of exact MCMC algorithms)**.** *Whichever way we pursue, it will transpire that the central difficulty of implementing exact MCMC algorithms lies in the path integrand*

$$\varphi_\theta(a) = 2^{-1}(\delta_\theta^2 + \delta_\theta')(a) \tag{5.3}$$

being unbounded in θ, even for fixed a, with δ_θ standing for the drift of the Lamperti-transformed diffusion. Hence, in accordance with the discussion of Section 4.5, there are parts of $\mathcal{T}$ in which exact algorithms, which are central to implementing the algorithms of this chapter, have arbitrarily long runtime, often of complexity $\mathcal{O}(\omega(\varphi_\theta^\uparrow - \varphi_\theta^\downarrow))$ or even $\mathcal{O}(e^{\omega(\varphi_\theta^\uparrow - \varphi_\theta^\downarrow)})$ for θ-dependent bounds $\varphi_\theta^\downarrow$ and $\varphi_\theta^\uparrow$. Loosely speaking, we will aim at developing methods such that

$$\mathrm{E}_{\Theta\sim\pi(\theta|v_s)}\left[\textit{Time per Iteration}\right] < \infty, \tag{5.4}$$

i.e. such that there is control of time per iteration when considering regions where the posterior probability is large. Keep in mind that this will require having some control of $\varphi_{\theta^\dagger}$ for any proposal $\theta^\dagger$. We have already introduced some mitigation techniques, namely the batch EA method of Section 4.1.3 and the Portkey Barker algorithm of Section 2.3.2. We further discuss those aspects in Sections 5.2.2 and 5.3.2.

5.1 Data Augmentation Strategy

We begin by obtaining an explicit model $\pi(v, \theta)$ by following the development given in Section 3.3. We introduce the diffusion bridges $V_{(\dot{s},\ddot{s})}$, upon which we can access the *complete transition density* $\pi(v_{(\dot{s},\ddot{s}]}|v_{\dot{s}}, \theta)$. The associated posterior on the augmented state space (V, Θ) is given by

$$\pi(v, \theta|v_s) \propto \pi(\theta) \prod_{(\dot{s}\sim\ddot{s})\in s} \pi(v_{(\dot{s},\ddot{s}]}|v_{\dot{s}}, \theta), \tag{5.5}$$

where we keep in mind that by targeting the augmented posterior, we also implicitly target the marginal posterior of interest.

The natural way of constructing an MCMC algorithm that targets (5.5) is to do Gibbs sampling, i.e. alternating updates to the full conditionals $\pi(v|v_s, \theta)$ and $\pi(\theta|v)$. As pointed out by [108], this is not immediately possible for many models, since V potentially contains perfect information about elements of θ, and vice versa. Hence, for distinct values θ and $\theta^\dagger$, the full conditionals $\pi(v|v_s, \theta)$ and $\pi(v|v_s, \theta^\dagger)$ may have disjoint support, and the Gibbs sampler is reducible and nonergodic. We can solve this problem with an invertible change of variables from (V, Θ) to the alternative *noncentered*

parameterization (Z, Θ). Under the new parameterization, the Gibbs sampler with full conditionals $\pi(z|v_s, \theta)$ and $\pi(\theta|v_s, z)$ is ergodic, and targets the *marginal* posterior

$$\pi(z, \theta|v_s) \propto \pi(\theta) \prod_{(\dot{s}\sim\ddot{s})\in s} \pi(z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta). \tag{5.6}$$

The details of the change of variable are given in Section 5.1.2. To implement the marginal algorithm, we must avoid the evaluation of expressions that involve the infinite dimensional path z.

In accordance with Section 4.2.2 and Theorem 14, a further augmentation step with the auxiliary Poisson process Ψ results in the *auxiliary* posterior

$$\pi(\psi, z, \theta|v_s) \propto \pi(\theta) \prod_{(\dot{s}\sim\ddot{s})\in s} \pi(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta), \tag{5.7}$$

which is derived in Section 5.1.3. Critically, the RHS can be evaluated in finite time, which allows for the development of an ordinary Metropolis-within-Gibbs sampler on the augmented space in Section 5.3.

5.1.1 Standing Assumptions and Complete Transition Density

Once θ is conditioned upon, the complete transition density is obtained as in the known SDE case given in Section 3.3 and Theorem 10. The Lamperti transform now depends on θ and corresponds to

$$\eta_\theta(a) = \int_{v^*}^{a} \frac{\mathrm{d}b}{\sigma_\theta(b)}, \qquad (v^*, a \in \mathcal{V}) \tag{5.8}$$

and the reduced process $X = \eta_\theta(V)$ follows the SDE

$$\mathrm{d}X_t = \delta_\theta(X_t)\,\mathrm{d}t + \mathrm{d}W_t, \qquad (X_0 = \eta_\theta(v_0)) \tag{5.9}$$

$$\delta_\theta(a) = \left(\frac{\mu_\theta}{\sigma_\theta} - \frac{\sigma'_\theta}{2}\right) \circ \eta_\theta^{-1}(a). \tag{5.10}$$

Suppose now that the usual assumptions apply for every $\theta \in \mathcal{T}$, i.e.

- $\delta_\theta(a)$ is continuously differentiable in $a \in \mathcal{X}$.
- The *Novikov condition* applies, i.e. $\mathrm{E}_{X_{(\dot{s},\ddot{s}]}}\left[\exp\left[\int_{\dot{s}}^{\ddot{s}} \delta_\theta^2(X_t)\,\mathrm{d}t\right]|x_{\dot{s}}, \theta\right] < \infty$.

Define $\mathbb{W}$ as the measure under which X is a Brownian motion. Then, by Theorem 10, the complete transition density with respect to the dominating measure $\mathbb{W}|(X_{\{\dot{s},\ddot{s}\}} =$

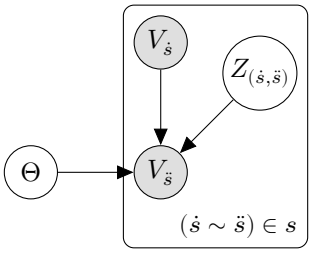


Figure 5.1: Plate diagram for the marginal noncentered model.

$\eta_\theta(v_{\{\dot{s},\ddot{s}\}})) \times \text{Leb}$ is given by

$$\pi(x_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta) = |\eta'_\theta(v_{\ddot{s}})| \, \mathrm{N}\left[\eta_\theta(v_{\ddot{s}}); \eta_\theta(v_{\dot{s}}), \ddot{s} - \dot{s}\right] \frac{\mathrm{d}\mathbb{X}|(X_{\dot{s}} = \eta_\theta(v_{\dot{s}}), \theta)}{\mathrm{d}\mathbb{W}|(X_{\dot{s}} = \eta_\theta(v_{\dot{s}}))}(x_{(\dot{s},\ddot{s})}, \eta_\theta(v_{\ddot{s}})), \tag{5.11}$$

$$\frac{\mathrm{d}\mathbb{X}|(x_{\dot{s}}, \theta)}{\mathrm{d}\mathbb{W}|x_{\dot{s}}}(x_{(\dot{s},\ddot{s}]}) = \exp\left[\Delta_\theta(x_{\ddot{s}}) - \Delta_\theta(x_{\dot{s}}) - \int_{\dot{s}}^{\ddot{s}} \varphi_\theta(x_t)\,\mathrm{d}t\right], \tag{5.12}$$

$$\varphi_\theta(a) = \frac{1}{2}\left(\delta_\theta^2(a) + \delta'_\theta(a)\right), \tag{5.13}$$

$$\Delta_\theta(a) = \int \delta_\theta(a)\,\mathrm{d}a. \tag{5.14}$$

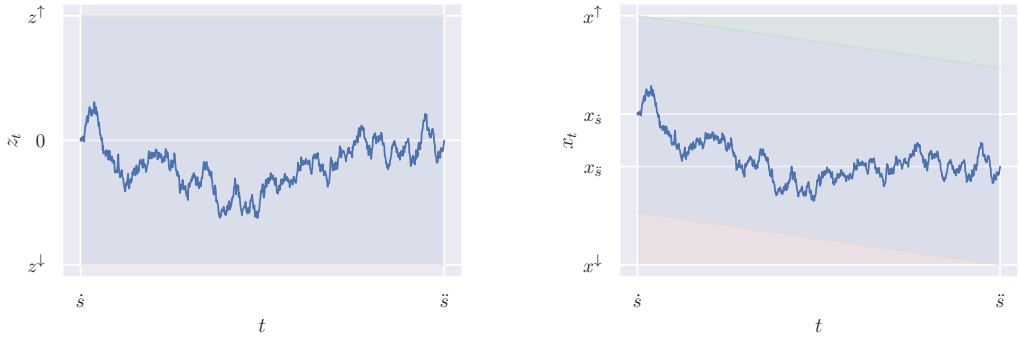
The result elucidates the connection between the dominating measure $\mathbb{W}|(X_{\{\dot{s},\ddot{s}\}} = \eta_\theta(v_{\{\dot{s},\ddot{s}\}}))$ and the failure of Gibbs sampling - for distinct values $\theta \neq \theta^\dagger$, $\mathbb{W}|(X_{\{\dot{s},\ddot{s}\}} = \eta_\theta(v_{\{\dot{s},\ddot{s}\}}))$ and $\mathbb{W}|(X_{\{\dot{s},\ddot{s}\}} = \eta_{\theta^\dagger}(v_{\{\dot{s},\ddot{s}\}}))$ have support on bridge paths with distinct endpoints, and they are mutually singular. Therefore, $\pi(x_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta)$ and $\pi(x_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta^\dagger)$ are mutually singular as well, precluding a move between values of Θ that change the endpoints of the bridge paths. Conversely, an appropriate change of variables must result in a density with respect to a dominating measure that is invariant in θ. This is an extreme version of the failure of Gibbs sampling where the fraction of missing information is large. In this instance, the latent variable has infinite dimension, and the fraction of missing information is 1. The phenomenon was investigated in detail in [108].

5.1.2 Marginal Noncentered Transition Density

When the fraction of missing information is large, the performance of Gibbs sampling may typically be improved by changing variables to a *noncentered* parameterization, as discussed in Section 2.2.1. We obtain this parameterization by removing the dependence of the imputed bridge on the endpoints $\eta_\theta(v_{\{\dot{s},\ddot{s}\}})$. Let $\zeta_\theta(x_t; v_{\{\dot{s},\ddot{s}\}})$ be defined by

$$\zeta_\theta(x_t; v_{\{\dot{s},\ddot{s}\}}) = x_t - \eta_\theta(v_{\dot{s}}) - (\eta_\theta(v_{\ddot{s}}) - \eta_\theta(v_{\dot{s}}))(t - \dot{s})/(\ddot{s} - \dot{s}), \qquad (t \in (\dot{s}, \ddot{s})) \tag{5.15}$$

We then change variables to $Z_{(\dot{s},\ddot{s})} = \zeta_\theta(X_{(\dot{s},\ddot{s})}; v_{\{\dot{s},\ddot{s}\}})$. Critically, if $X_{(\dot{s},\ddot{s})}$ is a Brownian bridge connecting $(\dot{s}, x_{\dot{s}})$ and $(\ddot{s}, x_{\ddot{s}})$, then $Z_{(\dot{s},\ddot{s})}$ is a Brownian bridge connecting $(\dot{s}, 0)$



(a) noncentered path.

(b) Centered path.

Figure 5.2: Illustration of a noncentered and a centered path, and the propagation of the noncentered to the centered path bounds. The blue-shaded area corresponds to the set to which we bound the noncentered Z and the centered X. While the uniform bounds on Z imply linear bounds bounds on X, we uniformize the bounds on X for simplicity. The red and green-shaded area correspond to the slack of the uniformized bounds on X.

and $(\ddot{s}, 0)$, and vice versa. The change of variables is illustrated in Figure 5.2. In addition, let ζ_θ^{-1} be the inverse of ζ_θ in its first argument, i.e.

$$\zeta_\theta^{-1}(z_t; v_{\{\dot{s},\ddot{s}\}}) = z_t + \eta_\theta(v_{\dot{s}}) + (\eta_\theta(v_{\ddot{s}}) - \eta_\theta(v_{\dot{s}}))(t - \dot{s})/(\ddot{s} - \dot{s}). \qquad (t \in (\dot{s}, \ddot{s})) \tag{5.16}$$

We further define $\mathbb{Z}|(x_{\{\dot{s},\ddot{s}\}}, \theta)$ and $\mathbb{B}^{(\dot{s},\ddot{s})}$ as the pushforward measures induced by $Z_{(\dot{s},\ddot{s})}$ under $\mathbb{X}|(x_{\{\dot{s},\ddot{s}\}}, \theta)$ and $\mathbb{W}|(x_{\{\dot{s},\ddot{s}\}}, \theta)$ respectively. Moreover, we note that under $\mathbb{B}_{(\dot{s},\ddot{s})}$, $Z_{(\dot{s},\ddot{s})}$ is a Brownian bridge hitting 0 at times $(\dot{s}, \ddot{s})$. Then, taking into account that probabilities have to be preserved under the change of variables, we find that

$$\begin{aligned} \frac{\mathrm{d}\mathbb{Z}|(x_{\{\dot{s},\ddot{s}\}}, \theta)}{\mathrm{d}\mathbb{B}_{(\dot{s},\ddot{s})}}(z_{(\dot{s},\ddot{s})}) &= \frac{\mathrm{d}\mathbb{X}|(x_{\{\dot{s},\ddot{s}\}}, \theta)}{\mathrm{d}\mathbb{W}|(x_{\{\dot{s},\ddot{s}\}}, \theta)} \circ \zeta_\theta^{-1}(z_{(\dot{s},\ddot{s})}; v_{\{\dot{s},\ddot{s}\}}) \\ &= \frac{\mathrm{N}\left[x_{\ddot{s}}; x_{\dot{s}}, \ddot{s} - \dot{s}\right]}{\pi(x_{\ddot{s}}|x_{\dot{s}}, \theta)} \frac{\mathrm{d}\mathbb{X}|(x_{\dot{s}}, \theta)}{\mathrm{d}\mathbb{W}|(x_{\dot{s}}, \theta)}(\zeta_\theta^{-1}(z_{(\dot{s},\ddot{s})}; v_{\{\dot{s},\ddot{s}\}}), x_{\ddot{s}}). \end{aligned} \tag{5.17}$$

Substituting that identity, we find the joint density of $(z_{(\dot{s},\ddot{s})}, x_{\ddot{s}})$:

$$\begin{aligned} \pi(z_{(\dot{s},\ddot{s})}, x_{\ddot{s}}|x_{\dot{s}}, \theta) &= \pi(x_{\ddot{s}}|x_{\dot{s}}, \theta) \frac{\mathrm{d}\mathbb{Z}|(x_{\{\dot{s},\ddot{s}\}}, \theta)}{\mathrm{d}\mathbb{B}_{(\dot{s},\ddot{s})}}(z_{(\dot{s},\ddot{s})}) \\ &= \mathrm{N}\left[x_{\ddot{s}}; x_{\dot{s}}, \ddot{s} - \dot{s}\right] \frac{\mathrm{d}\mathbb{X}|(x_{\dot{s}}, \theta)}{\mathrm{d}\mathbb{W}|(x_{\dot{s}}, \theta)}(\zeta_\theta^{-1}(z_{(\dot{s},\ddot{s})}; v_{\{\dot{s},\ddot{s}\}}), x_{\ddot{s}}) \end{aligned} \tag{5.18}$$

Finally, we change variables from $X_{\ddot{s}}$ to $V_{\ddot{s}}$, which gives us the noncentered complete transition density

$$\begin{aligned}\pi(z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta) &= |\eta_\theta'(v_{\ddot{s}})|\,\mathrm{N}\left[\eta_\theta(v_{\ddot{s}}); \eta_\theta(v_{\dot{s}}), \ddot{s}-\dot{s}\right] \\ &\quad \overbrace{\times \frac{\mathrm{d}\mathbb{X}|(X_{\dot{s}}=\eta_\theta(v_{\dot{s}}),\theta)}{\mathrm{d}\mathbb{W}|(X_{\dot{s}}=\eta_\theta(v_{\dot{s}}),\theta)}(\zeta_\theta^{-1}(z_{(\dot{s},\ddot{s})}; v_{\{\dot{s},\ddot{s}\}}), \eta_\theta(v_{\ddot{s}}))}^{d_\theta(v_{\{\dot{s},\ddot{s}\}})} \\ &= \left|\eta_\theta'(v_{\ddot{s}})\right|\mathrm{N}\left[\eta_\theta(v_{\ddot{s}}); \eta_\theta(v_{\dot{s}}), \ddot{s}-\dot{s}\right] e^{\Delta_\theta\circ\eta_\theta(v_{\ddot{s}})-\Delta_\theta\circ\eta_\theta(v_{\dot{s}})} \\ &\quad \times \underbrace{\exp\left[-\int_{\dot{s}}^{\ddot{s}} \varphi_\theta\circ\zeta_\theta^{-1}(z_t; v_{\{\dot{s},\ddot{s}\}})\,\mathrm{d}t\right]}_{q_\theta(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})}\end{aligned} \tag{5.19}$$

with respect to $\mathbb{B}_{(\dot{s},\ddot{s})} \times \mathrm{Leb}$. See Figure 5.1 for the corresponding graphical model. Armed with this density, we could in principle proceed to designing an ergodic Gibbs sampler. The remaining obstacle is that the paths $z_{(\dot{s},\ddot{s})}$ are infinite dimensional, and that the exponentiated path integrals $q_\theta(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})$ usually cannot be expressed in terms of a finite computation. One solution is to add additional latent variables, upon which we obtain a density which can be evaluated. This dual augmentation, derived in the following section, gives rise to the auxiliary algorithm.

Whether we carry out additional augmentation or not, we will require the ability to bound $\varphi_\theta\circ\zeta_\theta^{-1}$ on $(\dot{s},\ddot{s})$ in terms of the noncentered path $z_{(\dot{s},\ddot{s})}$, based on finite information. To save on ink, we frequently use the tilde accent as in

$$\widetilde{\varphi}_\theta = \varphi_\theta\circ\zeta_\theta^{-1} \tag{5.20}$$

to refer to the composition of a function in $z_{(\dot{s},\ddot{s})}$ with ζ_θ^{-1}. In the EA3 setting of Section 4.4, that information consists of lower and upper bounds

$$-\infty < z^{\downarrow}_{(\dot{s},\ddot{s})} \le z_t \le z^{\uparrow}_{(\dot{s},\ddot{s})} < \infty. \qquad (t\in(\dot{s},\ddot{s})) \tag{5.21}$$

Since $x_t = \zeta_\theta^{-1}(z_t; v_{\{\dot{s},\ddot{s}\}})$ is positive monotonous in z_t, the bounds are easily propagated:

$$\begin{aligned} x_t &\in \begin{pmatrix} z^{\downarrow}_{(\dot{s},\ddot{s})} + \eta_\theta(v_{\dot{s}}) + (\eta_\theta(v_{\ddot{s}})-\eta_\theta(v_{\dot{s}}))(t-\dot{s})/(\ddot{s}-\dot{s}), \\ z^{\uparrow}_{(\dot{s},\ddot{s})} + \eta_\theta(v_{\dot{s}}) + (\eta_\theta(v_{\ddot{s}})-\eta_\theta(v_{\dot{s}}))(t-\dot{s})/(\ddot{s}-\dot{s}) \end{pmatrix} \qquad (t\in(\dot{s},\ddot{s})) \\ &\in \begin{pmatrix} z^{\downarrow}_{(\dot{s},\ddot{s})} + \eta_\theta(v_{\dot{s}}) + \min_{t\in(\dot{s},\ddot{s})}\left[(\eta_\theta(v_{\ddot{s}})-\eta_\theta(v_{\dot{s}}))(t-\dot{s})\right]/(\ddot{s}-\dot{s}), \\ z^{\uparrow}_{(\dot{s},\ddot{s})} + \eta_\theta(v_{\dot{s}}) + \max_{t\in(\dot{s},\ddot{s})}\left[(\eta_\theta(v_{\ddot{s}})-\eta_\theta(v_{\dot{s}}))(t-\dot{s})\right]/(\ddot{s}-\dot{s}) \end{pmatrix} \\ &\in (\underbrace{z^{\downarrow}_{(\dot{s},\ddot{s})} + \eta_\theta(v_{\dot{s}}) \wedge \eta_\theta(v_{\ddot{s}})}_{x^{\downarrow}_{(\dot{s},\ddot{s})}}, \underbrace{z^{\uparrow}_{(\dot{s},\ddot{s})} + \eta_\theta(v_{\dot{s}}) \vee \eta_\theta(v_{\ddot{s}})}_{x^{\uparrow}_{(\dot{s},\ddot{s})}}). \end{aligned} \tag{5.22}$$

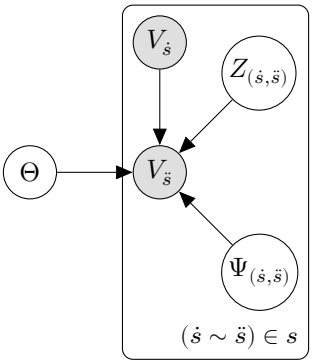


Figure 5.3: Plate diagram for the auxiliary noncentered model.

Notice that relative to the bounds on Z, these bounds are "loose" when the difference between $\eta_\theta(v_{\dot{s}})$ and $\eta_\theta(v_{\ddot{s}})$ is large. This is illustrated in Figure 5.2. We proceed to solving for the bounds

$$-\infty < \tilde{\varphi}_\theta^{\downarrow}(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}}) \leq \varphi_\theta(x) \leq \tilde{\varphi}_\theta^{\uparrow}(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}}) < \infty \qquad (x \in (x^{\downarrow}_{(\dot{s},\ddot{s})}, x^{\uparrow}_{(\dot{s},\ddot{s})})) \quad (5.23)$$

with the procedural methods developed in Chapter 8.

5.1.3 Auxiliary Noncentered Transition Density

We can avoid evaluations of the complete transition density by carrying out an additional data augmentation step. We do so by resorting to the transition density estimator introduced in Section 4.2, and adapting it to the noncentered parameterization.

The setup is similar. Let $\Psi_{(\dot{s},\ddot{s})}$ be a 2-dimensional Poisson process on $[\dot{s},\ddot{s}] \times [0,\infty)$ with induced measure $\mathbb{P}_{(\dot{s},\ddot{s})}$, and assume that for every θ, we have access to upper and lower bounds for $\varphi_\theta(x_t)$ on $(\dot{s},\ddot{s})$. Given these bounds, we define the truncation

$$\gamma_\theta(\psi_{(\dot{s},\ddot{s})}, x_{(\dot{s},\ddot{s})}) = \left\{ t : (t,\phi) \in \psi_{(\dot{s},\ddot{s})}, \phi \leq (\varphi_\theta^{\uparrow} - \varphi_\theta^{\downarrow})(x_{(\dot{s},\ddot{s})}) \right\}, \quad (5.24)$$

and notice that $\left|\gamma_\theta(\Psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})})\right|$ is almost surely finite. By Theorems 13 and 14, we obtain the *centered auxiliary complete transition density*

$$\begin{aligned} &\pi(\psi_{(\dot{s},\ddot{s})}, x_{(\dot{s},\ddot{s})}, v_{\ddot{s}} | v_{\dot{s}}, \theta) \\ &\quad = d_\theta(v_{\{\dot{s},\ddot{s}\}}) e^{(\dot{s}-\ddot{s})\varphi_\theta^{\downarrow}(x_{(\dot{s},\ddot{s})})} \prod_{t \in \gamma_\theta(\psi_{(\dot{s},\ddot{s})}, x_{(\dot{s},\ddot{s})})} \left\{ \left(\frac{\varphi_\theta{}^{\uparrow} - \varphi_\theta}{\varphi_\theta^{\uparrow} - \varphi_\theta^{\downarrow}} \right)(x_{(\dot{s},\ddot{s})}) \right\}_t \end{aligned} \quad (5.25)$$

with respect to $\mathbb{P}_{(\dot{s},\ddot{s})} \times \mathbb{W}|(X_{\{\dot{s},\ddot{s}\}} = \eta_\theta(v_{\{\dot{s},\ddot{s}\}})) \times \text{Leb}$, and with $d_\theta(v_{\{\dot{s},\ddot{s}\}})$ is defined as in (5.19). Changing variables from $X_{(\dot{s},\ddot{s})}$ to $Z_{(\dot{s},\ddot{s})} = \zeta_\theta(X_{(\dot{s},\ddot{s})}; v_{\{\dot{s},\ddot{s}\}})$, we move to the

noncentered auxiliary complete transition density

$$\pi(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta) = d_\theta(v_{\{\dot{s},\ddot{s}\}})\, \underbrace{e^{(\dot{s}-\ddot{s})\tilde{\varphi}_\theta^{\downarrow}(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})} \prod_{t\in\gamma_\theta(\psi_{(\dot{s},\ddot{s})}, x_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})} \left\{ \begin{pmatrix} \tilde{\varphi}_\theta^{\uparrow} - \tilde{\varphi}_\theta \\ \tilde{\varphi}_\theta^{\uparrow} - \tilde{\varphi}_\theta^{\downarrow} \end{pmatrix} (z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}}) \right\}_t}_{\bar{q}_\theta(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})}, \tag{5.26}$$

which is a density with respect to $\mathbb{P}_{(\dot{s},\ddot{s})} \times \mathbb{B}_{(\dot{s},\ddot{s})} \times \text{Leb}$, and where $\bar{q}_\theta(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})$ acts as the path integral estimate on the noncentered path. See Figure 5.1 for the corresponding graphical model. Since the product in the density is almost surely finite, there is no further obstacle to evaluating $\pi(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta)$, though we still have to address subtleties around the representation of the infinite dimensional path z and the computation of the bounds on φ_θ.

5.2 Marginal Algorithm

In this section, broadly following [53], we develop an MCMC algorithm that targets the marginal posterior

$$\pi(z, \theta|v_s) \propto \pi(\theta) \prod_{(\dot{s}\sim\ddot{s})\in s} \pi(z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta). \tag{5.27}$$

We work within the Gibbs sampling framework, where the most simple blocking scheme consists of the updates

$$Z: \quad \pi(z|v_s, \theta) \propto \prod_{(\dot{s}\sim\ddot{s})\in s} \pi(z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta), \tag{5.28}$$

$$\Theta: \quad \pi(\theta|v_s, z) \propto \pi(\theta) \prod_{(\dot{s}\sim\ddot{s})\in s} \pi(z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta). \tag{5.29}$$

Since the full conditional of Z factorizes into terms related to the individual bridges $Z_{(\dot{s},\ddot{s})}$, we may carry out the first update independently for each bridge according to

$$\pi(z_{(\dot{s},\ddot{s})}|v_{\dot{s}}, \theta) \propto \pi(z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta). \qquad ((\dot{s}\sim\ddot{s})\in s) \tag{5.30}$$

In keeping with the discussion in Section 2.2, we deem a method that exploits conditional independence to be most likely to result in a scalable procedure. Besides, while the dependence between Z and Θ typically increases with the observation horizon ω, the additional data will also tend to concentrate the marginal $\pi(\theta|v_s)$. As seen in Section 2.2.1, if both the full conditional and the marginal concentrate at similar rates, the efficiency of the Gibbs sampler is not affected.

5.2.1 Retrospective Simulation

In what follows, assume that the full conditional updates evolve by way of an accept-reject coin flip, where the acceptance probability depends on the complete transition densities $\pi(z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta)$, involving the path integral

$$\int_{\dot{s}}^{\ddot{s}} \tilde{\varphi}_\theta(z_t, v_{\{\dot{s},\ddot{s}\}})\,\mathrm{d}t. \tag{5.31}$$

Clearly, it is neither possible to exhaustively store z, nor to evaluate its path integral when z is rough. Hence, in order to implement the marginal algorithm, we need to avoid explicit likelihood evaluations. Indeed, we will use the Bernoulli factory MCMC approach introduced in Section 2.3 to simulate the accept-reject coin. All acceptance odds are represented in the form

$$\frac{c_1 p_1}{c_2 p_2}, \tag{5.32}$$

where p_1 and p_2 are of form $\exp\left[-\int_{\dot{s}}^{\ddot{s}} f(z_t)\,\mathrm{d}t\right]$ or $\exp\left[-\int_{\dot{s}}^{\ddot{s}} f(z_t, z_t^\dagger)\,\mathrm{d}t\right]$ for paths z_t and $z_t^\dagger$ and some nonnegative integrand f with known upper bound. Coins with probability corresponding to such functionals can be simulated by the Poisson coin algorithm of Section 4.1. This only requires evaluation of $z_{(\dot{s},\ddot{s})}$ at a finite set of times. Therefore, while the algorithm is formulated in terms of the infinite-dimensional path $z_{(\dot{s},\ddot{s})}$, that path need not be determined until required by the Poisson coin algorithm. Indeed, in the most general setting, $z_{(\dot{s},\ddot{s})}$ is represented in memory as (modulo some additional information specified in Section 4.4)

$$\left(\left\{z_{\dot{\nu}} : \dot{\nu} \in \nu_{(\dot{s},\ddot{s})}\right\}, z^{\downarrow}_{(\dot{s},\ddot{s})}, z^{\uparrow}_{(\dot{s},\ddot{s})}\right), \tag{5.33}$$

where $\nu_{(\dot{s},\ddot{s})}$ is the set of times at which $z_{(\dot{s},\ddot{s})}$ has been previously evaluated. The bounds on $z_{(\dot{s},\ddot{s})}$ are sufficient to evaluate the uniform bounds on the integrand f. Since $z_{(\dot{s},\ddot{s})}$ is proposed according to the Brownian bridge measure $\mathbb{B}_{(\dot{s},\ddot{s})}$, any additional evaluations can be carried out according to the conditional Brownian bridge simulation methods in Sections 4.3 and 4.4. That information is then propagated forward throughout the run of the algorithm.

5.2.2 Parameter Update

We implement the update to $\pi(\theta|v_s, z)$ as a Barker-within-Gibbs update. For a generic proposal $\kappa(\theta^\dagger|\theta)$, we could express the Barker acceptance odds in the following format

that is compatible with the 2-coin algorithm:

$$\begin{aligned}
\frac{\alpha_\Theta}{1-\alpha_\Theta} &= \frac{\pi(\theta^\dagger|v_s,z)}{\pi(\theta|v_s,z)}\frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)} \\
&= \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\frac{\pi(\theta^\dagger)}{\pi(\theta)}\prod_{(\dot{s}\sim\ddot{s})\in s}\frac{\pi(z_{(\dot{s},\ddot{s})},v_{\ddot{s}}|v_{\dot{s}},\theta^\dagger)}{\pi(z_{(\dot{s},\ddot{s})},v_{\ddot{s}}|v_{\dot{s}},\theta)} \\
&= \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\frac{\pi(\theta^\dagger)}{\pi(\theta)}\prod_{(\dot{s}\sim\ddot{s})\in s}\frac{d_{\theta^\dagger}(v_{\{\dot{s},\ddot{s}\}})}{d_\theta(v_{\{\dot{s},\ddot{s}\}})}\frac{e^{-\int_{\dot{s}}^{\ddot{s}}\tilde{\varphi}_{\theta^\dagger}(z_t,v_{\{\dot{s},\ddot{s}\}})\,\mathrm{d}t}}{e^{-\int_{\dot{s}}^{\ddot{s}}\tilde{\varphi}_{\theta}(z_t,v_{\{\dot{s},\ddot{s}\}})\,\mathrm{d}t}} \\
&= \underbrace{\frac{\overbrace{\kappa(\theta|\theta^\dagger)\pi(\theta^\dagger)\textstyle\prod_{(\dot{s}\sim\ddot{s})\in s}d_{\theta^\dagger}(v_{\{\dot{s},\ddot{s}\}})e^{(\ddot{s}-\dot{s})\tilde{\varphi}^\downarrow_{\theta^\dagger}(z_{(\dot{s},\ddot{s})},v_{\{\dot{s},\ddot{s}\}})}}^{c_1}}{\underbrace{\kappa(\theta^\dagger|\theta)\pi(\theta)\textstyle\prod_{(\dot{s}\sim\ddot{s})\in s}d_\theta(v_{\{\dot{s},\ddot{s}\}})e^{(\ddot{s}-\dot{s})\tilde{\varphi}^\downarrow_{\theta}(z_{(\dot{s},\ddot{s})},v_{\{\dot{s},\ddot{s}\}})}}_{c_2}}}_{} \\
&\quad\times\underbrace{\frac{\overbrace{\textstyle\prod_{(\dot{s}\sim\ddot{s})\in s}e^{\int_{\dot{s}}^{\ddot{s}}\tilde{\varphi}^\downarrow_{\theta^\dagger}(z_{(\dot{s},\ddot{s})},v_{\{\dot{s},\ddot{s}\}})-\tilde{\varphi}_{\theta^\dagger}(z_t,v_{\{\dot{s},\ddot{s}\}})\,\mathrm{d}t}}^{p_1}}{\underbrace{\textstyle\prod_{(\dot{s}\sim\ddot{s})\in s}e^{\int_{\dot{s}}^{\ddot{s}}\tilde{\varphi}^\downarrow_{\theta}(z_{(\dot{s},\ddot{s})},v_{\{\dot{s},\ddot{s}\}})-\tilde{\varphi}_{\theta}(z_t,v_{\{\dot{s},\ddot{s}\}})\,\mathrm{d}t}}_{p_2}},
\end{aligned} \tag{5.34}$$

where α_Θ is the acceptance probability for the Θ-update, p_1 and p_2 are probabilities because $\varphi^\downarrow_\theta - \varphi_\theta < 0$ by definition. While this factorization of the acceptance odds produces a valid 2-coin algorithm and was successfully implemented by [52], it is nonscalable in the length of the time series, and illustrates why it is critical to choose the correct factorization. Indeed,

$$\lim_{\omega\to\infty}\prod_{(\dot{s}\sim\ddot{s})\in s}\exp\left[\int_{\dot{s}}^{\ddot{s}}\tilde{\varphi}^\downarrow_\theta(z_{(\dot{s},\ddot{s})},v_{\{\dot{s},\ddot{s}\}})-\tilde{\varphi}_\theta(z_t,v_{\{\dot{s},\ddot{s}\}})\,\mathrm{d}t\right]\to 0 \tag{5.35}$$

almost surely unless $\varphi_\theta(X_t)=0$ almost everywhere in time. Therefore, p_1 and p_2 are increasingly small in the large ω regime, and the runtime (2.37) of the 2-coin algorithm diverges. This applies regardless of the attempted step size from θ to $\theta^\dagger$, and is not helped in the Portkey barker setting, where the acceptance probability goes to 0 rather than the runtime.

We therefore proceed to constructing a 2-coin algorithm whose runtime accelerates as $\left|\theta^\dagger-\theta\right|$ diminishes. The key operation consists of obtaining an integrand that vanishes in the step size:

$$\begin{aligned}
\frac{\alpha_\Theta}{1-\alpha_\Theta} &= \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\frac{\pi(\theta^\dagger)}{\pi(\theta)}\prod_{(\dot{s}\sim\ddot{s})\in s}\frac{d_{\theta^\dagger}(v_{\{\dot{s},\ddot{s}\}})}{d_\theta(v_{\{\dot{s},\ddot{s}\}})}\frac{e^{-\int_{\dot{s}}^{\ddot{s}}\tilde{\varphi}_{\theta^\dagger}(z_t,v_{\{\dot{s},\ddot{s}\}})\,\mathrm{d}t}}{e^{-\int_{\dot{s}}^{\ddot{s}}\tilde{\varphi}_{\theta}(z_t,v_{\{\dot{s},\ddot{s}\}})\,\mathrm{d}t}} \\
&= \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\frac{\pi(\theta^\dagger)}{\pi(\theta)}\prod_{(\dot{s}\sim\ddot{s})\in s}\frac{d_{\theta^\dagger}(v_{\{\dot{s},\ddot{s}\}})}{d_\theta(v_{\{\dot{s},\ddot{s}\}})}\exp\left[-\int_{\dot{s}}^{\ddot{s}}\chi_t\,\mathrm{d}t\right],
\end{aligned} \tag{5.36}$$

where α_Θ is the acceptance probability of the Θ-update, and we define

$$\chi_t = (\tilde{\varphi}_{\theta^\dagger} - \tilde{\varphi}_\theta)(z_t, v_{\{\dot{s},\ddot{s}\}}), \qquad (t \in (\dot{s}, \ddot{s})) \tag{5.37}$$

and denote its positive and negative parts $\chi_t^{(+)}$ and $\chi_t^{(-)}$. We obtain a factorization of the acceptance odds in an appropriate form:

$$\frac{\alpha_\Theta}{1-\alpha_\Theta} = \underbrace{\frac{\overbrace{\kappa(\theta|\theta^\dagger)\pi(\theta^\dagger)\prod_{(\dot{s}\sim\ddot{s})\in s} d_{\theta^\dagger}(v_{\{\dot{s},\ddot{s}\}})}^{c_1}}{\kappa(\theta^\dagger|\theta)\pi(\theta)\prod_{(\dot{s}\sim\ddot{s})\in s} d_\theta(v_{\{\dot{s},\ddot{s}\}})}}_{c_2} \underbrace{\frac{\overbrace{\prod_{(\dot{s}\sim\ddot{s})\in s} e^{-\int_{\dot{s}}^{\ddot{s}} \chi_t^{(+)}\,\mathrm{d}t}}^{p_1}}{\prod_{(\dot{s}\sim\ddot{s})\in s} e^{-\int_{\dot{s}}^{\ddot{s}} \chi_t^{(-)}\,\mathrm{d}t}}}_{p_2}, \tag{5.38}$$

where the integrands are nonnegative by definition, and therefore p_1 and p_2 are probabilities. Notice that the p_1- and p_2-coins can be simulated by independently simulating a Poisson coin corresponding to each factor in the product, and checking if all factor coins came up heads. The trivial upper bounds on $\chi_t^{(+)}$ and $\chi_t^{(-)}$ in the range $(\dot{s}, \ddot{s})$ are $(\varphi^\uparrow_{\theta^\dagger} - \varphi^\downarrow_\theta)(x_{(\dot{s},\ddot{s})})$ and $(\varphi^\uparrow_\theta - \varphi^\downarrow_{\theta^\dagger})(x_{(\dot{s},\ddot{s})})$, respectively. This factorization is advantageous in terms of 2-coin algorithm runtime because p_1 and p_2 decrease with the step size $|\theta^\dagger - \theta^\ddagger|$. In fact, we can derive vanishing quantitative bounds on the integrand given the step size.

Proposition 6 (Difference integrand bound)**.** *Let* C *be a convex set such that* $\theta, \theta^\dagger \in \mathrm{C}$ *and assume that* χ_t *is differentiable on* C*. Then, by the multivariate mean value theorem, there exists an* $a \in (0,1)$ *such that*

$$\chi_t = \left(\nabla_\theta \tilde{\varphi}_\theta(z_t, v_{\{\dot{s},\ddot{s}\}})\Big|_{\theta=(1-a)\theta^\dagger + a\theta^\ddagger}\right) \cdot (\theta^\dagger - \theta^\ddagger), \qquad (t \in (\dot{s}, \ddot{s})) \tag{5.39}$$

and by the Cauchy-Schwarz inequality we obtain the bound

$$|\chi_t| \le \sup_{a\in[0,1]} \left|\nabla_\theta \tilde{\varphi}_\theta(z_t, v_{\{\dot{s},\ddot{s}\}})\Big|_{\theta=(1-a)\theta^\dagger + a\theta^\ddagger}\right| |\theta^\dagger - \theta^\ddagger|. \tag{5.40}$$

If the posterior contracts at sufficient rate as the observation horizon ω *recedes, the optimal step size* $|\theta^\dagger - \theta^\ddagger|$ *can decrease sufficiently quickly to ensure that the expected 2-coin algorithm runtime is bounded above in* ω*.*

Example 12 (OU process)**.** *Suppose that* $\mathrm{d}V_t = -\beta V_t + \mathrm{d}W_t$ *with* $\beta > 0$*, so* X *follows the same SDE. Then,*

$$\tilde{\varphi}_\beta(z_t, v_{\{\dot{s},\ddot{s}\}}) = (\beta^2\omega^{-1}(z_t, v_{\{\dot{s},\ddot{s}\}})^2 - \beta)/2, \tag{5.41}$$

$$\partial_\beta \tilde{\varphi}_\beta(z_t, v_{\{\dot{s},\ddot{s}\}}) = (\beta\omega^{-1}(z_t, v_{\{\dot{s},\ddot{s}\}})^2 - 1/2), \tag{5.42}$$

and $|\chi_t|$ *is uniformly bounded on* $(\dot{s}, \ddot{s})$ *by the tractable expression*

$$\begin{aligned}|\chi_t| &\leq \sup_{a\in[0,1]} \left|((1-a)\beta^\dagger + a\beta^\ddagger)\zeta^{-1}(z_t, v_{\{\dot{s},\ddot{s}\}})^2 - 1/2\right| |\beta^\dagger - \beta^\ddagger| \\ &\leq ((\beta^\dagger \vee \beta^\ddagger)\zeta^{-1}(z_t; v_{\{\dot{s},\ddot{s}\}})^2 + 1/2)|\beta^\dagger - \beta^\ddagger| \\ &\leq ((\beta^\dagger \vee \beta^\ddagger)(\zeta^{-1}(z^{\downarrow}_{(\dot{s},\ddot{s})}; v_{\{\dot{s},\ddot{s}\}})^2 \vee \zeta^{-1}(z^{\uparrow}_{(\dot{s},\ddot{s})}; v_{\{\dot{s},\ddot{s}\}})^2) + 1/2)|\beta^\dagger - \beta^\ddagger|.\end{aligned} \tag{5.43}$$

Remark 2 (Sensitivity to proposal)**.** *Atypically, the 2-coin algorithm above has complexity depending on the gradient* $\nabla_\theta \tilde{\varphi}_\theta(z_t, v_{\{\dot{s},\ddot{s}\}})$*, rather than* φ_θ*. Nevertheless, the example illustrates that in many instances, we cannot bound the integrand for a given step size, nor the 2-coin algorithm runtime on an unbounded* $\mathcal{T}$*.*

5.2.3 Bridge Update

Contrary to the parameter update, the bridge update to $\pi(z|v_s, \theta)$ requires a more concrete discussion of the proposal mechanism. Since we will often retrospectively reveal z based on some constraints, we require a proposal for which such conditional simulation is well understood. Hence, it is natural to independently propose according to the Brownian bridge measure, i.e.

$$Z^\dagger_{(\dot{s},\ddot{s})} \sim \mathbb{B}_{(\dot{s},\ddot{s})}. \tag{5.44}$$

Notice that we write the proposal density $\kappa(z^\dagger_{(\dot{s},\ddot{s})})$ with respect to the same dominating measure as the full conditional, which is $\mathbb{B}_{(\dot{s},\ddot{s})}$, therefore $\kappa(z^\dagger_{(\dot{s},\ddot{s})}) = 1$. Exploiting conditional independence, we can carry out an independent Barker-within-Gibbs update to each bridge $Z^\dagger_{(\dot{s},\ddot{s})}$. The following factorization results in a valid 2-coin algorithm to sample the acceptance decision:

$$\begin{aligned}\frac{\alpha_{Z_{(\dot{s},\ddot{s})}}}{1-\alpha_{Z_{(\dot{s},\ddot{s})}}} &= \frac{\pi(z^\dagger_{(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}}, \theta)}{\pi(z_{(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}}, \theta)} \\ &= \frac{\pi(z^\dagger_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta)}{\pi(z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta)} \\ &= \frac{e^{-\int_{\dot{s}}^{\ddot{s}} \tilde{\varphi}_\theta(z^\dagger_t, v_{\{\dot{s},\ddot{s}\}})\,\mathrm{d}t}}{e^{-\int_{\dot{s}}^{\ddot{s}} \tilde{\varphi}_\theta(z_t, v_{\{\dot{s},\ddot{s}\}})\,\mathrm{d}t}} \\ &= \frac{\overbrace{e^{(\ddot{s}-\dot{s})\tilde{\varphi}^\downarrow_\theta(z^\dagger_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})}}^{c_1}}{\underbrace{e^{(\ddot{s}-\dot{s})\tilde{\varphi}^\downarrow_\theta(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})}}_{c_2}} \frac{\overbrace{e^{\int_{\dot{s}}^{\ddot{s}} \left\{(\tilde{\varphi}^\downarrow_\theta - \tilde{\varphi}_\theta)(z^\dagger_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})\right\}_t \mathrm{d}t}}^{p_1}}{\underbrace{e^{\int_{\dot{s}}^{\ddot{s}} \left\{(\tilde{\varphi}^\downarrow_\theta - \tilde{\varphi}_\theta)(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})\right\}_t \mathrm{d}t}}_{p_2}}.\end{aligned} \tag{5.45}$$

Since each bridge is of fixed length, this 2-coin algorithm typically terminates quickly. Nonetheless, because φ_θ involves the square of the drift function of X, it is usually

positive for a broad range of arguments, which suggests taking a similar route as in the parameter step and differencing the integrands. Define

$$\xi_t = \tilde{\varphi}_\theta(z_t^\dagger, v_{\{\dot{s},\ddot{s}\}}) - \tilde{\varphi}_\theta(z_t, v_{\{\dot{s},\ddot{s}\}}), \qquad (t \in (\dot{s}, \ddot{s})) \tag{5.46}$$

and denote its positive and negative parts $\xi_t^{(+)}$ and $\xi_t^{(-)}$. We obtain the following alternative 2-coin algorithm:

$$\begin{aligned} \frac{\alpha_{Z_{(\dot{s},\ddot{s})}}}{1-\alpha_{Z_{(\dot{s},\ddot{s})}}} &= \frac{e^{-\int_{\dot{s}}^{\ddot{s}} \tilde{\varphi}_\theta(z_t^\dagger, v_{\{\dot{s},\ddot{s}\}})\,\mathrm{d}t}}{e^{-\int_{\dot{s}}^{\ddot{s}} \tilde{\varphi}_\theta(z_t, v_{\{\dot{s},\ddot{s}\}})\,\mathrm{d}t}} \\ &= \exp\left[-\int_{\dot{s}}^{\ddot{s}} \xi_t \,\mathrm{d}t\right] \\ &= \frac{\overbrace{e^{-\int_{\dot{s}}^{\ddot{s}} \xi_t^{(+)}\,\mathrm{d}t}}^{p_1}}{\underbrace{e^{-\int_{\dot{s}}^{\ddot{s}} \xi_t^{(-)}\,\mathrm{d}t}}_{p_2}}, \end{aligned} \tag{5.47}$$

with $c_1 = c_2 = 1$. Corresponding bounds are given by

$$\xi_t^{(+)} \le \tilde{\varphi}_\theta^{\uparrow}(z_{(\dot{s},\ddot{s})}^\dagger, v_{\{\dot{s},\ddot{s}\}}) - \tilde{\varphi}_\theta^{\downarrow}(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}}), \qquad (t \in (\dot{s}, \ddot{s})) \tag{5.48}$$

$$\xi_t^{(-)} \le \tilde{\varphi}_\theta^{\uparrow}(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}}) - \tilde{\varphi}_\theta^{\downarrow}(z_{(\dot{s},\ddot{s})}^\dagger, v_{\{\dot{s},\ddot{s}\}}). \qquad (t \in (\dot{s}, \ddot{s})) \tag{5.49}$$

The disadvantage of this algorithm is that the integrand ξ_t has to be evaluated for both $z_{(\dot{s},\ddot{s})}^\dagger$ and $z_{(\dot{s},\ddot{s})}$ when simulating p_1- and p_2-tosses. In practice, we have found that the version using the integrand ξ_t runs moderately faster.

If the bridge update suffers from a very low acceptance probability, we can reduce the step size of the bridge update by conditioning on additional values on the path of V. We elaborate on this idea in Section 6.3.4 within the Markov switching context, where its application is more critical.

5.3 Auxiliary Algorithm

In this section, broadly following [111], we develop the alternative MCMC algorithm that targets the extended posterior

$$\pi(\psi, z, \theta | v_s) \propto \pi(\theta) \prod_{(\dot{s}\sim\ddot{s})\in s} \pi(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\ddot{s}} | v_{\dot{s}}, \theta). \tag{5.50}$$

The density on the RHS almost surely depends only on a finite number of evaluations of z, therefore it can be fully evaluated with finite computation, and we can apply

conventional accept-reject methods. The natural Gibbs sampler for this model consists of the updates

$$(\Psi, Z): \quad \pi(\psi, z|v_s, \theta) \propto \prod_{(\dot{s}\sim\ddot{s})\in s} \pi(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta), \tag{5.51}$$

$$\Theta: \quad \pi(\theta|v_s, \psi, z) \propto \pi(\theta) \prod_{(\dot{s}\sim\ddot{s})\in s} \pi(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta). \tag{5.52}$$

As for the marginal algorithm, the full conditional of (Ψ, Z) factorizes into terms relating to the sections $(\Psi_{(\dot{s},\ddot{s})}, Z_{(\dot{s},\ddot{s})})$. Thus, we independently update those according to

$$\pi(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}}, \theta) \propto \pi(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta). \qquad ((\dot{s}\sim\ddot{s})\in s) \tag{5.53}$$

Compared to the marginal algorithm, this algorithm involves additional conditioning in the update to $\pi(\theta|v_s, \psi, z)$.

While this blocking scheme is identical to the one proposed in [111], a notable difference is that both updates of this algorithm use Metropolis-within-Gibbs, while [111] uses rejection sampling to obtain an independent sample from $\pi(\psi, z|v_s, \theta)$, which mixes more quickly than the Metropolis-within-Gibbs update. Conversely, the cost of the Metropolis-within-Gibbs update is equivalent to Poisson estimation, while the rejection sampling update requires Poisson coin simulation until the first success, which is typically more expensive. Moreover, it is restricted to models for which φ_θ is uniformly lower bounded, since otherwise the Radon-Nikodym derivative is unbounded. One such model is the Wright-Fisher diffusion as seen in [52]. Moreover, as shown in Proposition 7, whenever the rejection sampler can be implemented, Proposition 7 shows that the Metropolis-within-Gibbs update is uniformly ergodic, which means that it quickly approaches the exact full conditional update when iterated a few times. Therefore, there is little downside to using Metropolis-within-Gibbs, for the upside of greater generality and comparability to the marginal algorithm.

5.3.1 Retrospective Simulation

While the joint density of the auxiliary model can be fully evaluated based on a finite subset of $(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})})$, that subset is not known ahead of time, and may evolve depending on θ. Therefore, just as in the marginal algorithm, we take the retrospective simulation approach, defining the model and the algorithm based on infinite-dimensional quantities, but only revealing as much information as required to carry out accept-reject decisions. Indeed, the information required for any state of the Markov chain is determined by

$$\tilde{\gamma}_\theta(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}}) = \left\{t : (t, \phi) \in \psi_{(\dot{s},\ddot{s})}, \phi \le (\tilde{\varphi}_\theta^{\uparrow} - \tilde{\varphi}_\theta^{\downarrow})(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})\right\}. \tag{5.54}$$

Therefore, if $\psi_{(\dot{s},\ddot{s})}$ is proposed according to $\mathbb{P}_{(\dot{s},\ddot{s})}$, we reveal $\tilde{\gamma}_\theta(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})$ as a Poisson process on the rectangle $[\dot{s}, \ddot{s}] \times [0, (\tilde{\varphi}_\theta^{\uparrow} - \tilde{\varphi}_\theta^{\downarrow})(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})]$, with $z_{(\dot{s},\ddot{s})}$ revealed at times corresponding to $\tilde{\gamma}_\theta(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})$.

Suppose now that $\phi^\uparrow$ is the largest value of $\widetilde{\varphi}_\theta^\uparrow - \widetilde{\varphi}_\theta^\downarrow$ so far encountered since $\psi_{(\dot{s},\ddot{s})}$ was proposed, and $\psi_{(\dot{s},\ddot{s})}$ is represented as and revealed up to

$$\left\{t : (t,\phi) \in \psi_{(\dot{s},\ddot{s})}, \phi \leq \phi^\uparrow\right\}. \tag{5.55}$$

When proposing a new parameter value $\theta^\dagger$ such that $\widetilde{\varphi}_{\theta^\dagger}^\uparrow - \widetilde{\varphi}_{\theta^\dagger}^\downarrow \leq \phi^\uparrow$, then $\tilde{\gamma}_{\theta^\dagger}$ is a subset of the previously revealed times, and no new information needs to be revealed. In the opposite case, we additionally reveal $\psi_{(\dot{s},\ddot{s})}$ on $[\dot{s},\ddot{s}] \times [\phi^\uparrow, (\widetilde{\varphi}_{\theta^\dagger}^\uparrow - \widetilde{\varphi}_{\theta^\dagger}^\downarrow)(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})]$ and update $\phi^\uparrow \leftarrow \widetilde{\varphi}_{\theta^\dagger}^\uparrow - \widetilde{\varphi}_{\theta^\dagger}^\downarrow$. $z_{(\dot{s},\ddot{s})}$ is then revealed at the new times in $\tilde{\gamma}_{\theta^\dagger}$. As in the marginal algorithm, when $z_{(\dot{s},\ddot{s})}$ is proposed according to $\mathbb{B}_{(\dot{s},\ddot{s})}$, we use the previously presented conditional Brownian bridge simulation methods of Sections 4.3 and 4.4. This finally allows for the evaluation of the product

$$\prod_{t \in \tilde{\gamma}_\theta(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})} \left\{ \left(\frac{\widetilde{\varphi}_\theta^\uparrow - \widetilde{\varphi}_\theta}{\widetilde{\varphi}_\theta^\uparrow - \widetilde{\varphi}_\theta^\downarrow} \right) (z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}}) \right\}_t. \tag{5.56}$$

5.3.2 Parameter Update

We implement the update to $\pi(\theta|v_s, \psi, z)$ as a Metropolis-within-Gibbs update. For a generic proposal $\kappa(\theta^\dagger|\theta)$, the acceptance probability is

$$\begin{aligned}
\alpha_\Theta &= 1 \wedge \frac{\pi(\theta^\dagger|v_s,\psi,z)}{\pi(\theta|v_s,\psi,z)} \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)} \\
&= 1 \wedge \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)} \frac{\pi(\theta^\dagger)}{\pi(\theta)} \prod_{(\dot{s}\sim\ddot{s})\in s} \frac{\pi(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta^\dagger)}{\pi(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta)} \\
&= 1 \wedge \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)} \frac{\pi(\theta^\dagger)}{\pi(\theta)} \prod_{(\dot{s}\sim\ddot{s})\in s} \frac{d_{\theta^\dagger}(v_{\{\dot{s},\ddot{s}\}})}{d_\theta(v_{\{\dot{s},\ddot{s}\}})} \frac{\bar{q}_{\theta^\dagger}(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})}{\bar{q}_\theta(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})}.
\end{aligned} \tag{5.57}$$

Notably, we need to evaluate $\bar{q}_{\theta^\dagger}(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})$, which corresponds to a Poisson estimator evaluation at expected cost $\mathcal{O}((\ddot{s}-\dot{s})(\widetilde{\varphi}_{\theta^\dagger}^\uparrow - \widetilde{\varphi}_{\theta^\dagger}^\downarrow)(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}}))$.

Remark 3 (Sensitivity to proposal)**.** *Since the expected cost is unbounded on $\mathcal{T}$, aggressive proposals $\theta^\dagger$ can result in arbitrarily expensive acceptance probability evaluations, even if that acceptance probability is extremely small. Moreover, unlike in the marginal algorithm, there is no possible mitigation through batch EA or Portkey Barker that allows for cheap rejection of $\theta^\dagger$. If evaluation of the acceptance probability would be prohibitive, e.g. in terms of memory, the proposal has to be rejected on faith, resulting in loss of pure exactness.*

5.3.3 Bridge and Poisson Process Update

As for the parameter update, we implement the bridge and Poisson process update to $\pi(\psi, z|v_s, \theta)$ as a Metropolis-within-Gibbs update. We use the independence proposal

$$\Psi^\dagger_{(\dot{s},\ddot{s})} \sim \mathbb{P}_{(\dot{s},\ddot{s})}, \quad Z^\dagger_{(\dot{s},\ddot{s})} \sim \mathbb{B}_{(\dot{s},\ddot{s})}, \tag{5.58}$$

where we notice that we write the proposal density $\kappa(\psi^\dagger_{(\dot{s},\ddot{s})})\kappa(z^\dagger_{(\dot{s},\ddot{s})})$ with respect to the dominating measure $\mathbb{P}_{(\dot{s},\ddot{s})} \times \mathbb{B}_{(\dot{s},\ddot{s})}$, therefore $\kappa(\psi^\dagger_{(\dot{s},\ddot{s})})\kappa(z^\dagger_{(\dot{s},\ddot{s})}) = 1$. We accept independently update $(\Psi_{(\dot{s},\ddot{s})}, Z_{(\dot{s},\ddot{s})})$ with probability

$$\begin{aligned}
\alpha_{(\Psi_{(\dot{s},\ddot{s})}, Z_{(\dot{s},\ddot{s})})} &= 1 \wedge \frac{\pi(\psi^\dagger_{(\dot{s},\ddot{s})}, z^\dagger_{(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}}, \theta)}{\pi(\psi^\dagger_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}}, \theta)} \\
&= 1 \wedge \frac{\pi(\psi^\dagger_{(\dot{s},\ddot{s})}, z^\dagger_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta)}{\pi(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta)} \\
&= 1 \wedge \frac{\bar{q}_\theta(\psi^\dagger_{(\dot{s},\ddot{s})}, z^\dagger_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})}{\bar{q}_\theta(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})}.
\end{aligned} \tag{5.59}$$

As in the marginal algorithm, this independence proposal may suffer from very low acceptance probabilities if observations are far apart in time. We can adopt the localization ideas of Section 6.4.4 to reduce the step size of the update. Nonetheless, we note that in the framework of [88], we can easily demonstrate that an update with independence proposal is uniformly ergodic for the class of models where rejection sampling is possible.

Proposition 7 (Uniform ergodicity of bridge and Poisson process update)**.** *Let $\pi(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}}, \theta)$ be the full conditional, and $\kappa(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})})$ the independence proposal as defined above. If φ_θ is uniformly bounded below on the reduced support $\mathcal{X} = \eta_\theta(\mathcal{V})$ for every $\theta \in \mathcal{T}$, the update is uniformly ergodic.*

Proof. By [88, Theorem 2.1], the update is uniformly ergodic if for any given θ, there is an upper bound to

$$\begin{aligned}
&\frac{\pi(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}}, \theta)}{\kappa(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})})} \\
&\quad = \bar{q}_\theta(\psi_{(\dot{s},\ddot{s})}, z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}}) \\
&\quad = e^{(\dot{s}-\ddot{s})\tilde{\varphi}^\downarrow_\theta(z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})} \prod_{t \in \gamma_\theta(\psi_{(\dot{s},\ddot{s})}, x_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}})} \left\{ \left(\frac{\tilde{\varphi}^\uparrow_\theta - \tilde{\varphi}_\theta}{\tilde{\varphi}^\uparrow_\theta - \tilde{\varphi}^\downarrow_\theta} \right) (z_{(\dot{s},\ddot{s})}, v_{\{\dot{s},\ddot{s}\}}) \right\}_t \\
&\quad \leq \exp\left[(\dot{s} - \ddot{s}) \inf_{a \in \mathcal{X}} \varphi_\theta(a) \right],
\end{aligned} \tag{5.60}$$

i.e. if φ_θ is uniformly bounded below on $\mathcal{X}$. □

5.4 Approximate Algorithm

In developing an approximate Bayesian inference algorithm, we rely on the Euler approximation framework introduced in Section 3.5. It results in the transition density approximation

$$\bar{\pi}(v_{\ddot{s}}|v_{\dot{s}},\theta) = |\eta'_\theta(v_{\ddot{s}})|\,\mathrm{N}\left[\eta_\theta(v_{\ddot{s}});\eta_\theta(v_{\dot{s}}) + (\ddot{s}-\dot{s})\delta_\theta \circ \eta_\theta(v_{\dot{s}}), \ddot{s}-\dot{s}\right]. \tag{5.61}$$

The natural way of refining this approximation is through a partial data augmentation with a finite subset of $X_{(\dot{s},\ddot{s})}$. Let $u_{[\dot{s},\ddot{s}]}$ be a partition of $[\dot{s},\ddot{s}]$, and $\bar{X}_{(\dot{s},\ddot{s})} = X_{u_{[\dot{s},\ddot{s}]}\setminus\{\dot{s},\ddot{s}\}}$. We obtain the approximate density

$$\bar{\pi}(v_{\ddot{s}},\bar{x}_{(\dot{s},\ddot{s})}|v_{\dot{s}},\theta) = |\eta'_\theta(v_{\ddot{s}})| \prod_{(\dot{u}\sim\ddot{u})\in u_{[\dot{s},\ddot{s}]}} \mathrm{N}\left[x_{\ddot{s}}; x_{\dot{s}} + (\ddot{s}-\dot{s})\delta_\theta(x_{\dot{s}}), \ddot{s}-\dot{s}\right], \tag{5.62}$$

$$\lim_{\text{mesh } u_{[\dot{s},\ddot{s}]}\to 0} \mathrm{E}_{\bar{X}_{(\dot{s},\ddot{s})}}\left[\bar{\pi}(v_{\ddot{s}},\bar{X}_{(\dot{s},\ddot{s})}|v_{\dot{s}},\theta)|v_{\{\dot{s},\ddot{s}\}},\theta\right] = \pi(v_{\ddot{s}}|v_{\dot{s}},\theta), \tag{5.63}$$

and due to weak convergence of the Euler-Maruyama scheme, it recovers the correct transition density as mesh $u_{[\dot{s},\ddot{s}]} \to 0$. We can apply this reasoning to posterior sampling by iterating parameter updates $\Theta|\bar{X}_{(\dot{s},\ddot{s})}$ and hidden data updates $\bar{X}_{(\dot{s},\ddot{s})}|\Theta$. As the mesh u is refined, the full conditionals $\pi(\theta|\bar{x}_{(\dot{s},\ddot{s})})$ and $\pi(\bar{x}_{(\dot{s},\ddot{s})}|\theta)$ become increasingly narrow, until for mesh $u_{[\dot{s},\ddot{s}]} \to 0$ we recover the mutual singularity of the exact case. This phenomenon kicks in even for moderate mesh sizes and is investigated in detail by [108]. The cure is the same as in the exact case, and consists of noncentering the imputed observations. We define the noncentered approximate density as

$$\begin{aligned}
&\bar{\pi}(v_{\ddot{s}},\bar{z}_{(\dot{s},\ddot{s})}|v_{\dot{s}},\theta) \\
&\quad = |\eta'_\theta(v_{\ddot{s}})| \prod_{z_t\in\bar{z}_{(\dot{s},\ddot{s})}} \left|\partial_{z_t}\zeta_\theta^{-1}(z_t; v_{\{\dot{s},\ddot{s}\}})\right| \\
&\qquad \prod_{(\dot{u}\sim\ddot{u})\in u_{[\dot{s},\ddot{s}]}} \mathrm{N}\left[\zeta_\theta^{-1}(z_{\ddot{u}}; v_{\{\dot{s},\ddot{s}\}}); \underset{\ddot{u}-\dot{u}}{\zeta_\theta^{-1}(z_{\dot{u}};v_{\{\dot{s},\ddot{s}\}}) + (\ddot{u}-\dot{u})\tilde{\delta}_\theta(z_{\dot{u}},v_{\{\dot{s},\ddot{s}\}})},\right],
\end{aligned} \tag{5.64}$$

where we slightly abuse notation by setting $\zeta_\theta^{-1}(z_{\dot{s}}; v_{\{\dot{s},\ddot{s}\}}) = v_{\dot{s}}$ and $\zeta_\theta^{-1}(z_{\ddot{s}}; v_{\{\dot{s},\ddot{s}\}}) = v_{\ddot{s}}$, and notice that the Jacobian drops from the formula:

$$\left|\partial_{z_t}\zeta_\theta^{-1}(z_t; v_{\{\dot{s},\ddot{s}\}})\right| = 1. \qquad (t \in (\dot{s},\ddot{s})) \tag{5.65}$$

This parameterization of the missing data conserves ergodicity as mesh $u_{[\dot{s},\ddot{s}]} \to 0$. It gives us a viable, approximate augmentation scheme within the same Gibbs blocking scheme as in the marginal algorithm of Section 5.2. The approximate posterior targeted by that sampler is

$$\bar{\pi}(\bar{z},\theta|v_s) \propto \pi(\theta) \prod_{(\dot{s}\sim\ddot{s})\in s} \bar{\pi}(v_{\ddot{s}},\bar{z}_{(\dot{s},\ddot{s})}|v_{\dot{s}},\theta) \tag{5.66}$$

and its Gibbs updates are

$$\bar{Z}: \quad \bar{\pi}(\bar{z}|v_s,\theta) \propto \prod_{(\dot{s}\sim\ddot{s})\in s} \bar{\pi}(\bar{z}_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}},\theta), \tag{5.67}$$

$$\Theta: \quad \bar{\pi}(\theta|v_s,\bar{z}) \propto \pi(\theta) \prod_{(\dot{s}\sim\ddot{s})\in s} \bar{\pi}(\bar{z}_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}},\theta). \tag{5.68}$$

As usual, the first update decomposes into bridge updates according to

$$\bar{\pi}(\bar{z}_{(\dot{s},\ddot{s})}|v_{\dot{s}},\theta) \propto \bar{\pi}(\bar{z}_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}},\theta). \qquad ((\dot{s}\sim\ddot{s})\in s) \tag{5.69}$$

Since the iteration time of the approximate algorithm is deterministic, it is will suited for "warm-starting" the exact algorithms, allowing them to start from a set of values close to the posterior mode, and setting any tuning parameters to useful starting values. Once properly tuned, it is less likely that the exact algorithms will visit parts of $\mathcal{T}$ where iteration times are onerous. This is very helpful in avoiding the concerns foreshadowed in Remark 1, and we follow that practice in our simulation studies in Section 5.8.

5.4.1 Parameter Update

We implement the update to $\bar{\pi}(\theta|v_s,\psi,z)$ as a Metropolis-within-Gibbs update. For a generic proposal $\kappa(\theta^\dagger|\theta)$, the acceptance probability is

$$\begin{aligned}\alpha_\Theta &= 1 \wedge \frac{\bar{\pi}(\theta^\dagger|v_s,\bar{z})}{\bar{\pi}(\theta|v_s,\bar{z})}\frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)} \\ &= 1 \wedge \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\frac{\pi(\theta^\dagger)}{\pi(\theta)}\prod_{(\dot{s}\sim\ddot{s})\in s}\frac{\bar{\pi}(\bar{z}_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}},\theta^\dagger)}{\bar{\pi}(\bar{z}_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}},\theta)}.\end{aligned} \tag{5.70}$$

5.4.2 Bridge Update

We again implement the update $\bar{\pi}(\bar{z}_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}},\theta)$ as a Metropolis-within-Gibbs update, with the independence proposal

$$\bar{Z}^\dagger_{(\dot{s},\ddot{s})} \sim \mathbb{B}_{(\dot{s},\ddot{s})}, \tag{5.71}$$

though we note that more sophisticated proposal mechanisms have been proposed, such as by [34]. There are various ways to choose $u^\dagger_{[\dot{s},\ddot{s}]}$, the most simple of which is to simply fix it to some grid width a fixed spacing. Notice that in this instance the dominating measure of $\bar{\pi}(\bar{z}_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}},\theta)$ is $\mathrm{Leb}^{|\bar{z}_{(\dot{s},\ddot{s})}|}$, therefore we also write the proposal density $\kappa(\bar{z}^\dagger_{(\dot{s},\ddot{s})})$ with respect to $\mathrm{Leb}^{|\bar{z}^\dagger_{(\dot{s},\ddot{s})}|}$:

$$\kappa(\bar{z}^\dagger_{(\dot{s},\ddot{s})}) = \frac{\prod_{(\dot{u}\sim\ddot{u})\in u^\dagger_{[\dot{s},\ddot{s}]}} \mathrm{N}\left[z^\dagger_{\ddot{u}}; z^\dagger_{\dot{u}}, \ddot{u}-\dot{u}\right]}{\mathrm{N}\left[0; 0, \ddot{s}-\dot{s}\right]}, \tag{5.72}$$

where $\bar{z}_{\ddot{s}} = 0$. We accept the proposal with probability

$$\begin{aligned}\alpha_{\bar{Z}_{(\dot{s},\ddot{s})}} &= 1 \wedge \frac{\bar{\pi}(\bar{z}^{\dagger}_{(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}},\theta)}{\bar{\pi}(\bar{z}_{(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}},\theta)} \frac{\kappa(\bar{z}_{(\dot{s},\ddot{s})})}{\kappa(\bar{z}^{\dagger}_{(\dot{s},\ddot{s})})} \\ &= 1 \wedge \frac{\kappa(\bar{z}_{(\dot{s},\ddot{s})})}{\kappa(\bar{z}^{\dagger}_{(\dot{s},\ddot{s})})} \frac{\bar{\pi}(\bar{z}^{\dagger}_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}},\theta)}{\bar{\pi}(\bar{z}_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}},\theta)}.\end{aligned} \tag{5.73}$$

5.5 MAP and Maximum Likelihood Estimation

A natural companion problem to posterior sampling is *maximum a posteriori* (MAP) estimation, i.e. finding the set of values $\theta^{\ddagger}$ such that

$$\theta^{\ddagger} = \underset{\theta}{\operatorname{argmax}}\ \pi(\theta, v_{s\setminus\{0\}}|v_0). \tag{5.74}$$

The MAP estimator also corresponds to the maximum likelihood estimator when setting $\pi(\theta) \propto 1$. In this section, we modify an approach originally proposed for Itō diffusions [17]. It consists of constructing a Monte Carlo EM algorithm, a framework introduced by [124], by leveraging parts of the Gibbs sampler that we developed for posterior sampling. An EM algorithm is an optimization consists of an *E-step* (for expectation), where a lower bound to the objective is constructed, and an *M-step* (for maximization), where the lower bound is maximized. These steps are alternated until some convergence criterion is reached. An MCEM algorithm replaces the construction of the lower bound with an unbiased estimator thereof.

5.5.1 Log Transition Density Estimation

In order to construct a MCEM algorithm, we require an unbiased estimator of the log complete transition density, and more specifically of the path integral therein:

$$\log \pi(z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}},\theta) = \log d_\theta(v_{\{\dot{s},\ddot{s}\}}) - \int_{\dot{s}}^{\ddot{s}} \varphi_\theta(x_t)\,\mathrm{d}t. \tag{5.75}$$

Unbiased estimation is easily accomplished by uniform sampling along the path:

$$-(\ddot{s}-\dot{s})\varphi_\theta(x_U), \quad U \sim \mathrm{Unif}\left[\dot{s},\ddot{s}\right]. \tag{5.76}$$

Thus, we define the log augmented transition density estimator

$$\bar{\ell}_u(z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}},\theta) = \log d_\theta(v_{\{\dot{s},\ddot{s}\}}) - (\ddot{s}-\dot{s})\varphi_\theta(x_u), \tag{5.77}$$

$$\mathrm{E}_U\left[\bar{\ell}_U(z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}},\theta)\right] = \log \pi(z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}},\theta). \tag{5.78}$$

The relative simplicity of the log augmented transition estimator is the main benefit of taking the MCEM approach.

5.5.2 E-Step

The E-step consists of finding a lower bound on the objective $\pi(\theta, v_{s\setminus\{0\}}|v_0)$. It is obtained by averaging the joint density over the posterior of the latent variables, i.e.

$$\begin{aligned} Q(\theta^\dagger, \theta) &= \mathrm{E}_Z \left[\log \pi(Z, \theta^\dagger, v_{s\setminus\{0\}}|v_0)|v_s, \theta\right] \\ &= \sum_{(\dot{s}\sim\ddot{s})\in s} \mathrm{E}_{Z_{(\dot{s},\ddot{s})}} \left[\log \pi(Z_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta^\dagger)|v_{\{\dot{s},\ddot{s}\}}, \theta\right] + \log \pi(\theta^\dagger), \end{aligned} \tag{5.79}$$

where we take expectations with respect to $\pi(z_{(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}}, \theta)$. Since this integral is intractable, we may instead obtain samples from $\pi(z_{(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}}, \theta)$ by repeatedly carrying out the bridge update from Section 5.2.3. That results in a dependent sequence of $\hat{l}$ samples $z^{(l)}$, to which we independently add the uniform variates $(u^{(l)}_{(\dot{\tau},\ddot{\tau})} : (\dot{s} \sim \ddot{s}) \in s)$. Given such a sequence, we obtain the unbiased Q-estimator:

$$\bar{Q}(\theta^\dagger) = \log \pi(\theta^\dagger) + \hat{l}^{-1} \sum_{l=1}^{\hat{l}} \sum_{(\dot{s}\sim\ddot{s})\in s} \bar{\ell}_{u^{(l)}_{\dot{s},\ddot{s}}}(z^{(l)}_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta^\dagger), \tag{5.80}$$

where the fidelity of the estimator depends on the mixing properties of $\pi(z_{(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}}, \theta)$ and the sample size $\hat{l}$.

5.5.3 M-Step

Having carried out the approximate E-step, we proceed to maximize the estimated lower bound functions by solving the optimization problems

$$\operatorname*{argmax}_{\theta^\dagger} \bar{Q}(\theta^\dagger). \tag{5.81}$$

If we assume that $\pi(\theta)$, μ_θ and σ_θ are continuously differentiable in θ, as is usually the case in applications, $\bar{Q}(\theta^\dagger)$ is also continuously differentiable and it is easily optimized with the help of a numerical optimization routine, e.g. BFGS.

5.5.4 Standard Error Estimation

From the perspective of maximum likelihood estimation, the natural companion problem to obtaining the MLE $\theta^\ddagger$ is to provide its standard errors. Such analysis of the MLE is justified by a range of conditions on both the observation sampling design and the transition density of the model to ensure that the MLE is asymptotically normal around the true value θ_0, see e.g. [3]. Assuming those conditions are met, the sampling covariance of the MLE is close to the inverse *Fisher information matrix*, i.e.

$$\mathrm{Cov}\left[\theta^\ddagger\right] \approx \left(\mathrm{E}\left[-\nabla^2_\theta \log \pi(V_{s\setminus\{0\}}|v_0, \theta)\Big|_{\theta=\theta_0}\right]\right)^{-1}, \tag{5.82}$$

where the covariance and expectation operators are applied with respect to the data generating process. The natural estimate thereof is the *observed information*, replacing the expectation by the observed data, i.e.

$$\mathcal{I}_{\text{obs}}(\theta) = -\nabla_\theta^2 \log \pi(v_{s\setminus\{0\}}|v_0, \theta), \tag{5.83}$$

noting that the estimate is obtained by evaluating the observed information at $\mathcal{I}_{\text{obs}}(\theta^\ddagger)$. Since $\pi(v_{s\setminus\{0\}}|v_0, \theta)$ is of course intractable, we follow [84] in constructing an estimator of the observed information on the basis of the MCEM output. We first define the *complete* and *missing information*

$$\mathcal{I}_{\text{com}}(\theta) = \mathrm{E}_Z\left[-\nabla_\theta^2 \log \pi(v_{s\setminus\{0\}}, Z|v_0, \theta)|v_s, \theta\right] = -\nabla_\theta^2 Q(\theta, \theta), \tag{5.84}$$

$$\begin{aligned}\mathcal{I}_{\text{mis}}(\theta) &= \mathrm{Var}_Z\left[\nabla_\theta \log \pi(v_{s\setminus\{0\}}, Z|v_0, \theta)|v_s, \theta\right] \\ &= \mathrm{E}_Z\left[(\nabla_\theta \log \pi(v_{s\setminus\{0\}}, Z|v_0, \theta) - \nabla_\theta Q(\theta, \theta))^2|v_s, \theta\right],\end{aligned} \tag{5.85}$$

noting that expectation and variance are applied with respect to the bridge update $\pi(Z|v_s, \theta)$. Under mild conditions given by [84], we then decompose the observed information as

$$\mathcal{I}_{\text{obs}}(\theta^\ddagger) = \mathcal{I}_{\text{com}}(\theta^\ddagger) - \mathcal{I}_{\text{mis}}(\theta^\ddagger). \tag{5.86}$$

The complete and missing information are again intractable, but we may use the output of last step of the MCEM algorithm to construct estimators thereof. For the complete information,

$$-\nabla_\theta^2 \bar{Q}(\theta)\Big|_{\theta=\theta^\ddagger} \tag{5.87}$$

is an unbiased estimator of $\mathcal{I}_{\text{com}}(\theta^\ddagger)$, while for the missing information $\mathcal{I}_{\text{mis}}(\theta^\ddagger)$, the sample variance

$$\frac{1}{\hat{l}-1}\sum_{l=1}^{\hat{l}}\left(\sum_{(\dot{s}\sim\ddot{s})\in s}\nabla_\theta \bar{\ell}_{u^{(l)}_{\dot{s},\ddot{s}}}(z^{(l)}_{(\dot{s},\ddot{s})}, v_{\ddot{s}}|v_{\dot{s}}, \theta^\dagger)\Big|_{\theta=\theta^\ddagger} - \nabla_\theta \bar{Q}(\theta)\Big|_{\theta=\theta^\ddagger}\right)^2 \tag{5.88}$$

would be unbiased if the MCEM samples were independent. Adjustments can be made by using (2.46), otherwise the estimate may be seen as a lower bound on the missing information.

5.6 Bayesian Prediction

Where the parameters θ are not of intrinsic interest, the goal of the modelling exercise often consists of ascertaining the expectation $\mathrm{E}\left[f(V_{\text{fut}})|v_{\text{pres}}\right]$ of a test function f of the diffusion at some future date, conditioning on the last known state of the diffusion. A typical example is the pricing of European options. If V is the price process of the underlying asset, then the expected payoff at expiration of a European call with

expiration time "fut" and strike price "strike" is $\mathrm{E}\left[\max\left[V_{\mathrm{fut}} - \mathrm{strike}, 0\right] | v_{\mathrm{pres}}\right]$. The option price corresponds to the discounted expected value.

In the Bayesian formalism, uncertainty about θ is explicitly incorporated into the expected value, in accordance with the full probability model:

$$\mathrm{E}\left[f(V_{\mathrm{fut}})|v_{\mathrm{pres}}\right] = \iint f(v_{\mathrm{fut}})\pi(v_{\mathrm{fut}}|v_{\mathrm{pres}},\theta)\pi(\theta|v_s)\,\mathrm{d}\theta\,\mathrm{d}v_{\mathrm{fut}}, \tag{5.89}$$

where we presume that fut $>$ pres $\geq \omega$, and our uncertainty about θ is captured by its posterior density $\pi(\theta|v_s)$. While this expectation is of course intractable in general, we have developed all the necessary material to construct a Monte Carlo estimator. We may use one of the posterior inference algorithms developed in this chapter to sample Θ, and the forward simulation method of Section 4.1 to sample $V_{\mathrm{fut}}^{(k)}$. We then obtain a sample from the distribution of $f(V_{\mathrm{fut}})$ by evaluating $f(v_{\mathrm{fut}}^{(k)})$, and an unbiased estimator by averaging over $\hat{k}$ samples:

$$\hat{k}^{-1}\sum_{k=1}^{\hat{k}} f(v_{\mathrm{fut}}^{(k)}), \quad v_{\mathrm{fut}}^{(k)} \sim \pi(v_{\mathrm{fut}}|v_{\mathrm{pres}},\theta^{(k)}), \quad \theta^{(k)} \sim \pi(\theta|v_s). \tag{5.90}$$

No such general solution is available for estimation of $\mathrm{E}\left[f(V_{(\mathrm{pres,fut}]})|v_{\mathrm{pres}}\right]$ on the basis of exact algorithms, i.e. when pricing American options, though in special cases exact estimation is still possible, e.g. using the Poisson estimator of Section 4.2.1.

5.7 Bayesian Model Evaluation

A standard Bayesian approach to model evaluation consists of specifying [51] The quality of a prediction is assessed according to a *score function* $\varsigma(v_{\mathrm{fut}}, \pi_{v_{\mathrm{pres}}\to v_{\mathrm{fut}}})$, where $\pi_{v_{\mathrm{pres}}\to v_{\mathrm{fut}}}$ is the conditional model for V_{fut} specified in Section 5.6. We seek to maximize the expected score

$$\mathrm{E}_{V_{\mathrm{pres}},V_{\mathrm{fut}}}\left[\varsigma(V_{\mathrm{fut}}, \pi_{V_{\mathrm{pres}}\to V_{\mathrm{fut}}})\right], \tag{5.91}$$

which averages over the true data-generating process. A "good" model is a model for which the expected score is large. Using a set $v_{s^\ddagger}$ of held-out data, we obtain an unbiased estimator:

$$(|s|-1)^{-1}\sum_{(\dot{s}\sim\ddot{s})\in s^\ddagger} \varsigma(v_{\ddot{s}}, \pi_{v_{\dot{s}}\to v_{\ddot{s}}}). \tag{5.92}$$

The default choice for ς is the *log score* for which $\varsigma(v_{\mathrm{fut}}, \pi_{v_{\mathrm{pres}}\to v_{\mathrm{fut}}}) = \log\pi(v_{\mathrm{fut}}|v_{\mathrm{pres}}, v_s)$. Another option is the more easily estimated *continuous ranked probability score* (CRSP), given by

$$\varsigma(v_{\mathrm{fut}}, \pi_{v_{\mathrm{pres}}\to v_{\mathrm{fut}}}) = 2^{-1}\,\mathrm{E}\left[|A-B|\right] - \mathrm{E}\left[|A - v_{\mathrm{fut}}|\right], \tag{5.93}$$

$$A, B \sim \pi_{v_{\mathrm{pres}}\to v_{\mathrm{fut}}}, \qquad A \perp B \tag{5.94}$$

The CRSP is a *strictly proper scoring rule* in the language of [51], and it can be estimated without bias given samples from $\pi(v_{\text{fut}}|v_{\text{pres}}, v_s)$ at log-linear complexity in the number of samples [59]. It may be seen as a "pragmatic" scoring rule that rewards predictions close to v_{fut}, rather than requiring large support exactly at v_{fut}. Another alternative is the following proper scoring rule that only depends on the predictive mean $\mu_{v_{\text{pres}}\to v_{\text{fut}}}$ and predictive SD $\varsigma_{v_{\text{pres}}\to v_{\text{fut}}}$:

$$\varsigma(v_{\text{fut}}, \pi_{v_{\text{pres}}\to v_{\text{fut}}}) = -\varsigma^{-2}_{v_{\text{pres}}\to v_{\text{fut}}}(v_{\text{fut}} - \mu_{v_{\text{pres}}\to v_{\text{fut}}})^2 - \log \varsigma^2_{v_{\text{pres}}\to v_{\text{fut}}}. \tag{5.95}$$

This rule corresponds to the log score when $\pi_{v_{\text{pres}}\to v_{\text{fut}}}$ is Gaussian.

5.8 Simulation Study

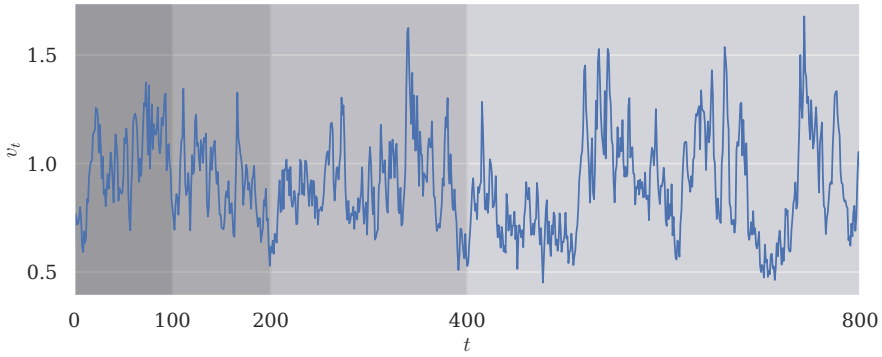


Figure 5.4: Input time series for the extension regime, generated according to the logistic growth model with parameters $(\beta, \kappa, \rho) = (1, 1, 1/8)$. The darkest region corresponds to the smallest input series, with lighter regions being appended successively to obtain the larger input series.

In this section, we venture to explore the scaling behavior of our methods in two regimes:

- The *extension regime*, where we append further data to the end of the time series, at constant observation frequency. This regime is akin to the large-n regime for IID data.
- The *infill regime*, where we increase the observation frequency such that mesh $s \to 0$.

We study the two regimes in the context of the *logistic growth model*, usually applied to population dynamics and defined by the SDE

$$\mathrm{d}V_t = \rho V_t(\beta(1 - V_t/\kappa)\,\mathrm{d}t + \mathrm{d}W_t), \qquad (\beta, \kappa, \rho > 0) \tag{5.96}$$

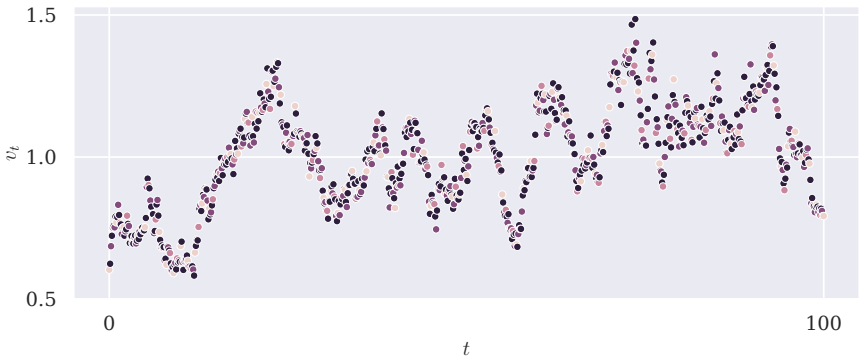


Figure 5.5: Input time series for the infill regime, generated according to the logistic growth model with parameters $(\beta, \kappa, \rho) = (1, 1, 1/8)$. The lightest dots correspond to the slowest observation frequency, with darker dots filled in to obtain the higher observation frequencies.

where ρ is a scale parameter, β is the reproduction rate and κ^{-1} is the *carrying capacity* of the environment. We set the following prior distribution on θ:

$$\log \beta, \log \kappa, \log \rho \sim \mathrm{N}\,[0, 1] \tag{5.97}$$

This is a fairly heavy-tailed (lognormal) prior which somewhat discourages computationally problematic values at the edges of the parameter space. The model is easily implemented by way of a symbolic prexprocessor, as described in Chapter 8, requiring only the following code snippet:

```
v = sp.symbols('v', positive=True)
x = sp.symbols('x', real=True)
b, k, r = sp.symbols('b k r', positive=True)
thi = sp.Array([b, k, r])
mu = b * r * v * (1 - k * v)
sig = r * v
```

Therefore, further manual analysis of the model is not necessary for the purpose of implementation.

We follow the efficiency notion of average CPU time per effective sample (T/ES) set out in Section 2.4 and (2.45), and estimate it from the output of the MCMC algorithm. Both the average time per iteration (T/I) and the average number of iterations per effective sample (I/ES) are estimated from MCMC output for various statistics. The computational cost is part deterministic and part random, with either part affected differently in the scaling regimes. It is clear that the deterministic part of the cost per iteration is linear in the number of observations in both regimes. For the extension regime, the optimistic scenario is that random costs remain linear in expectation, while the effective sample size remains constant. For the infill regime, we note that random

costs depend on the length of the time series and the uncertainty about the diffusion bridges. Since the length of the series is constant but uncertainty is reduced, random costs should decrease. Clearly, conclusions from those experiments have limited external validity, and should be seen as setting a benchmark for the behavior of the algorithms under favorable circumstances, i.e. for models that are fairly smooth in θ and exhibit sufficient posterior concentration rates. Other than tracking these performance metrics for the elements of θ, we may also investigate the mixing of a more global statistic. For comparability between algorithms with different state spaces, we evaluate the density

$$\pi(v_{s\setminus\{0\}}, \breve{z}^{(k)}, \breve{\psi}^{(k)}, \theta^{(k)}|v_0) = \pi(\theta) \prod_{(\dot{s}\sim\ddot{s})\in s} \pi(v_{\ddot{s}}, \breve{z}^{(k)}_{(\dot{s},\ddot{s})}, \breve{\psi}^{(k)}_{(\dot{s},\ddot{s})}|\theta^{(k)}, v_{\dot{s}}), \tag{5.98}$$

where $\theta^{(k)}$ is the value at MCMC iteration k, and $\breve{z}^{(k)}_{(\dot{s},\ddot{s})}$ and $\breve{\psi}^{(k)}_{(\dot{s},\ddot{s})}$ are random samples from $\mathbb{B}_{(\dot{s},\ddot{s})}$ and $\mathbb{P}_{(\dot{s},\ddot{s})}$, rather than being taken from the MCMC chain. Therefore, by Theorem 14,

$$\mathrm{E}_{\mathbb{B}_{(\dot{s},\ddot{s})}\times\mathbb{P}_{(\dot{s},\ddot{s})}}\left[\pi(v_{s\setminus\{0\}}, \breve{Z}, \breve{\Psi}, \theta^{(k)}|v_0)\right] = \pi(v_{s\setminus\{0\}}, \theta^{(k)}|v_0), \tag{5.99}$$

i.e. the summary has a natural interpretation as an unbiased estimator of the joint density on the minimal state space Θ. This quantity allows us to assess whether the algorithm is mixing adequately in terms of model fit, while avoiding the storage of the large objects Z and Ψ during the run of the algorithm.

Each MCMC run that contributes to this section's results consists of 100000 iterations, of which we discard 10000 for burn-in. We precede the exact MCMC run by an approximate run of 10000 iterations according to the algorithm of Section 5.4. We target an acceptance probability of 23.4% for Metropolis-within-Gibbs steps, and 25% for Barker-within-Gibbs, with a Portkey probability of 1%. Step sizes are adapted according to the *Adapting Increasingly Rarely* (AIR) method of [25]. This method is a variant of the most commonly used stochastic optimization methods for MCMC tuning, improving their practical and especially theoretical properties by modifying the tuning parameters at increasing intervals, rather than at every iteration.

For Poisson coin simulations within the marginal algorithm we adopt the limiting batch EA version of Section 4.1.3. When the integrand bounds in Poisson estimator simulations within the auxiliary algorithm exceed 10000, the proposal is rejected to avoid memory errors. Such events occur a few times in the auxiliary simulations, and while they represent a small departure from exactness, proposals implying such large bounds would usually result in rejection even if the full simulation were to be carried out. Nevertheless, in this instance, the marginal version is fully exact as the batch EA method avoids carrying out excessively expensive simulations without loss of exactness.

5.8.1 Extension Regime

For the extension regime, we run the marginal and augmented algorithms on a time series of 100, 200, 400 and 800 observations respectively, with an inter-observation interval of

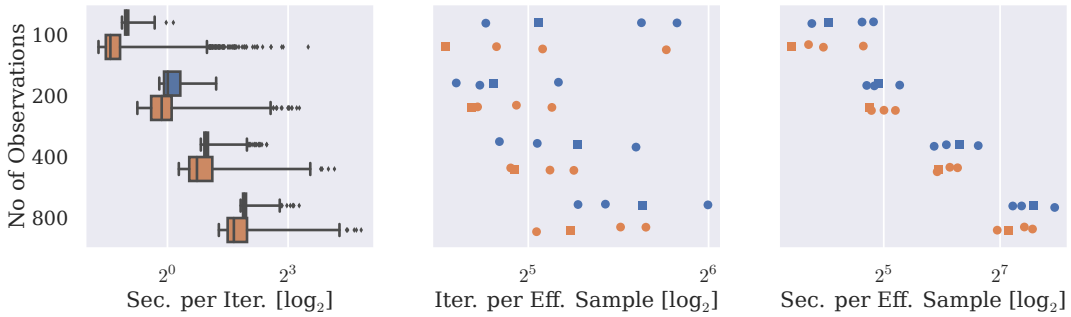


Figure 5.6: Sampling efficiency in the extension regime for the auxiliary (blue) and marginal (orange) algorithms. The left panel shows the distribution of CPU time per iteration (T/I) throughout the MCMC run. The medium panel shows estimates of the required number of MCMC iterations to generate an effective sample (I/ES), where each dot corresponds to an element of Θ, and the square to the fit metric defined in (5.98). The right panel shows estimates of the required CPU time to generate an effective sample (T/ES). Notice that the right panel is obtained by scaling the middle panel by the mean of the distributions in the left panel.

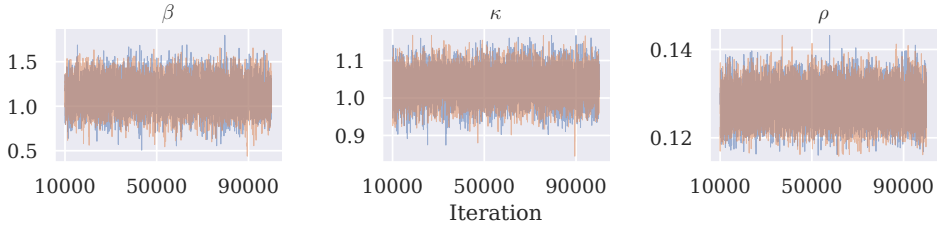


Figure 5.7: Trace plots of Θ for the 800-observation time series in the extension regime for the auxiliary (blue) and marginal (orange) algorithms.

1. The series is generated from the logistic growth model with parameters $(\beta, \kappa, \rho) = (1, 1, 1/8)$ and plotted in Figure 5.4.

We observe very similar performance between the marginal and the auxiliary algorithms. Figure 5.6 provides various performance summaries, showing that the algorithms are similar in their T/I and I/ES, and accordingly in their T/ES. T/I variance is higher for the marginal algorithm, with a pronounced right tail. We deem the measurements in Figure 5.6 to be consistent with a linear scaling of T/ES in the number of observations. T/I appears to follow the linear scaling particularly closely, while I/ES shows no clear upward trend. For a less quantitative, but more robust and transparent assessment of efficiency, Figure 5.7 displays the full traces for Θ in the 800-observation run.

For all elements of Θ, we observe concentration of the posterior around the true simulation values $(\beta, \kappa, \rho) = (1, 1, 1/8)$, see Figure 5.9. The concentration rate for κ is

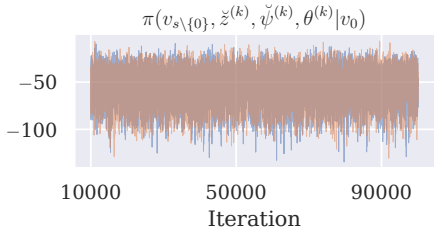


Figure 5.8: Trace plot of (5.98) for the 800-observation time series in the extension regime for the auxiliary (blue) and marginal (orange) algorithms.

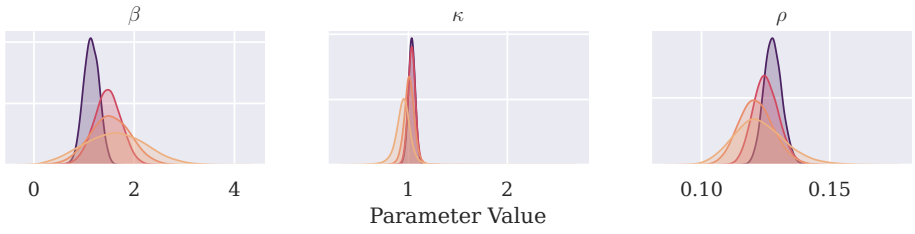


Figure 5.9: Posterior marginals of Θ in the extension regime, as estimated by a KDE. Darker shades correspond to a larger observation number.

slightly slower at the longer time series end, which could be due to random idiosyncrasies in the simulated data. Nonetheless, concentration rates are sufficient to keep T/ES approximately linear, while I/ES remains constant.

5.8.2 Infill Regime

For the infill regime, we interpolate the first 100 observations used in the extension experiment at frequencies 2, 4 and 8, with identical parameters $(\beta, \kappa, \rho) = (1, 1, 1/8)$. The resulting observations are plotted in Figure 5.5.

In this instance, we observe a clear improvement of the marginal over the auxiliary algorithm in the higher observation frequency range, see Figure 5.10. The auxiliary algorithm appears linear in T/I and constant in I/ES, while the marginal algorithm appears sublinear in T/I and possibly decreasing in I/ES. Therefore, we conjecture that the auxiliary algorithm is linear and the marginal algorithm sublinear in T/ES. For completeness, we again provide full traces for Θ in Figure 5.11.

Unlike the extension regime, the infill regime only results in the concentration of the ρ-marginal, see Figure 5.13. Therefore, the higher frequency is only informative about the scale of the drift, while the information about the other drift parameters is already reflected in the slowest frequency.

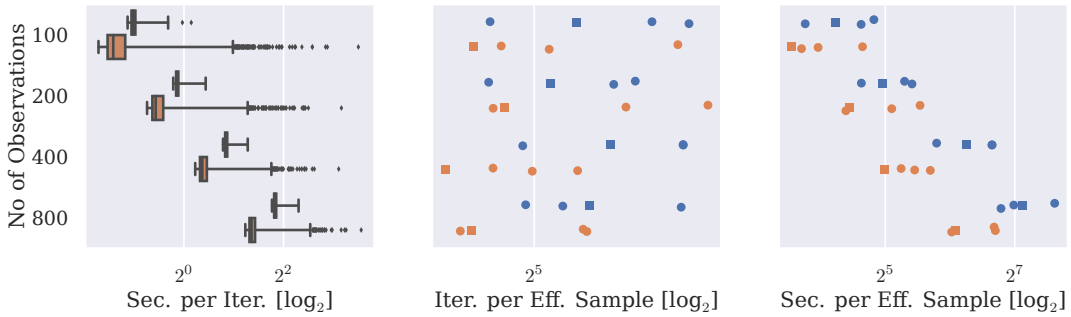


Figure 5.10: Sampling efficiency in the infill regime for the auxiliary (blue) and marginal (orange) algorithms. The left panel shows the distribution of CPU time per iteration (T/I) throughout the MCMC run. The medium panel shows estimates of the required number of MCMC iterations to generate an effective sample (I/ES), where each dot corresponds to an element of Θ, and the square to the fit metric defined in (5.98). The right panel shows estimates of the required CPU time to generate an effective sample (T/ES). Notice that the right panel is obtained by scaling the middle panel by the mean of the distributions in the left panel.

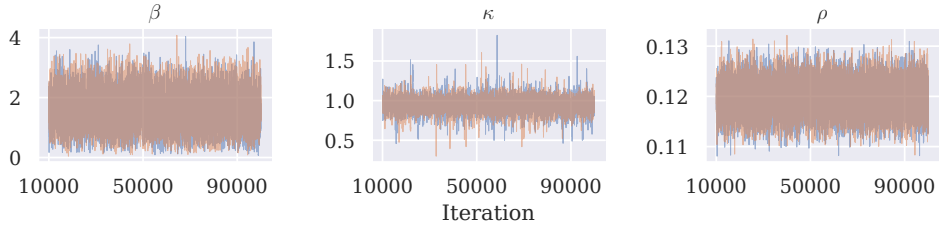


Figure 5.11: Trace plots of Θ for the 800-observation time series in the infill regime for the auxiliary (blue) and marginal (orange) algorithms.

5.9 Discussion

Overall, on the basis of both simulation experiments and theoretical arguments, we cannot express strong recommendations for either the marginal or the auxiliary marginal approach. Properly implemented and configured, both methods can reliably scale in both extension and infill regimes, with no departure from exactness for the marginal algorithm, and minimal departure for the auxiliary one. There is some evidence of a minor scaling edge for the marginal algorithm in the infill regime. Both algorithms rely on complex foundations, with the additional groundwork on Bernoulli factories in the marginal case, and the need to properly specify the nonunique 2-coin factorization. Therefore, in terms of parsimony, the auxiliary algorithm is slightly more accessible.

In the light of Remark 1, we emphasize that the average iteration time appears well-

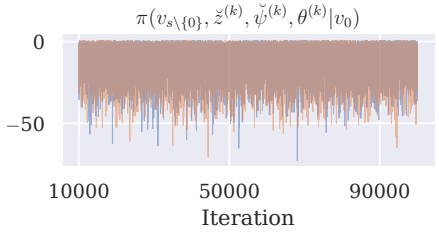


Figure 5.12: Trace plot of (5.98) for the 800-observation time series in the infill regime for the auxiliary (blue) and marginal (orange) algorithms.

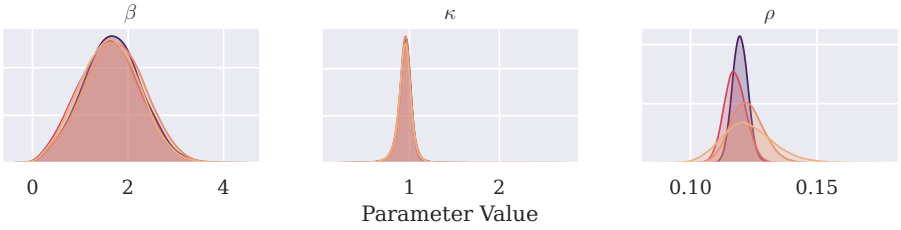


Figure 5.13: Posterior marginals of Θ in the infill regime, as estimated by a KDE. Darker shades correspond to a larger observation number.

behaved, and the presented algorithms mostly avoid regions of the algorithms' state space in which T/I is prohibitively large. This is due to a combination of mitigation techniques, including warm starting. In the case of the marginal algorithm, batch EA, portkey Barker and the new 2-coin algorithm of Section 5.2.2 are essential. While definitive statements are difficult to make for general diffusion models, we believe that the algorithms of this chapter are well-behaved when the posterior concentrates sufficiently quickly, and exludes regions of $\mathcal{T}$ where φ_θ is large or highly variable. In summary, both frameworks are adequate foundations for the development of MCMC algorithms for more complex models, as seen in the following Chapters 6 and 7.

6 Exact Inference for Markov Switching Diffusion Models

The primary limitation of the Itō diffusion models that we have focused on up to now is that they are *time-invariant*, i.e. the dynamics specified by the SDE do not depend on the clock index t. Nonetheless, many stochastic phenomena are best modelled by dynamics whose parameters are allowed to depend on a time changing regime.

Our regime-switching framework is as follows. Let Y be a latent, discrete space, continuous time Markov jump process with states $\left\{1, \dots, \widehat{k}\right\}$ and denote the set of possible trajectories by $\mathcal{Y}$. The jump process evolves according to its generator matrix λ, where $\lambda_{i,j\neq i} \geq 0$ are the jump rates from state i to j and the diagonal elements are given by $\lambda_{ii} = -\sum_{j\neq j} \lambda_{ij}$. We follow the common convention of denoting the exit rates $\lambda_i = -\lambda_{ii}$. The density function of Y is defined with respect to the measure $\mathbb{L}$ induced by a rate 1 marked Poisson process, and given by

$$\pi(y|\lambda) = \exp\left[\int_0^\omega (1 - \lambda_{y_t})\,\mathrm{d}t\right] \prod_{(\dot{r}\sim\ddot{r})\in r\cup\{0\}} \lambda_{y_{\dot{r}} y_{\ddot{r}}}, \qquad (y \in \mathcal{Y}) \tag{6.1}$$

where r is the set of times at which the trajectory y changes values. The latent process controls the dynamics of the observable process V by way of the Markov switching SDE

$$\mathrm{d}V_t = \mu_\theta(V_t, Y_t)\,\mathrm{d}t + \sigma_\theta(V_t)\rho_\theta(Y_t)\,\mathrm{d}W_t, \qquad (V_0 = v_0, \quad V_0 \perp Y_0) \tag{6.2}$$

where W is a standard Brownian motion and $\theta \in \mathcal{T}$ is a parameter vector. Figures 6.1 and 6.2 show example trajectories from V and Y. Suppose that the SDE admits a unique solution for every $y \in \mathcal{Y}$ and θ and therefore a Markov transition density $\pi(v_{t+\epsilon}|v_t, y, \theta)$, which we assume to be intractable. As usual, let V be observed at times s with values v_s, while all other quantities are unknown, i.e. θ and λ denote variates of the random variables Θ and Λ. Our primary aim is to generate samples from the exact posterior

$$\pi(y, \theta, \lambda|v_s) = \frac{\pi(\theta)\pi(\lambda)\pi(y|\lambda)\prod_{(\dot{s}\sim\ddot{s})\in s}\pi(v_{\ddot{s}}|v_{\dot{s}}, y_{[\dot{s},\ddot{s})}, \theta)}{\iiint \pi(\theta')\pi(\lambda')\pi(y'|\lambda')\prod_{(\dot{s}\sim\ddot{s})\in s}\pi(v_{\ddot{s}}|v_{\dot{s}}, y'_{[\dot{s},\ddot{s})}, \theta')\mathbb{L}(\mathrm{d}y')\,\mathrm{d}\theta'\,\mathrm{d}\lambda'} \tag{6.3}$$

for a given product prior $\pi(\theta, \lambda) = \pi(\theta)\pi(\lambda)$. The factorization of the volatility term is necessary for technical reasons discussed in Section 6.1.1. It need not be unique and this arbitrariness does not affect the algorithms presented in this chapter.

Discrete time versions of such models are common in economics and finance, where they were first proposed to infer business cycles from GDP growth data [56]. Other economic

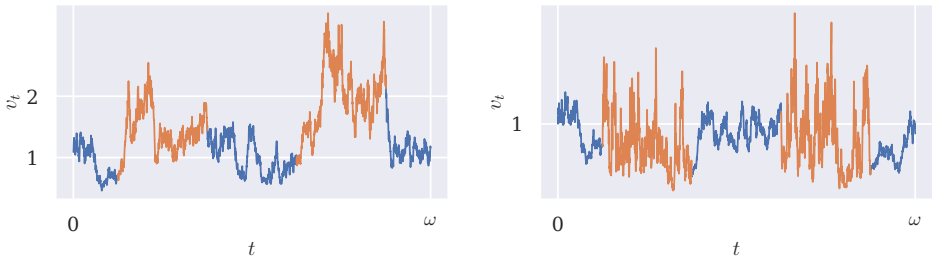


Figure 6.1: Illustration of a mean switching and a volatility switching time series. The left series follows $\mathrm{d}V_t = (1/8)(V_t(1 - V_t/\kappa_{Y_t})\,\mathrm{d}t + \mathrm{d}W_t)$ where $\kappa_1 = 1$ (blue) and $\kappa_2 = 2$ (orange). The right series follows $\mathrm{d}V_t = \rho_{Y_t}(V_t(1-V_t)\,\mathrm{d}t + \mathrm{d}W_t)$, where $\rho_1 = 1/8$ (blue) and $\rho_2 = 1/2$ (orange). We observe the diffusion discretely.

Figure 6.2: Illustration of the regime trajectory corresponding to Figure 6.1.

time series exhibiting cyclical regime shifts include exchange rates [36], interest rates [23], stock prices [57], commodity prices [41] and energy prices [93]. Regime switching processes also lend themselves to modelling structural breaks in economic regimes, such as in [74, 87]. While discrete time models are dominant in econometrics, we are interested in their continuous-time formulation given above, for which few inference methods are available. Mathematicians have long investigated stability and optimal control of such models [48, 10, 85]. Just as the transition density of standard Itō diffusion models is typically intractable, so is the transition density of a Markov switching model. Therefore, the statistical literature relies on discrete approximations of the switching diffusion [65, 81], a limitation which we seek to address. We also note the related literature which relaxes the recurrent Markovian assumption on the latent process, see e.g. [73, 38].

This chapter largely mimics the structure of the previous Chapter 5 on Itō diffusion inference. We begin by setting out an explicit joint model $\pi(v, y, \theta, \lambda)$ involving the diffusion and the regime path. Gibbs sampling on that space will require an appropriate reparameterization. The infinite dimensional terms may be addressed by a marginal Bernoulli factory MCMC algorithm, or an auxiliary algorithm that relies on additional nuisance variables. Sections 6.3 and 6.4 present analogous marginal and auxiliary algorithms, Section 6.5 discusses approximate Bayesian inference and Section 6.6 proposes an MCEM algorithm for MAP inference. All algorithms in this section are fully novel.

In conjunction with the novel aspects of the previous chapter, this chapter seeks to

extend the methodology for diffusion inference to a more complex and challenging class of models. Therefore, it represents the main payoff from the work underlying this thesis. While the approach of this chapter is similar to the previous one, the computational problem is much harder. On the one hand, we seek to explore a posterior that is higher-dimensional, and often exhibits new dependencies and multiple modes. Any MCMC method struggles with such geometries. Moreover, Markov switching diffusions can exhibit strong drift discontinuities when Y changes states, which is precisely where our retrospective simulation methods are most expensive, as laid out by Remark 1.

6.1 Data Augmentation Strategy

The underlying theme of this section is to cast the data augmentation strategy in terms of the simple Itō diffusion setting to the largest extent possible. We can do so by observing that conditional on a regime trajectory y with transition times r and the corresponding diffusion values v_r, V is independent of Λ, and by the Markov property $\pi(v_r, v_{s\setminus\{0\}}|v_0, y, \theta)$ factorizes as follows:

$$\pi(v_{\tau\setminus\{0\}}|v_0, y, \theta) = \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta), \tag{6.4}$$

where we define the ordered *set of event times* $\tau = r \cup s$. Since y is by definition constant within $(\dot{\tau}, \ddot{\tau})$, $\pi(v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)$ is the law of a time-invariant diffusion process. Therefore, if we include Y and V_r in the augmented model, we will be able to port many arguments and techniques with only minor modifications. We will also introduce the inter-event diffusion bridges $V_{(\dot{\tau},\ddot{\tau})}$ in Section 6.1.1, resulting in the extended posterior

$$\pi(v, y, \theta, \lambda|v_s) \propto \pi(\theta)\pi(\lambda) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(v_{(\dot{\tau},\ddot{\tau}]}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}}|y_{\dot{\tau}}, \lambda). \tag{6.5}$$

Then, in the accustomed manner, in Section 6.1.2 we reparameterize the bridges $V_{(\dot{\tau},\ddot{\tau})}$ to obtain a model in terms of the noncentered bridges $z_{(\dot{\tau},\ddot{\tau})}$:

$$\pi(v_r, z, y, \theta, \lambda|v_s) \propto \pi(\theta)\pi(\lambda) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}}|y_{\dot{\tau}}, \lambda) \tag{6.6}$$

Under this parameterization, we can design an ergodic marginal Gibbs sampler with full conditionals $\pi(\theta, \lambda|v_\tau, z, y)$ and $\pi(v_r, z, y|v_s, \theta)$. We will also consider an extended, finite dimensional version with auxiliary stochastic process Ψ in Section 6.1.3, targeting the posterior

$$\pi(v_r, \psi, z, y, \theta, \lambda|v_s) \propto \pi(\theta)\pi(\lambda) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(\psi_{(\dot{\tau},\ddot{\tau})}, z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}}|y_{\dot{\tau}}, \lambda). \tag{6.7}$$

Since the development largely parallels Chapter 5, the reader is best served by first reading Section 5.1, which introduces critical concepts with lighter notation.

6.1.1 Standing Assumptions and Complete Transition Density

Conditional on θ and y, the complete transition density follows from a slightly extended version of the argument in Section 5.1.1. We define the transformation

$$\eta_\theta(a) = \int_{v^*}^{a} \frac{\mathrm{d}b}{\sigma_\theta(b)}, \qquad (v^*, a \in \mathcal{V}) \tag{6.8}$$

which yields the reduced process $X = \eta_\theta(V)$ with SDE

$$\mathrm{d}X_t = \delta_\theta(X_t, y_{\dot\tau})\,\mathrm{d}t + \rho_\theta(y_{\dot\tau})\,\mathrm{d}W_t, \qquad (X_0 = \eta_\theta(v_0), \quad t \in [\dot\tau, \ddot\tau)) \tag{6.9}$$

$$\delta_\theta(a, b) = \left(\frac{\mu_\theta(\cdot, b)}{\sigma_\theta} - \frac{\sigma_\theta'}{2}\right) \circ \eta_\theta^{-1}(a). \tag{6.10}$$

Notice that while $\delta_\theta(X_t, y_{\dot\tau})$ is discontinuous at times r, X itself remains continuous. This would not be the case if we allowed general volatility functions $\sigma_\theta(v_t, y_t)$ and used the discontinuous transformation $\eta_\theta(v_t, y_t) = \int \frac{\mathrm{d}v_t}{\sigma_\theta(v_t, y_t)}$. The transformed process has no tractable dominating process, substantially complicating the development of an MCMC algorithm. Therefore, we limit ourselves to studying Markov switching diffusions with the volatility factorization $\sigma_\theta(v_t)\rho_\theta(y_t)$.

We require throughout that for any $\theta \in \mathcal{T}$, $b \in \{1, \ldots, k\}$ and $\dot\tau < \ddot\tau$,

- $\delta_\theta(a, b)$ is continuously differentiable in a on $\mathcal{V}$.
- The *Novikov condition* applies, i.e. $\mathrm{E}_{X_{(\dot\tau, \ddot\tau]}}\left[\exp\left[\int_{\dot\tau}^{\ddot\tau} \delta_\theta^2(X_t, b)\,\mathrm{d}t\right] | x_{\dot\tau}, \{y_{\dot\tau} = b\}, \theta\right] < \infty$. This is sufficient, albeit not necessary.

Since in this instance X is not a unit-volatility process, the derivation from the Itō diffusion case is slightly modified. Let $\mathbb{X}|(x_{\dot\tau}, y_{\dot\tau}, \theta)$ be induced by $X_{(\dot\tau, \ddot\tau]}$ for $X_{\dot\tau} = x_{\dot\tau}$ and $Y_{\dot\tau} = y_{\dot\tau}$. Furthermore, let $\mathbb{M}|(x_{\dot\tau}, y_{\dot\tau}, \theta)$ be the driftless measure induced by $\mathrm{d}X_t = \rho_\theta(y_{\dot\tau})\,\mathrm{d}W_t$. Then, by Theorem 3.2.5, $\mathbb{M}|(x_{\dot\tau}, y_{\dot\tau}, \theta) \gg \mathbb{X}|(x_{\dot\tau}, y_{\dot\tau}, \theta)$, and the RND between the two measures is

$$\frac{\mathrm{d}\mathbb{X}|(x_{\dot\tau}, y_{\dot\tau}, \theta)}{\mathrm{d}\mathbb{M}|(x_{\dot\tau}, y_{\dot\tau}, \theta)}(x_{(\dot\tau, \ddot\tau]}) = \exp\left[\int_{\dot\tau}^{\ddot\tau} \frac{\delta_\theta(x_t, y_{\dot\tau})}{\rho_\theta(y_{\dot\tau})}\,\mathrm{d}W_t + \frac{1}{2}\int_{\dot\tau}^{\ddot\tau} \frac{\delta_\theta^2(x_t, y_{\dot\tau})}{\rho_\theta^2(y_{\dot\tau})}\,\mathrm{d}t\right]. \tag{6.11}$$

For the rest of the derivation, the logic of Theorem 10 applies almost unchanged, with slightly modified definitions, and gives us the complete transition density with respect

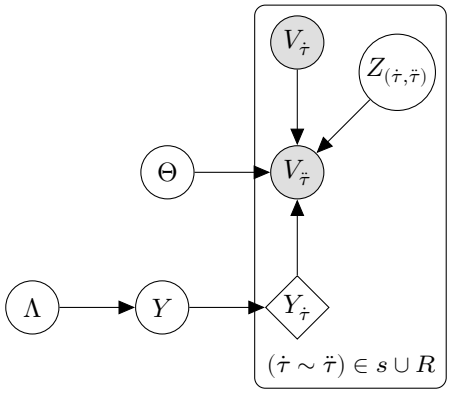


Figure 6.3: Plate diagram for the marginal noncentered model. $V_{\dot\tau}$ and $V_{\ddot\tau}$ may be observed or latent, depending on whether $\dot\tau, \ddot\tau \in s$.

to the dominating measure $\mathbb{M}|(X_{\{\dot\tau,\ddot\tau\}} = \eta_\theta(v_{\{\dot\tau,\ddot\tau\}}), y_{\dot\tau}, \theta) \times \mathrm{Leb}$:

$$\begin{aligned}\pi(x_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta) &= |\eta'_\theta(v_{\ddot\tau})|\,\mathrm{N}\left[\eta_\theta(v_{\ddot\tau}); \eta_\theta(v_{\dot\tau}), (\ddot\tau - \dot\tau)\rho^2_\theta(y_{\dot\tau})\right] \\ &\quad \times \frac{\mathrm{d}\mathbb{X}|(X_{\dot\tau} = \eta_\theta(v_{\dot\tau}), y_{\dot\tau}, \theta)}{\mathrm{d}\mathbb{M}|(X_{\dot\tau} = \eta_\theta(v_{\dot\tau}), y_{\dot\tau}, \theta)}(x_{(\dot\tau,\ddot\tau)}, \eta_\theta(v_{\ddot\tau})),\end{aligned} \tag{6.12}$$

$$\frac{\mathrm{d}\mathbb{X}|(x_{\dot\tau}, y_{\dot\tau}, \theta)}{\mathrm{d}\mathbb{M}|(x_{\dot\tau}, y_{\dot\tau}, \theta)}(x_{(\dot\tau,\ddot\tau]}) = \exp\left[\frac{\Delta_\theta(x_{\ddot\tau}, y_{\dot\tau}) - \Delta_\theta(x_{\dot\tau}, y_{\dot\tau})}{\rho^2_\theta(y_{\dot\tau})} - \int_{\dot\tau}^{\ddot\tau} \varphi_\theta(x_t, y_{\dot\tau})\,\mathrm{d}t\right], \tag{6.13}$$

$$\varphi_\theta(a,b) = \frac{1}{2}\left(\frac{\delta^2_\theta(a,b)}{\rho^2_\theta(b)} + \partial_a\delta_\theta(a,b)\right), \tag{6.14}$$

$$\Delta_\theta(a,b) = \int \delta_\theta(a,b)\,\mathrm{d}a. \tag{6.15}$$

Just as in the Itō diffusion case, we note that for distinct values $\theta \neq \theta^\dagger$, $\mathbb{M}|(X_{\dot\tau} = \eta_\theta(v_{\dot\tau}), y_{\dot\tau}, \theta)$ and $\mathbb{M}|(X_{\dot\tau} = \eta_{\theta^\dagger}(v_{\dot\tau}), y_{\dot\tau}, \theta^\dagger)$ are mutually singular, and therefore $\pi(x_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta)$ and $\pi(x_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta^\dagger)$ are mutually singular as well. Hence, a noncentered parameterization is required.

6.1.2 Marginal Noncentered Transition Density

We proceed to changing variables from the centered bridges $X_{(\dot\tau,\ddot\tau)}$ to noncentered, a priori independent bridges. Again, only minor changes are required compared to the Itō diffusion setting of Section 5.1.2. We define

$$\zeta_\theta(x_t; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}) = \frac{x_t - \eta_\theta(v_{\dot\tau}) - (\eta_\theta(v_{\ddot\tau}) - \eta_\theta(v_{\dot\tau}))\frac{t-\dot\tau}{\ddot\tau-\dot\tau}}{\rho_\theta(y_{\dot\tau})}, \qquad (t \in (\dot\tau, \ddot\tau)) \tag{6.16}$$

and let ζ^{-1}_θ be the inverse in the first argument:

$$\zeta^{-1}_\theta(z_t; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}) = \rho_\theta(y_{\dot\tau})z_t + \eta_\theta(v_{\dot\tau}) + (\eta_\theta(v_{\ddot\tau}) - \eta_\theta(v_{\dot\tau}))\frac{t-\dot\tau}{\ddot\tau-\dot\tau} \qquad (t \in (\dot\tau, \ddot\tau)) \tag{6.17}$$

We change variables to $Z_{(\dot\tau,\ddot\tau)} = \zeta_\theta(X_{(\dot\tau,\ddot\tau)}; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}})$ and note that under $\mathbb{M}|(x_{\dot\tau}, y_{\dot\tau}, \theta)$, $Z_{(\dot\tau,\ddot\tau)}$ is a Brownian bridge spanning the origin at times $(\dot\tau, \ddot\tau)$. We further define $\mathbb{Z}|(x_{\{\dot s,\ddot s\}}, y_{\dot\tau}, \theta)$ and $\mathbb{B}_{(\dot\tau,\ddot\tau)}$ as the pushforward measures induced by $Z_{(\dot\tau,\ddot\tau)}$ under $\mathbb{X}|(x_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau}, \theta)$ and $\mathbb{W}|(x_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau}, \theta)$, respectively. Probabilities being conserved under a change of variable, we find that

$$\begin{aligned}\frac{\mathrm{d}\mathbb{Z}|(x_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau}, \theta)}{\mathrm{d}\mathbb{B}_{(\dot\tau,\ddot\tau)}}(z_{(\dot\tau,\ddot\tau)}) &= \frac{\mathrm{d}\mathbb{X}|(x_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau}, \theta)}{\mathrm{d}\mathbb{M}|(x_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau}, \theta)} \circ \zeta_\theta^{-1}(z_{(\dot\tau,\ddot\tau)}; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}) \\ &= \frac{\mathrm{N}\left[x_{\ddot\tau}; x_{\dot\tau}, (\ddot\tau - \dot\tau)\rho_\theta^2(y_{\dot\tau})\right]}{\pi(x_{\ddot\tau}|x_{\dot\tau}, y_{\dot\tau}, \theta)} \\ &\quad \times \frac{\mathrm{d}\mathbb{X}|(x_{\dot\tau}, y_{\dot\tau}, \theta)}{\mathrm{d}\mathbb{W}|(x_{\dot\tau}, y_{\dot\tau}, \theta)}(\zeta_\theta^{-1}(z_{(\dot\tau,\ddot\tau)}; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}), x_{\ddot\tau}),\end{aligned} \tag{6.18}$$

which, in conjunction with $\pi(x_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta)$, gives us the noncentered complete transition density:

$$\begin{aligned}\pi(z_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta) &= |\eta_\theta'(v_{\ddot\tau})|\,\mathrm{N}\left[\eta_\theta(v_{\ddot\tau}); \eta_\theta(v_{\dot\tau}), (\ddot\tau - \dot\tau)\rho_\theta^2(y_{\dot\tau})\right] \\ &\quad \underbrace{\times \frac{\mathrm{d}\mathbb{X}|(X_{\dot\tau} = \eta_\theta(v_{\dot\tau}), y_{\dot\tau}, \theta)}{\mathrm{d}\mathbb{M}|(X_{\dot\tau} = \eta_\theta(v_{\dot\tau}), y_{\dot\tau}, \theta)}(\zeta_\theta^{-1}(z_{(\dot\tau,\ddot\tau)}; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}), \eta_\theta(v_{\ddot\tau}))}_{d_\theta(v_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau})} \\ &= \left|\eta_\theta'(v_{\ddot\tau})\right|\mathrm{N}\left[\eta_\theta(v_{\ddot\tau}); \eta_\theta(v_{\dot\tau}), (\ddot\tau - \dot\tau)\rho_\theta^2(y_{\dot\tau})\right] e^{\Delta_\theta(\eta_\theta(v_{\ddot\tau}), y_{\dot\tau}) - \Delta_\theta(\eta_\theta(v_{\dot\tau}), y_{\dot\tau})} \\ &\quad \underbrace{\times \exp\left[-\int_{\dot\tau}^{\ddot\tau} \varphi_\theta(\zeta_\theta^{-1}(z_t; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}), y_{\dot\tau})\,\mathrm{d}t\right]}_{q_\theta(z_{(\dot\tau,\ddot\tau)}, v_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau})},\end{aligned} \tag{6.19}$$

where the dominating measure is $\mathbb{B}_{(\dot\tau,\ddot\tau)} \times \mathrm{Leb}$, and

$$\int \pi(z_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta)\mathbb{B}_{(\dot\tau,\ddot\tau)}(\mathrm{d}z_{(\dot\tau,\ddot\tau)}) = \pi(v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta). \tag{6.20}$$

See Figure 6.3 for the corresponding graphical model. As in the previous chapter, we will use the notation

$$\tilde\varphi_\theta(z_t, y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}) = \varphi_\theta(\zeta_\theta^{-1}(z_t; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}), y_{\dot\tau}) \tag{6.21}$$

to denote composition with ζ_θ^{-1} in x_t, with analogous expressions for the bounds $\tilde\varphi_\theta^\downarrow$ and $\tilde\varphi_\theta^\uparrow$. Any noncentered Brownian bridge bounds

$$-\infty < z_{(\dot\tau,\ddot\tau)}^\downarrow \le z_t \le z_{(\dot\tau,\ddot\tau)}^\uparrow < \infty \qquad (t \in (\dot\tau, \ddot\tau)) \tag{6.22}$$

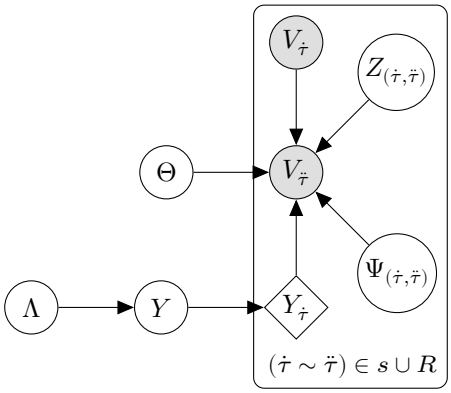


Figure 6.4: Plate diagram for the auxiliary noncentered model. $V_{\dot\tau}$ and $V_{\ddot\tau}$ may be observed or latent, depending on whether $\dot\tau, \ddot\tau \in s$.

are propagated to $x_{(\dot\tau,\ddot\tau)}$ through

$$\begin{aligned} x_t &\in \begin{pmatrix} \rho_\theta(y_{\dot\tau}) z^{\downarrow}_{(\dot\tau,\ddot\tau)} + \eta_\theta(v_{\dot\tau}) + (\eta_\theta(v_{\ddot\tau}) - \eta_\theta(v_{\dot\tau}))(t-\dot\tau)/(\ddot\tau-\dot\tau), \\ \rho_\theta(y_{\dot\tau}) z^{\uparrow}_{(\dot\tau,\ddot\tau)} + \eta_\theta(v_{\dot\tau}) + (\eta_\theta(v_{\ddot\tau}) - \eta_\theta(v_{\dot\tau}))(t-\dot\tau)/(\ddot\tau-\dot\tau) \end{pmatrix} \\ &\in (\underbrace{\rho_\theta(y_{\dot\tau}) z^{\downarrow}_{(\dot\tau,\ddot\tau)} + \eta_\theta(v_{\dot\tau}) \wedge \eta_\theta(v_{\ddot\tau})}_{x^{\downarrow}_{(\dot\tau,\ddot\tau)}}, \underbrace{\rho_\theta(y_{\dot\tau}) z^{\uparrow}_{(\dot\tau,\ddot\tau)} + \eta_\theta(v_{\dot\tau}) \vee \eta_\theta(v_{\ddot\tau})}_{x^{\uparrow}_{(\dot\tau,\ddot\tau)}}). \end{aligned} \tag{6.23}$$

6.1.3 Auxiliary Noncentered Transition Density

We now proceed to deriving a finite-dimensional auxiliary model analogous to Section 5.1.3. Let $\Psi_{(\dot\tau,\ddot\tau)}$ be a 2-dimensional Poisson process on $[\dot\tau,\ddot\tau] \times [0,\infty)$ with induced measure $\mathbb{P}_{(\dot\tau,\ddot\tau)}$, and assume that for every θ and $y_{\dot\tau}$, we have access to upper and lower bounds for $\varphi_\theta(x_t, y_{\dot\tau})$ on $t \in (\dot\tau,\ddot\tau)$. Given these bounds, we define the truncation

$$\gamma_\theta(\psi_{(\dot\tau,\ddot\tau)}, x_{(\dot\tau,\ddot\tau)}, y_{\dot\tau}) = \left\{ t : (t,\phi) \in \psi_{(\dot\tau,\ddot\tau)}, \phi \le (\varphi^{\uparrow}_\theta - \varphi^{\downarrow}_\theta)(x_{(\dot\tau,\ddot\tau)}, y_{\dot\tau}) \right\}. \tag{6.24}$$

Again, $\left|\gamma_\theta(\Psi_{(\dot\tau,\ddot\tau)}, x_{(\dot\tau,\ddot\tau)}, y_{\dot\tau})\right|$ is almost surely finite. By Theorem 13 and analogy to Theorem 14, we obtain the *centered auxiliary complete transition density*

$$\begin{aligned} &\pi(\psi_{(\dot\tau,\ddot\tau)}, x_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau} | v_{\dot\tau}, y_{\dot\tau}, \theta) \\ &\quad = d_\theta(v_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau}) e^{(\dot\tau-\ddot\tau)\varphi_\theta{}^{\downarrow}(x_{(\dot\tau,\ddot\tau)}, y_{\dot\tau})} \prod_{t \in \gamma_\theta(\psi_{(\dot\tau,\ddot\tau)}, x_{(\dot\tau,\ddot\tau)}, y_{\dot\tau})} \left\{ \left(\frac{\varphi^{\uparrow}_\theta - \varphi_\theta}{\varphi^{\uparrow}_\theta - \varphi^{\downarrow}_\theta} \right) (x_{(\dot\tau,\ddot\tau)}, y_{\dot\tau}) \right\}_t . \end{aligned} \tag{6.25}$$

$$Y(\tau_0) \xrightarrow{\text{Exp}(\lambda_{Y(\tau_0)})} Y(\tau_1) \xrightarrow{\text{Exp}(\lambda_{Y(\tau_1)})} Y(\tau_2) \xrightarrow{\text{Exp}(\lambda_{Y(\tau_2)})} \dots$$

Figure 6.5: Jump-hold construction of a 2-state Markov jump process. State holding times are distributed exponentially.

Changing variables from X to Z, we move to the *noncentered auxiliary complete transition density*

$$\begin{aligned}
&\pi(\psi_{(\dot\tau,\ddot\tau)}, z_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta) \\
&\quad = d_\theta(v_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau})\, e^{(\dot\tau-\ddot\tau)\tilde\varphi_\theta^\downarrow(z_{(\dot\tau,\ddot\tau)}, y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}})} \underbrace{\prod_{t\in\gamma_\theta(\psi_{(\dot\tau,\ddot\tau)}, x_{(\dot\tau,\ddot\tau)}, y_{\dot\tau})} \left\{ \begin{pmatrix} \tilde\varphi_\theta^\uparrow - \tilde\varphi_\theta \\ \tilde\varphi_\theta^\uparrow - \tilde\varphi_\theta^\downarrow \end{pmatrix} (z_{(\dot\tau,\ddot\tau)}, y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}) \right\}_t}_{\bar q_\theta(\psi_{(\dot\tau,\ddot\tau)}, z_{(\dot\tau,\ddot\tau)}, v_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau})},
\end{aligned} \tag{6.26}$$

which is a density with respect to $\mathbb{P}_{(\dot\tau,\ddot\tau)} \times \mathbb{B}_{(\dot\tau,\ddot\tau)} \times \text{Leb}$, and where $\bar q_\theta(\psi_{(\dot\tau,\ddot\tau)}, z_{(\dot\tau,\ddot\tau)}, v_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau})$ acts as the path integral estimate on the noncentered path. See Figure 6.4 for the corresponding graphical model.

6.2 Simulation of Markov Jump Processes

Up to now, we have treated the trajectory of the Markov jump process as fixed, even though the process is latent in our framework. In order to infer Y as well, we require procedures for its forward, backward, and bridge simulation. Formally, these correspond to simulating according to $\pi(y_{(0,\omega]}|y_0, \lambda)$, $\pi(y_{[0,\omega)}|y_\omega, \lambda)$, and $\pi(y_{(0,\omega)}|y_{\{0,\omega\}}, \lambda)$ for a given generator λ. λ is a $\hat k \times \hat k$ matrix where $0 \le \lambda_{ij}$ gives the *jump rate* from state i to state j for $i \neq j$. The diagonal entries are set to $\lambda_{ii} = -\sum_{j\neq i} \lambda_{ij}$ such that row sums are 0. It is also convenient to define $\lambda_i = -\lambda_{ii}$. We further impose the following standing assumptions:

- Y is *irreducible*, i.e. any state is reachable from any other state. It is necessary for every column of λ to have at least one nonzero element. It is sufficient for all elements of λ to be nonzero.
- The process is *nonexplosive*, i.e. $\lambda_{ij} < \infty$ for all pairs $i \neq j$.

These assumptions are sufficient for the system to be *positive-recurrent*, i.e. the time to return to a state after visiting it for the first time is finite almost surely and in expectation. In the context of Bayesian inference for an unknown generator Λ, this is easily ensured by setting a prior for which $0 < \lambda_{ij} < \infty$ almost surely.

We will work with the jump-hold representation of Y, given by the almost surely finite set of transition times R and corresponding destination states:

$$\left\{(\dot R, Y_{\dot R}) : \dot R \in R\right\} \tag{6.27}$$

The construction is most easily understood in the $\hat{k} = 2$ case, illustrated in Figure 6.5. Here, from a starting value of i, Y switches to the other state after a holding time with distribution $\text{Exp}[\lambda_i]$. Since the exponential distribution is memoryless, Y is Markovian - the holding time conditional on $\{Y_{\dot{R}} = i\}$ always has distribution $\text{Exp}[\lambda_i]$, regardless of the previous trajectory. In the general instance, from a starting value of i, Y switches to state j after a holding time $E_j \sim \text{Exp}\left[\lambda_{ij}\right]$, *if it has not switched to another state yet.* Therefore, the new state corresponds to the smallest of the potential holding times. From an initial event $\{Y_{\dot{R}} = i\}$, the law of the next transition pair follows from

$$(\ddot{R}, Y_{\ddot{R}}) = \left(\dot{R} + \min_j E_j, \operatorname*{argmin}_j E_j\right). \tag{6.28}$$

We proceed with the statement of some relevant properties of Markov jump processes before presenting the necessary simulation algorithms.

6.2.1 Transition and Stationary Distribution

A useful property of Markov jump processes is that the probabilities $p_{ij}(\dot{\tau}, \ddot{\tau}) = \Pr[Y_{\ddot{\tau}} = j | Y_{\dot{\tau}} = i]$ can be described in terms of the *Kolmogorov backward and forward equations*

$$\frac{\partial p_{ij}(\dot{\tau}, \ddot{\tau})}{\partial \dot{\tau}} = -\sum_k \lambda_{ik} p_{kj}(\dot{\tau}, \ddot{\tau}), \tag{6.29}$$

$$\frac{\partial p_{ij}(\dot{\tau}, \ddot{\tau})}{\partial \ddot{\tau}} = \sum_k p_{ik}(\dot{\tau}, \ddot{\tau}) \lambda_{kj}, \tag{6.30}$$

for $\dot{\tau} < \ddot{\tau}$ and $i, j = 1, \ldots, \hat{k}$. These are systems of linear ordinary differential equations, and, assuming that λ can be eigendecomposed, either system is solved by

$$p_{ij}(\dot{\tau}, \ddot{\tau}) = \left\{e^{\omega\lambda}\right\}_{ij}, \tag{6.31}$$

where the exponential on the right hand side is to be understood as the *matrix exponential.* It is easily obtained from the eigendecomposition $\lambda = q(\text{diag}\,\varepsilon)q^{-1}$ as $p_{ij}(\dot{\tau}, \ddot{\tau}) = \left\{qe^{\omega\varepsilon}q^{-1}\right\}_{ij}$. Y being positive recurrent necessitates the existence of the limiting distribution

$$\bar{p}_j = \lim_{\ddot{\tau}\to\infty} p_{ij}(\dot{\tau}, \ddot{\tau}), \qquad (i, j = 1, \ldots, \hat{k}) \tag{6.32}$$

i.e. all the rows of the transition probability matrix $p(\dot{\tau}, \ddot{\tau})$ tend to the same limit, which we call $\bar{p}$. We obtain that limit by observing that the *balance equations*

$$\bar{p}_j \lambda_j = \sum_{i \neq j} \bar{p}_i \lambda_{ij} \tag{6.33}$$

must hold, subject to $\sum_i \bar{p}_i = 1$. Intuitively, the balance equations require the probability flow out of a state to be equal to the flow into the state. The solution can be represented and solved for in terms of the linear system

$$\lambda^{\mathsf{T}} \bar{p} = 0, \quad 1^{\mathsf{T}} \bar{p} = 1. \tag{6.34}$$

6.2.2 Forward and Backward Simulation

The forward simulation algorithm follows from the jump-hold construction of Y for a finite number of states $\hat{k}$. In this instance, We obtain a full trajectory from Y by repeating the procedure until $\ddot{R} > \omega$.

Algorithm 17 Forward simulation of $y_{(0,\omega]} \sim \pi(y_{(0,\omega]}|y_0, \lambda)$.

$(t, y_t) \leftarrow (0, y_0)$, $y_{(0,\omega]} \leftarrow \{(t, y_t)\}$
while $t \leq \omega$ **do**
  $e_j \sim \text{Exp}\left[\lambda_{y_t}\right]$ for $j = 1, \ldots, \hat{k}$
  $(t, y_t) \leftarrow \left(t + \min_j e_j, \text{argmin}_j\, e_j\right)$
  $y_{(0,\omega]} \leftarrow y_{(0,\omega]} \cup \{(t, y_t)\}$
$y_{(0,\omega]} \leftarrow \left\{(t, y_t) \in y_{(0,\omega]} : 0 < t \leq \omega\right\}$

Backward simulation according to $\pi(y_{[0,\omega)}|y_\omega, \lambda)$ is most easily carried out by exploiting the representation

$$\pi(y_{[0,\omega)}|y_\omega, \lambda) = \sum_{y_0=1}^{K} \pi(y_{(0,\omega)}|y_{\{0,\omega\}}, \lambda)\pi(y_0|y_\omega, \lambda), \tag{6.35}$$

which suggests simulating $Y_0 \sim \pi(y_0|y_\omega, \lambda)$ and $Y_{(0,\omega)} \sim \pi(y_{(0,\omega)}|Y_0, y_\omega, \lambda)$ in sequence. $\pi(y_0|y_\omega, \lambda)$ is obtained by way of Bayes' theorem:

$$\pi(y_0|y_\omega, \lambda) = p_{y_\omega, y_0}(0, \omega)\frac{\bar{p}_{y_0}}{\bar{p}_{y_\omega}} \tag{6.36}$$

6.2.3 Rejection Bridge Simulation

The brute-force approach to bridge sampling consists of rejection sampling $y^\dagger_{(0,\omega]}$ from the forward proposal $\pi(y_{(0,\omega]}|y_0, \lambda)$ and accepting $y^\dagger_{(0,\omega)}$ if $y^\dagger_\omega = y_\omega$. This occurs with probability $p_{y_0,y_\omega}(0, \omega)$, which goes to 0 in concert with ω unless $y_0 = y_\omega$. [95] proposes a tweak to avoid that shortfall which consists of sampling from the distribution of the first state change, given that at least one change occurs. That distribution has an explicit quantile function, allowing sampling according to the inverse transform method:

$$-\log\left[1 - u(1 - e^{\omega\lambda_{y(0)}})\right]/\lambda_{y(0)} \qquad (u \in [0, 1]) \tag{6.37}$$

The new state then takes value j with probability $\lambda_{y(0),j}/\lambda_{y(0)}$. Hence, even the tweaked method will perform badly if $\lambda_{y(0),y(\omega)}/\lambda_{y(0)}$ is small. On the other hand, each iteration only has cost $\mathcal{O}(\hat{k})$, so a lot of trajectories can be simulated cheaply.

6.2.4 Direct Bridge Simulation

Motivated by the inefficiency of rejection sampling in certain contexts, [63] proposed a more robust method to directly sample from the joint distribution of the next jump time and state. Such a sampling procedure could then be applied iteratively until all the jumps have been simulated. We distinguish the two cases of equality and inequality in current and final states $(y_{\dot\tau}, y_\omega)$.

We begin with the case $y_{\dot\tau} \neq y_\omega$, where at least one jump must occur by definition. The joint density of the next jump time and state $(\ddot T, Y_{\ddot T})$ is given by

$$\begin{aligned}
\pi(\ddot\tau, y_{\ddot\tau} | y_{\{\dot\tau,\omega\}}) &= \frac{\pi(y_\omega|y_{\ddot\tau})}{\pi(y_\omega|y_{\dot\tau})} \pi(y_{\ddot\tau}|\ddot\tau, y_{\dot\tau}) \pi(\ddot\tau|y_{\dot\tau}) \\
&= \frac{p_{y_{\ddot\tau},y_\omega}(\ddot\tau,\omega)}{p_{y_{\dot\tau},y_\omega}(\dot\tau,\omega)} \frac{\lambda_{y_{\dot\tau},y_{\ddot\tau}}}{\lambda_{y_{\dot\tau}}} \operatorname{Exp}\left[\ddot\tau - \dot\tau; \lambda_{y_{\dot\tau}}\right] \\
&\propto \lambda_{y_{\dot\tau},y_{\ddot\tau}} \sum_{k=1}^{\hat k} \{q\}_{y_{\ddot\tau},k} \left\{q^{-1}\right\}_{k,\omega} \exp\left[\omega\varepsilon_k - \ddot\tau(\lambda_{\dot\tau} + \varepsilon_k)\right].
\end{aligned} \tag{6.38}$$

We then extract the $Y_{\ddot T}$-marginal by integrating out the holding time. The resulting expression recurs throughout this section, we therefore define

$$\begin{aligned}
f_{y_{\ddot\tau}}(t) &\propto \int_{\dot\tau}^{t} \pi(\ddot\tau, y_{\ddot\tau}|y_{\{\dot\tau,\omega\}})\,\mathrm{d}\ddot\tau \\
&= \lambda_{y_{\dot\tau},y_{\ddot\tau}} \sum_{k=1}^{\hat k} \{q\}_{y_{\ddot\tau},k} \left\{q^{-1}\right\}_{k,\omega} e^{\omega\varepsilon_k} \int_{\dot\tau}^{t} \exp\left[-\ddot\tau(\lambda_{\dot\tau}+\varepsilon_k)\right]\mathrm{d}\ddot\tau \\
&= \lambda_{y_{\dot\tau},y_{\ddot\tau}} \sum_{k=1}^{\hat k} \{q\}_{y_{\ddot\tau},k} \left\{q^{-1}\right\}_{k,\omega} e^{\omega\varepsilon_k} \begin{cases} (t-\dot\tau) & (\lambda_{\dot\tau}+\varepsilon_k = 0) \\ \dfrac{e^{-\dot\tau(\lambda_{\dot\tau}+\varepsilon_k)} - e^{-t(\lambda_{\dot\tau}+\varepsilon_k)}}{\lambda_{\dot\tau}+\varepsilon_k} & \text{otherwise} \end{cases}
\end{aligned} \tag{6.39}$$

and note that $\pi(y_{\ddot\tau}|y_{\{\dot\tau,\omega\}}) \propto f_{y_{\ddot\tau}}(\omega)$. Since the marginal is a categorical distribution, it is easily renormalized. Given $y_{\ddot\tau}$, we obtain the conditional CDF

$$\begin{aligned}
\Pr\left[\ddot T \leq \ddot\tau | y_{\{\dot\tau,\ddot\tau,\omega\}}\right] &= \frac{\int_{\dot\tau}^{\ddot\tau} \pi(\ddot\tau, y_{\ddot\tau}|y_{\{\dot\tau,\omega\}})\,\mathrm{d}\ddot\tau}{\int_{\dot\tau}^{\omega} \pi(\ddot\tau, y_{\ddot\tau}|y_{\{\dot\tau,\omega\}})\,\mathrm{d}\ddot\tau} \\
&= \frac{f_{y_{\ddot\tau}}(\ddot\tau)}{f_{y_{\ddot\tau}}(\omega)}.
\end{aligned} \tag{6.40}$$

This expression is not analytically invertible, but we can still apply the inverse transform method by noting that there is a random variable U such that

$$f_{y_{\ddot\tau}}(\ddot T) - U f_{y_{\ddot\tau}}(\omega) = 0, \quad U \sim \operatorname{Uniform}[0,1]. \tag{6.41}$$

Therefore, the solution of $f_{y_{\dot\tau}}(\ddot\tau) - u f_{y_{\dot\tau}}(\omega) = 0$ in $\ddot\tau$ for a given uniform variate u is a draw from the conditional CDF.

We next consider the closely related case $y_{\dot\tau} = y_\omega = i$, where either no jumps occur, or 2 or more jumps occur. We first evaluate the no-jump event, which has conditional probability

$$\begin{aligned}\Pr\left[\text{no jumps}|Y_{\dot\tau} = Y_\omega = i\right] &= \frac{\Pr\left[\text{no jumps}, Y_\omega = Y_{\dot\tau}|Y_{\dot\tau} = i\right]}{\Pr\left[Y_\omega = Y_{\dot\tau}|Y_{\dot\tau} = i\right]} \\ &= \frac{\Pr\left[\ddot T > \omega|Y_{\dot\tau} = i\right]}{\Pr\left[Y_\omega = i|Y_{\dot\tau} = i\right]} \\ &= \frac{e^{-(\omega-\dot\tau)\lambda_i}}{p_{ii}(\dot\tau,\omega)},\end{aligned} \tag{6.42}$$

where the second step is due to $\{\text{no jumps}\} \subset \{Y_\omega = Y_{\dot\tau}\}$. If there are no further jumps, the algorithm terminates. If there are further jumps, the joint density of $(\ddot T, Y_{\ddot T})$ is given by $\pi(\ddot\tau, y_{\ddot\tau}|y_{\{\dot\tau,\omega\}})$, and all remaining steps can proceed as in the $y_{\dot\tau} \neq y_\omega$ case.

The main appeal of the direct method is its robustness - it only requires as many iterations as the Y-bridge carries out jumps. Its main downside is the need for an eigendecomposition of λ at $\mathcal{O}(\hat k^3)$ cost, a cost per iteration of order $\mathcal{O}(\hat k^2)$ to compute f_i for all states, and the need for an iterative numerical root finder, which itself might require many function evaluations.

6.2.5 Uniformized Bridge Simulation

[64] contrast rejection sampling and direct sampling with the *uniformization* approach. It consists of decomposing Y into a convolution of an unmarked Poisson process Φ of intensity $\lambda^\uparrow = \max_i \lambda_i$ with a discrete time process $\breve Y$ that transitions at epochs given by Φ, according to the probabilities

$$\breve\lambda = \mathrm{id} + \lambda/\lambda^\uparrow. \tag{6.43}$$

While Φ generates events at least as often at Y transitions in any state, the self-transitions in $\breve Y$ cause it to marginally generate as many transitions to other states as Y. Indeed, noting that $|\Phi| \sim \mathrm{Pois}\left[\omega\lambda^\uparrow\right]$, we find that the marginal transition probabilities

are given by

$$\begin{aligned}
\Pr\left[\breve{Y}_\omega = j|\breve{Y}_0 = i\right] &= \sum_{n=0}^{\infty} \text{Pois}\left[n; \omega\lambda^{\uparrow}\right] \Pr\left[\breve{Y}_\omega = j|\breve{Y}_0 = i, |\Phi| = n\right] \\
&= \sum_{n=0}^{\infty} e^{-\omega\lambda^{\uparrow}} \frac{(\omega\lambda^{\uparrow})^n}{n!} \left\{\breve{\lambda}^n\right\}_{ij} \\
&= e^{-\omega\lambda^{\uparrow}} \sum_{n=0}^{\infty} \frac{1}{n!} \left\{(\omega\lambda^{\uparrow}\breve{\lambda})^n\right\}_{ij} \\
&= e^{-\omega\lambda^{\uparrow}} \left\{e^{\omega\lambda^{\uparrow}\breve{\lambda}}\right\}_{ij} \\
&= \left\{e^{\omega\lambda}\right\}_{ij} \\
&= p_{ij}(0, \omega),
\end{aligned} \tag{6.44}$$

where we have applied the elementary definition of the matrix exponential. Since $\breve{Y}$ marginally follows the same transition probabilities as Y, the two processes are equivalent. We may therefore use the discretized construction in terms of $\breve{Y}$ and Φ to obtain Y-bridges. The procedure begins by simulating from the conditional distribution of $|\Phi|$, which follows from Bayes' law:

$$\begin{aligned}
\Pr\left[|\Phi| = n|Y_0 = i, Y_\omega = j\right] &= \Pr\left[|\Phi| = n\right] \frac{\Pr\left[Y_\omega = j|Y_0 = i, |\Phi| = n\right]}{\Pr\left[Y_\omega = j|Y_0 = i\right]} \\
&= \text{Pois}\left[n; \omega\lambda^{\uparrow}\right] \frac{\left\{\breve{\lambda}^n\right\}_{ij}}{p_{ij}(0, \omega)}.
\end{aligned} \tag{6.45}$$

We can sample from the corresponding CDF

$$\Pr\left[|\Phi| \leq n|Y_0 = i, Y_\omega = j\right] = \sum_{n'=1}^{n} \text{Pois}\left[n'; \omega\lambda^{\uparrow}\right] \frac{\left\{\breve{\lambda}^{n'}\right\}_{ij}}{p_{ij}(0, \omega)} \tag{6.46}$$

using the inverse transform method, by finding

$$\max\left\{n : U \leq \Pr\left[|\Phi| \leq n|Y_0 = i, Y_\omega = j\right]\right\}, \quad U \sim \text{Unif}\left[0, 1\right]. \tag{6.47}$$

The sequence of matrix powers $\breve{\lambda}^n$ is also required by the next step of the procedure, and should be stored. Conditional on $\{|\Phi| = n\}$, Φ consists of n uniformly distributed points on $[0, \omega]$. Finally, conditional on $\{\Phi = \phi\}$, transition probabilities of Y between times $(\dot{\tau} \sim \ddot{\tau}) \in \phi$ again follow from Bayes' law:

$$\begin{aligned}
\Pr\left[Y_{\ddot{\tau}} = j|Y_{\dot{\tau}} = i, Y_\omega = k, \Phi = \phi\right] &= \Pr\left[Y_{\ddot{\tau}} = j|Y_{\dot{\tau}} = i, \Phi = \phi\right] \\
&\quad \times \frac{\Pr\left[Y_\omega = k|Y_{\ddot{\tau}} = j, \Phi = \phi\right]}{\Pr\left[Y_\omega = k|Y_{\dot{\tau}} = i, \Phi = \phi\right]} \\
&\propto \left\{\breve{\lambda}\right\}_{ij} \left\{\breve{\lambda}^{|\phi \cap (\ddot{\tau}, \omega]|}\right\}_{jk}
\end{aligned} \tag{6.48}$$

This method has similar complexity characteristics as direct sampling, requiring a $\mathcal{O}(\hat{k}^3)$ upfront investment and an expenditure of order $\mathcal{O}(\hat{k}^2)$ at each iteration. In practice, it iterates more quickly than the direct sampler because it doesn't require additional iterative procedures. On the other hand, the number of iterations in direct sampling lower bounds the number of iterations of uniformized sampling. Whether it is preferable to direct sampling depends on $\operatorname{tr}\left[\lambda^{\uparrow}\,\mathrm{id}\,+\lambda\right]$ - where that trace is large, many virtual jumps occur, which requires iterations that direct sampling avoids.

6.3 Marginal Algorithm

In this section, we develop an MCMC algorithm that targets the marginal posterior

$$\pi(v_r, z, y, \theta, \lambda | v_s) \propto \pi(\theta)\pi(\lambda) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}} | v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}} | y_{\dot{\tau}}, \lambda). \tag{6.49}$$

We construct a Gibbs sampler with full conditionals

$$(\Theta, \Lambda): \quad \pi(\theta, \lambda | v_\tau, z, y) \propto \pi(\theta)\pi(\lambda) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}} | v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}} | y_{\dot{\tau}}, \lambda), \tag{6.50}$$

$$(V_R, Z, Y): \quad \pi(v_r, z, y | v_s, \theta, \lambda) \propto \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}} | v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}} | y_{\dot{\tau}}, \lambda), \tag{6.51}$$

where, keeping in mind that the set of jumps R and event times T follow deterministically from Y, and that R is almost surely finite, the dominating measure of the latter is the product measure

$$\mathbb{L}(\mathrm{d}y) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \mathbb{B}_{(\dot{\tau},\ddot{\tau})}(\mathrm{d}z_{(\dot{\tau},\ddot{\tau})}) \prod_{\dot{r}\in r} \mathrm{Leb}(\mathrm{d}v_{\dot{r}}). \tag{6.52}$$

Therefore, the parameterization of the diffusion bridges evolves according to the number of jumps in Y. This blocking is likely to be negatively affected by inevitable dependence between Θ and Y, but it offers various opportunities for exploiting conditional independence and tuning proposals. The immediate benefit is that the (Θ, Λ)-update decomposes into the independent updates

$$\Theta: \quad \pi(\theta | v_\tau, z, y) \propto \pi(\theta) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}} | v_{\dot{\tau}}, y_{\dot{\tau}}, \theta), \tag{6.53}$$

$$\Lambda: \quad \pi(\lambda | y) \propto \pi(\lambda) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(y_{\ddot{\tau}} | y_{\dot{\tau}}, \lambda). \tag{6.54}$$

In particular, the Λ-update is conjugate for the class of priors discussed in Section 6.3.2, and may thus be sampled exactly. In addition, the Θ-update simplifies further for models in the class

$$\mathrm{d}V_t = \mu_{\theta_{Y_t}}(V_t)\,\mathrm{d}t + \sigma(V_t)\rho_{\theta_{Y_t}}\,\mathrm{d}W_t, \qquad \theta = (\theta_1, \ldots, \theta_{\hat{k}}), \tag{6.55}$$

with product prior $\pi(\theta) = \prod_{k=1}^{\hat{k}} \pi(\theta_k)$. Now, each Θ_k may be updated independently according to

$$\pi(\theta_k|v_\tau, z, y) \propto \pi(\theta_k) \prod_{(\dot\tau\sim\ddot\tau)\in\tau: y_{\dot\tau}=k} \pi(z_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta_k). \tag{6.56}$$

We exploit that structure in designing the algorithm for the experiments in Sections 6.8 and 6.7. Conversely, the main departure, and indeed complication compared to the Itō diffusion algorithm is that the (V_R, Z, Y)-update does not immediately factorize. This is due to the fact that in this setup, none of the elements of Y are known and conditioned upon. We would have to move to the setting of [20], where Y is observed along with V at times s to obtain the independent updates

$$\begin{aligned}&\pi(v_{r\cap(\dot s,\ddot s)}, z_{(\dot s,\ddot s)}, y_{(\dot s,\ddot s)}|v_{\{\dot s,\ddot s\}}, y_{\{\dot s,\ddot s\}}, \theta, \lambda),\\ &\qquad\propto \prod_{(\dot\tau\sim\ddot\tau)\in(\tau\cap(\dot s,\ddot s))} \pi(z_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta)\pi(y_{\ddot\tau}|y_{\dot\tau}, \lambda).\end{aligned} \tag{6.57}$$

Since this setting excludes many applications where Y is entirely latent, we will remain in the unobserved setting, which is easily adapted to the observed one. Having to update (V_R, Z, Y) jointly is extremely difficult for long trajectories, and we adopt the strategy of randomly conditioning on finite subsets of Y and V in order to obtain more manageable updates.

Remark 4 (Diffusion bridge representation)**.** *Our representation of the infinite-dimensional paths z remains unchanged from Section 5.2.1.*

6.3.1 Diffusion Parameter Update

We carry out the diffusion parameter by way of a Barker-within-Gibbs step with generic proposal density $\kappa(\theta^\dagger|\theta)$, following the strategy developed in Section 5.2.2. The proposal has acceptance odds

$$\begin{aligned}\frac{\alpha_\Theta}{1-\alpha_\Theta} &= \frac{\pi(\theta^\dagger|v_\tau, z, y)}{\pi(\theta|v_\tau, z, y)}\frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\\ &= \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\frac{\pi(\theta^\dagger)}{\pi(\theta)}\prod_{(\dot\tau\sim\ddot\tau)\in\tau}\frac{\pi(z_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta^\dagger)}{\pi(z_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta)}\\ &= \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\frac{\pi(\theta^\dagger)}{\pi(\theta)}\prod_{(\dot\tau\sim\ddot\tau)\in\tau}\frac{d_{\theta^\dagger}(v_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau})}{d_\theta(v_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau})}\frac{e^{-\int_{\dot s}^{\ddot s}\tilde\varphi_{\theta^\dagger}(z_t, y_{\dot\tau}, v_{\{\dot s,\ddot s\}})\,\mathrm{d}t}}{e^{-\int_{\dot s}^{\ddot s}\tilde\varphi_{\theta}(z_t, y_{\dot\tau}, v_{\{\dot s,\ddot s\}})\,\mathrm{d}t}}.\end{aligned} \tag{6.58}$$

We define the appropriate vanishing integrand

$$\upsilon_t = \tilde\varphi_{\theta^\dagger}(z_t, y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}) - \tilde\varphi_\theta(z_t, y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}), \qquad (t\in(\dot\tau,\ddot\tau)) \tag{6.59}$$

with positive and negative parts $\upsilon_t^{(+)}$ and $\upsilon_t^{(-)}$, and obtain the valid 2-coin factorization

$$\begin{aligned}\frac{\alpha_\Theta}{1-\alpha_\Theta} &= \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)}\frac{\pi(\theta^\dagger)}{\pi(\theta)}\prod_{(\dot\tau\sim\ddot\tau)\in s}\frac{d_{\theta^\dagger}(v_{\{\dot\tau,\ddot\tau\}},y_{\dot\tau})}{d_\theta(v_{\{\dot\tau,\ddot\tau\}},y_{\dot\tau})}\exp\left[-\int_{\dot\tau}^{\ddot\tau}\upsilon_t\,\mathrm{d}t\right] \\ &= \underbrace{\frac{\overbrace{\kappa(\theta|\theta^\dagger)\pi(\theta^\dagger)\prod_{(\dot\tau\sim\ddot\tau)\in s}d_{\theta^\dagger}(v_{\{\dot\tau,\ddot\tau\}},y_{\dot\tau})}^{c_1}}{\kappa(\theta^\dagger|\theta)\pi(\theta)\prod_{(\dot\tau\sim\ddot\tau)\in s}d_\theta(v_{\{\dot\tau,\ddot\tau\}},y_{\dot\tau})}}_{c_2}\underbrace{\frac{\overbrace{\prod_{(\dot\tau\sim\ddot\tau)\in s}e^{-\int_{\dot\tau}^{\ddot\tau}\upsilon_t^{(+)}\,\mathrm{d}t}}^{p_1}}{\prod_{(\dot\tau\sim\ddot\tau)\in s}e^{-\int_{\dot\tau}^{\ddot\tau}\upsilon_t^{(-)}\,\mathrm{d}t}}}_{p_2}.\end{aligned} \tag{6.60}$$

Naive bounds on the integrands in the range $t\in(\dot\tau,\ddot\tau)$ are given by are given by

$$\upsilon_t^{(+)} \le \tilde\varphi^{\uparrow}_{\theta^\dagger}(z_{(\dot\tau,\ddot\tau)},y_{\dot\tau},v_{\dot\tau,\ddot\tau}) - \tilde\varphi^{\downarrow}_{\theta}(z_{(\dot\tau,\ddot\tau)},y_{\dot\tau},v_{\dot\tau,\ddot\tau}), \tag{6.61}$$

$$\upsilon_t^{(-)} \le \tilde\varphi^{\uparrow}_{\theta}(z_{(\dot\tau,\ddot\tau)},y_{\dot\tau},v_{\dot\tau,\ddot\tau}) - \tilde\varphi^{\downarrow}_{\theta^\dagger}(z_{(\dot\tau,\ddot\tau)},y_{\dot\tau},v_{\dot\tau,\ddot\tau}), \tag{6.62}$$

and vanishing bounds can be obtained following identical arguments as for Itō diffusions, proceeding from the inequality

$$|\upsilon_t| \le \sup_{a\in[0,1]}\left|\nabla_\theta\tilde\varphi_\theta(z_t,y_{\dot\tau},v_{\{\dot\tau,\ddot\tau\}})\Big|_{\theta=(1-a)\theta^\dagger+a\theta^\ddagger}\right||\theta^\dagger-\theta^\ddagger|. \tag{6.63}$$

6.3.2 Regime Parameter Update

Since the full conditional density for Λ does not involve the augmented transition densities, it is the easiest update to carry out:

$$\pi(\lambda|y)\propto\pi(y|\lambda)\pi(\lambda) \tag{6.64}$$

For jump times r, the density $\pi(y|\lambda)$ with respect to $\mathbb{L}$ is given by

$$\pi(y|\lambda)=\exp\left[\int_0^\omega(1-\lambda_{y_t})\,\mathrm{d}t\right]\prod_{(\dot r\sim\ddot r)\in r\cup\{0\}}\lambda_{y_{\dot r}y_{\ddot r}}. \tag{6.65}$$

Defining the cumulative holding times $\chi_i=\int_0^\omega 1_{y_t=\{i\}}\,\mathrm{d}t$ and the jump counts n_{ij} from state i to j, we find that they are a sufficient statistic:

$$\pi(y|\lambda)\propto\prod_i\left(e^{-\lambda_i\chi_i}\prod_{j\neq i}\lambda_{ij}^{n_{ij}}\right). \tag{6.66}$$

We set the following conjugate product prior:

$$\pi(\lambda)=\prod_{i\neq j}\mathrm{Gamma}\left[\lambda_{ij};\alpha,\beta\right], \tag{6.67}$$

that is, the free elements of Λ are independent a priori. The posterior distribution then becomes

$$\pi(\lambda|y) = \prod_{i \neq j} \text{Gamma}\left[\lambda_{ij}; n_{ij} + \alpha, \chi_i + \beta\right]. \tag{6.68}$$

The reader interested in applications should note that this prior is necessarily informative - Y is usually ill-identified by the data v_s alone. This is especially the case if λ allows for states that are ephemeral relative to the observation frequency on the diffusion path and therefore vacuous. Accordingly, the prior expectation of Λ_i, given by

$$\mathrm{E}\left[\Lambda_i\right] = \sum_{i \neq j} \mathrm{E}\left[\Lambda_{ij}\right] = \sum_{i \neq j} \frac{\alpha_{ij}}{\beta_{ij}}, \tag{6.69}$$

should be chosen such that it is smaller than the mean observation rate.

6.3.3 Independence Hidden Data Update

We begin by considering an independence proposal for updating $\pi(v_r, z, y|v_s, \theta, \lambda)$. This is computationally feasible when v_s is fairly uninformative, or the time horizon short, and results in notation that is easier to parse. We construct a Barker-within Gibbs update with a hierarchical proposal

$$\kappa(v_{r^\dagger}^\dagger, z^\dagger, y^\dagger|v_s) \propto \kappa(y^\dagger)\kappa(v_r^\dagger|v_s, y^\dagger) \prod_{(\dot\tau \sim \ddot\tau) \in \tau} \kappa(z_{(\dot\tau,\ddot\tau)}^\dagger), \tag{6.70}$$

where $Z_{(\dot\tau,\ddot\tau)}^\dagger \sim \mathbb{B}_{(\dot\tau,\ddot\tau)}$ and $\kappa(z_{(\dot\tau,\ddot\tau)}^\dagger) = 1$. $Y^\dagger$ is proposed independently from its prior distribution, i.e. $\kappa(y^\dagger) = \pi(y^\dagger|\lambda)$. The prior is easily simulated from according to the algorithm in Section 6.2.1. Given $Y^\dagger$, the proposal for $V_{r^\dagger}^\dagger$ is most readily understood in terms of $X_{r^\dagger}^\dagger = \eta_\theta(V_{r^\dagger}^\dagger)$. We propose $X_{r^\dagger}^\dagger$ according to the dominating SDE $\mathrm{d}X_t = \rho_\theta(Y_t^\dagger)\,\mathrm{d}W_t$ with induced measure $\mathbb{M}|(x_s, y^\dagger, \theta)$. By the Markov property,

$$\begin{aligned}
\mathbb{M}|(x_s, y, \theta)(\mathrm{d}x_r) &= \prod_{(\dot s \sim \ddot s) \in s} \mathbb{M}|(x_{\{\dot s,\ddot s\}}, y_{[\dot s,\ddot s)}, \theta)(\mathrm{d}x_{r \cap (\dot s,\ddot s)}) \\
&= \prod_{(\dot s \sim \ddot s) \in s} \frac{\prod_{(\dot\tau \sim \ddot\tau) \in r \cap (\dot s,\ddot s)} \mathbb{M}|(x_{\dot\tau}, y_{\dot\tau}, \theta)(\mathrm{d}x_{\ddot\tau})}{\mathbb{M}|(x_{\dot s}, y_{[\dot s,\ddot s)}, \theta)(\mathrm{d}x_{\ddot s})},
\end{aligned} \tag{6.71}$$

and each subset $X_{r^\dagger \cap (\dot s,\ddot s)}^\dagger$ may be simulated independently. We do so by observing that by the time change representation of the stochastic integral, the transformation

$$(t, x_t) \mapsto \left(\int_{\dot s}^t \rho_\theta^2(y_u^\dagger)\,\mathrm{d}u, x_t\right) \tag{6.72}$$

maps X to a unit volatility Brownian bridge connecting $(0, x_{\dot s}) \to (\int_{\dot s}^{\ddot s} \rho_\theta^2(y_u^\dagger)\,\mathrm{d}u, x_{\ddot s})$. Conversely, a sample from that bridge at time $\int_{\dot s}^t \rho_\theta^2(y_u^\dagger)\,\mathrm{d}u$ follows the proposal law

of $X_t^\dagger$. We then obtain $V_{r^\dagger}^\dagger = \eta_\theta^{-1}(X_{r^\dagger}^\dagger)$. The measure $\mathbb{M}|(x_{\{\dot{s},\ddot{s}\}}, y_{[\dot{s},\ddot{s})}, \theta)(\mathrm{d}x_{r\cap(\dot{s},\ddot{s})})$ is Gaussian and has density

$$\kappa(x^\dagger_{r^\dagger\cap(\dot{s},\ddot{s})}|x_{\{\dot{s},\ddot{s}\}}, y^\dagger) = \frac{\prod_{(\dot{\tau}\sim\ddot{\tau})\in r^\dagger\cap(\dot{s},\ddot{s})} \mathrm{N}\left[x^\dagger_{\ddot{\tau}}; x^\dagger_{\dot{\tau}}, (\ddot{\tau}-\dot{\tau})\rho^2_\theta(y^\dagger_{\dot{\tau}})\right]}{\mathrm{N}\left[x_{\ddot{s}}; x_{\dot{s}}, \sum_{(\dot{\tau}\sim\ddot{\tau})\in r^\dagger\cap(\dot{s},\ddot{s})}(\ddot{\tau}-\dot{\tau})\rho^2_\theta(y^\dagger_{\dot{\tau}})\right]}, \tag{6.73}$$

from which we recover the proposal density on $V^\dagger_{r^\dagger\cap(\dot{s},\ddot{s})}$ by the change of variable formula:

$$\kappa(v^\dagger_{r^\dagger\cap(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}}, y^\dagger) = \frac{\prod_{(\dot{\tau}\sim\ddot{\tau})\in r^\dagger\cap(\dot{s},\ddot{s})} |\eta_\theta^{-1}(v_{\ddot{\tau}})|\,\mathrm{N}\left[\eta_\theta(v^\dagger_{\ddot{\tau}}); \eta_\theta(v^\dagger_{\dot{\tau}}), (\ddot{\tau}-\dot{\tau})\rho^2_\theta(y^\dagger_{\dot{\tau}})\right]}{\mathrm{N}\left[\eta_\theta(v_{\ddot{s}}); \eta_\theta(v_{\dot{s}}), \sum_{(\dot{\tau}\sim\ddot{\tau})\in r^\dagger\cap(\dot{s},\ddot{s})}(\ddot{\tau}-\dot{\tau})\rho^2_\theta(y^\dagger_{\dot{\tau}})\right]}. \tag{6.74}$$

Thus, the proposal density on $V_{r^\dagger}$ is given by $\kappa(v^\dagger_{r^\dagger}|v_s, y^\dagger) = \prod_{(\dot{s}\sim\ddot{s})\in s} \kappa(v^\dagger_{r^\dagger\cap(\dot{s},\ddot{s})}|v_{\{\dot{s},\ddot{s}\}}, y^\dagger)$. We give the step-by-step routine below.

Algorithm 18 Algorithm for generating proposal from $\kappa(v_r|v_s, y)$.

$x_s \leftarrow \eta_\theta(v_s)$
for $(\dot{s}\sim\ddot{s})\in s$ **do**
  $u \leftarrow \left\{\int_{\dot{s}}^{\dot{r}} \rho^2_\theta(y_t)\,\mathrm{d}t : \dot{r}\in r\right\}$
  $w_u \sim \mathbb{W}|(W_0 = x_{\dot{s}}, W(\int_{\dot{s}}^{\ddot{s}} \rho^2_\theta(y_t)\,\mathrm{d}t) = x_{\ddot{s}})$
  $x_{r\cap(\dot{s},\ddot{s})} \leftarrow w_u$
  $v_{r\cap(\dot{s},\ddot{s})} \leftarrow \eta_\theta^{-1}(x_{r\cap(\dot{s},\ddot{s})})$

With the proposal fully specified, we state the acceptance odds as

$$\begin{aligned}
\frac{\alpha_{(V_R,Z,Y)}}{1-\alpha_{(V_R,Z,Y)}} &= \frac{\kappa(v_r, z, y|v_s)}{\kappa(v^\dagger_{r^\dagger}, z^\dagger, y^\dagger|v_s)} \frac{\pi(v^\dagger_{r^\dagger}, z^\dagger, y^\dagger|v_s, \theta, \lambda)}{\pi(v_r, z, y|v_s, \theta, \lambda)} \\
&= \frac{\kappa(v_r|v_s, y)}{\kappa(v^\dagger_{r^\dagger}|v_s, y^\dagger)} \frac{\prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau^\dagger} \pi(z^\dagger_{(\dot{\tau},\ddot{\tau})}, v^\dagger_{\ddot{\tau}}|v^\dagger_{\dot{\tau}}, y^\dagger_{\dot{\tau}}, \theta)}{\prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)} \\
&= \frac{\kappa(v_r|v_s, y)}{\kappa(v^\dagger_{r^\dagger}|v_s, y^\dagger)} \frac{\prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau^\dagger} d_\theta(v^\dagger_{\{\dot{\tau},\ddot{\tau}\}}, y^\dagger_{\dot{\tau}}) e^{-\int_{\dot{\tau}}^{\ddot{\tau}} \tilde{\varphi}_\theta(z^\dagger_t, y^\dagger_{\dot{\tau}}, v^\dagger_{\{\dot{\tau},\ddot{\tau}\}})\,\mathrm{d}t}}{\prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} d_\theta(v_{\{\dot{\tau},\ddot{\tau}\}}, y_{\dot{\tau}}) e^{-\int_{\dot{\tau}}^{\ddot{\tau}} \tilde{\varphi}_\theta(z_t, y_{\dot{\tau}}, v_{\{\dot{\tau},\ddot{\tau}\}})\,\mathrm{d}t}}.
\end{aligned} \tag{6.75}$$

For any two intersecting event pairs $(\dot{\tau}\sim\ddot{\tau})\in\tau$ and $(\dot{\tau}^\dagger\sim\ddot{\tau}^\dagger)\in\tau^\dagger$, we define the differenced integrand

$$\xi_t = \tilde{\varphi}_\theta(z^\dagger_t, y^\dagger_{\dot{\tau}^\dagger}, v^\dagger_{\{\dot{\tau}^\dagger,\ddot{\tau}^\dagger\}}) - \tilde{\varphi}_\theta(z_t, y_{\dot{\tau}}, v^\dagger_{\{\dot{\tau},\ddot{\tau}\}}), \qquad (t\in(\dot{\tau},\ddot{\tau})\cap(\dot{\tau}^\dagger,\ddot{\tau}^\dagger)) \tag{6.76}$$

and denote its positive and negative parts $\xi_t^{(+)}$ and $\xi_t^{(-)}$. We obtain the 2-coin algorithm

$$\frac{\alpha_{(V_R,Z,Y)}}{1-\alpha_{(V_R,Z,Y)}} = \underbrace{\overbrace{\kappa(v_r|v_s,y)\prod_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger} d_\theta(v^\dagger_{\{\dot\tau,\ddot\tau\}},y^\dagger_{\dot\tau})}^{c_1}}_{c_2} \underbrace{\overbrace{\prod_{(\dot s\sim\ddot s)\in s} e^{-\int_{\dot s}^{\ddot s}\xi_t^{(+)}\,\mathrm{d}t}}^{p_1}}_{p_2}, \tag{6.77}$$

with integrand bounds in the range $(\dot s, \ddot s)$ given by

$$\xi_t^{(+)} \le \max_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger\cap[\dot s,\ddot s]} \widetilde\varphi^\uparrow_\theta(z^\dagger_{(\dot\tau,\ddot\tau)},y^\dagger_{\dot\tau},v^\dagger_{\{\dot\tau,\ddot\tau\}}) - \min_{(\dot\tau\sim\ddot\tau)\in\tau\cap[\dot s,\ddot s]} \widetilde\varphi^\downarrow_\theta(z_{(\dot\tau,\ddot\tau)},y_{\dot\tau},v_{\{\dot\tau,\ddot\tau\}},y_{\dot\tau}), \tag{6.78}$$

$$\xi_t^{(-)} \le \max_{(\dot\tau\sim\ddot\tau)\in\tau\cap[\dot s,\ddot s]} \widetilde\varphi^\uparrow_\theta(z_{(\dot\tau,\ddot\tau)},y_{\dot\tau},v_{\{\dot\tau,\ddot\tau\}}) - \min_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger\cap[\dot s,\ddot s]} \widetilde\varphi^\downarrow_\theta(z^\dagger_{(\dot\tau,\ddot\tau)},y^\dagger_{\dot\tau},v^\dagger_{\{\dot\tau,\ddot\tau\}}). \tag{6.79}$$

Remark 5 (Sensitivity to Proposal)**.** *The form of the integrand ξ_t illustrates some of the incremental difficulty of the Markov switching diffusion case. For any two $y_t \neq y_t^\dagger$, the integrand can take large values if the drift behavior of the diffusion differs strongly between states. Large values of the integrand slow down the 2-coin algorithm, increasing the iteration time.*

6.3.4 Conditional Hidden Data Update

The simple independence proposal described in the previous section will tend to break down in various ways as data accrues. We consider the infill asymptotic regime, where mesh $s \to 0$, and the asymptotic extension regime, where $|s|, \omega \to \infty$, and develop strategies to maintain good computational performance in both. Both in the infill and the extension regime, the independence proposal from $\pi(y^\dagger|\lambda)$ will be an increasingly bad fit for the full conditional, causing a degradation in the acceptance probability and slow mixing of the algorithm. Similarly, the independence proposal from $\mathbb{M}|(x_s, y^\dagger, \theta)$ becomes a bad fit as the time interval between observation increases, or in the presence of transitions in y and the associated discontinuities in the drift function. To those obstacles we add the aforementioned phenomenon of the exponential slowdown of the 2-coin algorithm as the time horizon recedes. Thus, we devise a localized, scalable (V_R, Z, Y)-update that addresses both the infill and the extension asymptotic regime.

The approach consists of conditioning the (V_R, Z, Y)-update on V and Y at a random set of times N. If no time is included in N with probability 1, this update may be thought of as a *random scan* Gibbs update, which preserves ergodicity of the Markov chain. The main impediment to doing so is that in its EA2/EA3 representation, Z is only semi-Markovian at times τ, and therefore the full conditional does not factorize neatly at times $\nu \not\subset \tau$. The solution we propose is to introduce a set of *virtual observation times* $\breve s$ at which we also noncenter the diffusion path, resulting in the parameterization

$$h = \{v_{r\cup\breve s}\} \cup \left\{ z_{(\dot\tau,\ddot\tau)} : (\dot\tau\sim\ddot\tau) \in s\cup r\cup\breve s \right\}. \tag{6.80}$$

We then evolve that set by proposing $\breve{s}^\dagger$ in a way that preserves ergodicity, and set $\nu = \breve{s} \cap \breve{s}^\dagger$. When using *virtual observation times* $\breve{s}$, we understand the extended event times τ to be defined as $s \cup r \cup \breve{s}$.

We immediately observe multiple advantages to updating $\pi(h, y|v_{s\cup\nu}, y_\nu, \theta, \lambda)$. On the one hand, the conditional $\kappa(h^\dagger, y^\dagger|v_{s\cup\nu}, y_\nu)$ has a smaller step size, thereby increasing the acceptance probability. On the other hand, by the Markov property, we immediately benefit from the factorizations

$$\begin{aligned}\pi(h, y|v_{s\cup\nu}, y_\nu, \theta, \lambda) = &\prod_{(\dot\nu\sim\ddot\nu)\in\nu} \pi(h_{(\dot\nu,\ddot\nu)}, y_{(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}}, \theta, \lambda)\\ &\times \pi(h_{(0,\nu_1)}, y_{[0,\nu_1)}|v_{s\cup\{\nu_1\}}, y_{\nu_1}, \theta, \lambda)\\ &\times \pi(h_{(\nu_{|\nu|},\omega)}, y_{(\nu_{|\nu|},\omega]}|v_{s\cup\{\nu_{|\nu|}\}}, y_{\nu_{|\nu|}}, \theta, \lambda),\end{aligned} \tag{6.81}$$

$$\begin{aligned}\kappa(h^\dagger, y^\dagger|v_{s\cup\nu}, y_\nu) = &\prod_{(\dot\nu\sim\ddot\nu)\in\nu} \kappa(h^\dagger_{(\dot\nu,\ddot\nu)}, y^\dagger_{(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}})\\ &\times \kappa(h^\dagger_{(0,\nu_1)}, y^\dagger_{[0,\nu_1)}|v_{s\cup\{\nu_1\}}, y_{\nu_1})\\ &\times \kappa(h^\dagger_{(\nu_{|\nu|},\omega)}, y^\dagger_{(\nu_{|\nu|},\omega]}|v_{s\cup\{\nu_{|\nu|}\}}, y_{\nu_{|\nu|}}),\end{aligned} \tag{6.82}$$

where

$$h_{(\dot\nu,\ddot\nu)} = \left\{v_{(r\cup\breve{s})\cap(\dot\nu,\ddot\nu)}\right\} \cup \left\{z_{(\dot\tau,\ddot\tau)} : (\dot\tau \sim \ddot\tau) \in \tau \cap [\dot\nu, \ddot\nu]\right\}, \tag{6.83}$$

so generation and acceptance of the proposal is partitioned according to ν. This further increases the acceptance probability, and reduces Bernoulli factory run time. The proposal law

$$\begin{aligned}&\kappa(h^\dagger_{(\dot\nu,\ddot\nu)}, y^\dagger_{(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}})\\ &\quad= \pi(y^\dagger_{(\dot\nu,\ddot\nu)}|y_{\{\dot\nu,\ddot\nu\}}, \lambda)\kappa(v^\dagger_{(r^\dagger\cup\breve{s}^\dagger)\cap(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y^\dagger_{[\dot\nu,\ddot\nu]}) \prod_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger\cap[\dot\nu,\ddot\nu]} \kappa(z^\dagger_{(\dot\tau,\ddot\tau)})\end{aligned} \tag{6.84}$$

involves the Markov jump process bridge law $\pi(y^\dagger_{(\dot\nu,\ddot\nu)}|y_{\{\dot\nu,\ddot\nu\}}, \lambda)$. Section 6.2.2 gives a summary of [64] on the simulation of such bridges. Notice that the edge proposals

$$\begin{aligned}&\kappa(h^\dagger_{(0,\nu_1)}, y^\dagger_{[0,\nu_1)}|v_{s\cup\{\nu_1\}}, y_{\nu_1})\\ &\quad= \pi(y^\dagger_{[0,\nu_1)}|y_{\nu_1}, \lambda)\kappa(v^\dagger_{(r^\dagger\cup\breve{s}^\dagger)\cap(0,\nu_1)}|v_{s\cup\{0,\nu_1\}}, y^\dagger_{[0,\nu_1]}) \prod_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger\cap[0,\nu_1]} \kappa(z^\dagger_{(\dot\tau,\ddot\tau)}),\end{aligned} \tag{6.85}$$

$$\begin{aligned}&\kappa(h^\dagger_{(\nu_{|\nu|},\omega)}, y^\dagger_{(\nu^\dagger_{|\nu|},\omega]}|v_{s\cup\{\nu_{|\nu|}\}}, y_{\nu_{|\nu|}})\\ &\quad= \pi(y^\dagger_{(\nu_{|\nu|},\omega]}|y_{\nu_{|\nu|}}, \lambda)\kappa(v^\dagger_{(r^\dagger\cup\breve{s}^\dagger)\cap(\nu_{|\nu|},\omega)}|v_{s\cup\{\nu_{|\nu|},\omega\}}, y^\dagger_{[\nu_{|\nu|},\omega]}) \prod_{(\dot\tau\sim\ddot\tau)\in\tau\cap[\nu_{|\nu|},\omega]} \kappa(z^\dagger_{(\dot\tau,\ddot\tau)}),\end{aligned} \tag{6.86}$$

merely involve the backward and forward law $\pi(y^\dagger_{[0,\nu_1)}|y_{\nu_1}, \lambda)$ and $\pi(y^\dagger_{(\nu_{|\nu|},\omega]}|y_{\nu_{|\nu|}}, \lambda)$ respectively.

Simulation of the Markov jump process bridges from $\pi(y^\dagger_{(\dot\nu,\ddot\nu)}|y_{\dot\nu}, y_{\ddot\nu}, \lambda)$ may be carried out according to any of the schemes proposed in Sections 6.2.3, 6.2.4 and 6.2.5, while the terms $\pi(y^\dagger_0|y_{\nu_1}, \lambda)$ and $\pi(y^\dagger_\omega|y_{\nu_{|\nu|}}, \lambda)$ correspond to simple forward and backward simulation respectively. Simulation according to $\kappa(v^\dagger_{(r^\dagger\cup\breve s^\dagger)\cap(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y^\dagger_{[\dot\nu,\ddot\nu]})$ proceeds as in the independence update with Algorithm 18, and the density is given by

$$\begin{aligned}&\kappa(v^\dagger_{(r^\dagger\cup\breve s^\dagger)\cap(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y^\dagger_{[\dot\nu,\ddot\nu]})\\ &\quad= \prod_{(\dot s\sim\ddot s)\in(s\cup\nu)\cap[\dot\nu,\ddot\nu]} \frac{\prod_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger\cap[\dot s,\ddot s]} |\eta_\theta^{-1}(v_{\ddot\tau})|\,\mathrm{N}\left[\eta_\theta(v^\dagger_{\dot\tau}); \eta_\theta(v^\dagger_{\ddot\tau}), (\ddot\tau-\dot\tau)\rho^2_\theta(y^\dagger_{\dot\tau})\right]}{\mathrm{N}\left[\eta_\theta(v_{\ddot s}); \eta_\theta(v_{\dot s}), \sum_{(\dot\tau\sim\ddot\tau)\in r^\dagger\cap(\dot s,\ddot s)}(\ddot\tau-\dot\tau)\rho^2_\theta(y^\dagger_{\dot\tau})\right]},\end{aligned} \tag{6.87}$$

for $(\dot\nu,\ddot\nu)\in\nu\cup\{0,\omega\}$. The proposals $(v^\dagger_{(r^\dagger\cup\breve s^\dagger)\cap(\dot\nu,\ddot\nu)}, z^\dagger_{(\dot\nu,\ddot\nu)}, y^\dagger_{(\dot\nu,\ddot\nu)})$ are accepted with odds

$$\begin{aligned}\frac{\alpha_{(H_{(\dot\nu,\ddot\nu)},Y_{(\dot\nu,\ddot\nu)})}}{1-\alpha_{(H_{(\dot\nu,\ddot\nu)},Y_{(\dot\nu,\ddot\nu)})}} &= \frac{\kappa(h_{(\dot\nu,\ddot\nu)}, y_{(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}})}{\kappa(h^\dagger_{(\dot\nu,\ddot\nu)}, y^\dagger_{(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}})}\\ &\quad\times\frac{\pi(h^\dagger_{(\dot\nu,\ddot\nu)}, y^\dagger_{(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}},\theta,\lambda)}{\pi(h_{(\dot\nu,\ddot\nu)}, y_{(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}},\theta,\lambda)}\\ &= \frac{\kappa(v_{(r\cup\breve s)\cap(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{[\dot\nu,\ddot\nu]})}{\kappa(v^\dagger_{(r^\dagger\cup\breve s^\dagger)\cap(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y^\dagger_{[\dot\nu,\ddot\nu]})}\\ &\quad\times\frac{\prod_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger\cap[\dot\nu,\ddot\nu]}\pi(z^\dagger_{(\dot\tau,\ddot\tau)}, v^\dagger_{\ddot\tau}|v^\dagger_{\dot\tau}, y^\dagger_{\dot\tau},\theta)}{\prod_{(\dot\tau\sim\ddot\tau)\in\tau\cap[\dot\nu,\ddot\nu]}\pi(z_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau},\theta)},\end{aligned} \tag{6.88}$$

where the last equality applies to the edge sections $(h^\dagger_{[0,\nu_1)}, y^\dagger_{[0,\nu_1)})$ and $(h^\dagger_{(\nu_{|\nu|},\omega)}, y^\dagger_{(\nu^\dagger_{|\nu|},\omega]})$ as well.

Updates to $\breve s$ may be carried out in a flexible way, assuming that they are irreducible. Ideally, the size of the overlap $|\nu|$ is locally chosen or adapted to maintain a given acceptance probability in a certain interval, e.g. $(\dot s, \ddot s)$. We proceed to describing such a procedure. Let the time axis be partitioned by a set of times, for example observation times s. In between any neighbouring pair $(\dot s, \ddot s)$ we introduce a proposal scale parameter $\phi_{\dot s,\ddot s}$, i.e. smaller values $\phi_{\dot s,\ddot s}$ result in a more local proposal. We then generate virtual observations at rate Pois $\left[e^{(\dot s-\ddot s)\phi_{\dot s,\ddot s}}\right]$, uniformly deleting elements from $\breve s$ or uniformly sampling additional ones from the interval $(\dot s, \ddot s)$ as needed. The exact algorithm is given below. The hidden data update is then carried out independently for sections $(\dot\nu\sim\ddot\nu)\in\nu$, yielding a corresponding acceptance indicator $\alpha_{\dot\nu,\ddot\nu}$. We compute the average acceptance rate

$$\alpha_{\dot s,\ddot s} = \frac{\sum_{(\dot\nu\sim\ddot\nu)\in\nu:(\dot\nu,\ddot\nu)\cap(\dot s,\ddot s)\neq\emptyset}\alpha_{\dot\nu,\ddot\nu}}{|\{(\dot\nu\sim\ddot\nu)\in\nu:(\dot\nu,\ddot\nu)\cap(\dot s,\ddot s)\neq\emptyset\}|}, \tag{6.89}$$

taking into account all updates that overlap with $(\dot s, \ddot s)$. We then apply a standard Robbins-Monro recursion to increase $\phi_{\dot s,\ddot s}$ if $\alpha_{\dot s,\ddot s}$ is above the target acceptance rate and vice versa.

A slight complication arises when $\breve{s}$, and therefore ν, is empty in a part of the trajectory where the acceptance rate is low. This will cause rejection of $\breve{s}^\dagger$, and failure to populate ν and increase the acceptance rate. A workaround consists of adding the observation times s to ν with probability $p_s \ll 1$. This will facilitate acceptance of $\breve{s}^\dagger$, making subsequent updates easier. To preserve irreducibility, p_s must be less than one. Otherwise, Y would forever be held constant at the observation times. Moreover, since adding s to ν increases the acceptance probability, the algorithm should not be adapted on those iterations.

Algorithm 19 Algorithm for generating virtual observation times.

$\breve{s}^\dagger \leftarrow \emptyset$
for $(\dot{s} \sim \ddot{s}) \in s$ **do**
  $n_{\dot{s},\ddot{s}} \sim \text{Pois}\left[e^{(\dot{s}-\ddot{s})\phi_{\dot{s},\ddot{s}}}\right]$
  if $n_{\dot{s},\ddot{s}} > |\breve{s} \cap [\dot{s},\ddot{s}]|$ **then**
    append $\breve{s} \cap [\dot{s},\ddot{s}]$ to $\breve{s}^\dagger$
    draw $n_{\dot{s},\ddot{s}} - |\breve{s} \cap [\dot{s},\ddot{s}]|$ samples from $\text{Unif}\,[\dot{s},\ddot{s}]$ and append them to $\breve{s}^\dagger$
  else
    draw $n_{\dot{s},\ddot{s}}$ elements without replacement from $\breve{s} \cap [\dot{s},\ddot{s}]$ and append them to $\breve{s}^\dagger$
$\nu \leftarrow \breve{s} \cap \breve{s}^\dagger$
With probability $p_s \in (0,1)$, append s to ν

Notice that while the conditional hidden data update mostly addresses the concerns raised in this Section, it does not address the fundamental issue that ξ_t is highly discontinuous in y_t if the drift of the diffusion strongly differs between the available states, and the algorithm remains liable to visit or propose moves to parts of the state space where the cost of a single iteration is very large.

6.4 Auxiliary Algorithm

We continue with the development a fully tractable MCMC algorithm that targets the extended posterior

$$\pi(v_r, \psi, z, y, \theta, \lambda | v_s) \propto \pi(\theta)\pi(\lambda) \prod_{(\dot{\tau} \sim \ddot{\tau}) \in \tau} \pi(\psi_{(\dot{\tau},\ddot{\tau})}, z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}} | v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}} | y_{\dot{\tau}}, \lambda). \tag{6.90}$$

We construct a Gibbs sampler with full conditionals

$$\pi(\theta, \lambda | v_\tau, \psi, z, y) \propto \pi(\theta)\pi(\lambda) \prod_{(\dot{\tau} \sim \ddot{\tau}) \in \tau} \pi(\psi_{(\dot{\tau},\ddot{\tau})}, z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}} | v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}} | y_{\dot{\tau}}, \lambda), \tag{6.91}$$

$$\pi(v_r, \psi, z, y | v_s, \theta, \lambda) \propto \prod_{(\dot{\tau} \sim \ddot{\tau}) \in \tau} \pi(\psi_{(\dot{\tau},\ddot{\tau})}, z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}} | v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}} | y_{\dot{\tau}}, \lambda), \tag{6.92}$$

where the dominating measure of the latter is the product measure

$$\mathbb{L}(\mathrm{d}y) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \mathbb{P}_{(\dot{\tau},\ddot{\tau})}(\mathrm{d}\psi_{(\dot{\tau},\ddot{\tau})})\mathbb{B}_{(\dot{\tau},\ddot{\tau})}(\mathrm{d}z_{(\dot{\tau},\ddot{\tau})}) \prod_{\dot{r}\in r} \mathrm{Leb}(\mathrm{d}v_{\dot{r}}). \tag{6.93}$$

In contrast with the marginal algorithm of Section 6.3, there is the disadvantage of additional conditioning in the Θ-update. The (Θ, Λ) update decomposes into the independent updates

$$\Theta: \quad \pi(\theta|v_\tau, \psi, z, y) \propto \pi(\theta) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(\psi_{(\dot{\tau},\ddot{\tau})}, z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta), \tag{6.94}$$

$$\Lambda: \quad \pi(\lambda|y) \propto \pi(\lambda) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(y_{\ddot{\tau}}|y_{\dot{\tau}}, \lambda). \tag{6.95}$$

Furthermore, the Θ-update simplifies further for models in the class

$$\mathrm{d}V_t = \mu_{\theta_{Y_t}}(V_t)\,\mathrm{d}t + \sigma(V_t)\rho_{\theta_{Y_t}}\,\mathrm{d}W_t, \quad \theta = (\theta_1, \ldots, \theta_{\hat{k}}), \tag{6.96}$$

with product prior $\pi(\theta) = \prod_{k=1}^{\hat{k}} \pi(\theta_k)$. Now, each Θ_k may be updated independently according to

$$\pi(\theta_k|v_\tau, \psi, z, y) \propto \pi(\theta_k) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau: y_{\dot{\tau}}=k} \pi(\psi_{(\dot{\tau},\ddot{\tau})}, z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta_k). \tag{6.97}$$

Remark 6 (Infinite-dimensional representation)**.** *Our representation of the infinite-dimensional objects z and ψ remains unchanged from Section 5.3.1.*

6.4.1 Diffusion Parameter Update

We implement the update to $\pi(\theta|v_\tau, \psi, z, y)$ as a Metropolis-within-Gibbs update, analogously to Section 5.3.2. For a generic proposal $\kappa(\theta^\dagger|\theta)$, the acceptance probability is

$$\begin{aligned}
\alpha_\Theta &= 1 \wedge \frac{\pi(\theta^\dagger|v_s, \psi, z, y)}{\pi(\theta|v_s, \psi, z, y)} \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)} \\
&= 1 \wedge \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)} \frac{\pi(\theta^\dagger)}{\pi(\theta)} \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \frac{\pi(\psi_{(\dot{\tau},\ddot{\tau})}, z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta^\dagger)}{\pi(\psi_{(\dot{\tau},\ddot{\tau})}, z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)} \\
&= 1 \wedge \frac{\kappa(\theta|\theta^\dagger)}{\kappa(\theta^\dagger|\theta)} \frac{\pi(\theta^\dagger)}{\pi(\theta)} \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \frac{d_{\theta^\dagger}(v_{\{\dot{\tau},\ddot{\tau}\}}, y_{\dot{\tau}})}{d_\theta(v_{\{\dot{\tau},\ddot{\tau}\}}, y_{\dot{\tau}})} \frac{\bar{q}_{\theta^\dagger}(\psi_{(\dot{\tau},\ddot{\tau})}, z_{(\dot{\tau},\ddot{\tau})}, v_{\{\dot{\tau},\ddot{\tau}\}}, y_{\dot{\tau}})}{\bar{q}_\theta(\psi_{(\dot{\tau},\ddot{\tau})}, z_{(\dot{\tau},\ddot{\tau})}, v_{\{\dot{\tau},\ddot{\tau}\}}, y_{\dot{\tau}})}.
\end{aligned} \tag{6.98}$$

6.4.2 Regime Parameter Update

The update to $\pi(\lambda|y)$ is identical to the one given in Section 6.3.2, and carried out by sampling directly from the tractable full conditional.

6.4.3 Independence Hidden Data Update

We implement the update to $\pi(v_r, \psi, h, y|v_s, \theta, \lambda)$ as a Metropolis-within-Gibbs update with independence proposal

$$\begin{aligned}\kappa(v^\dagger_{r^\dagger}, \psi^\dagger, z^\dagger, y^\dagger|v_s) &\propto \kappa(v^\dagger_{r^\dagger}, z^\dagger, y^\dagger|v_s)\kappa(\psi^\dagger|y^\dagger) \\ &\propto \kappa(y^\dagger)\kappa(v^\dagger_{r^\dagger}|v_s, y^\dagger) \prod_{(\dot\tau\sim\ddot\tau)\in\tau} \kappa(z^\dagger_{(\dot\tau,\ddot\tau)})\kappa(\psi^\dagger_{(\dot\tau,\ddot\tau)}),\end{aligned} \tag{6.99}$$

where $(V^\dagger_{R^\dagger}, Z^\dagger, Y^\dagger)$ is simulated and $\kappa(v^\dagger_{r^\dagger}, \psi^\dagger, z^\dagger|v_s)$ evaluated as in Section 6.3.3. In particular, $\kappa(v^\dagger_{r^\dagger}|v_s, y^\dagger)$ is given by (6.74). We further set $\Psi^\dagger_{(\dot\tau,\ddot\tau)} \sim \mathbb{P}_{(\dot\tau,\ddot\tau)}$ with $\kappa(\psi^\dagger_{(\dot\tau,\ddot\tau)}) = 1$ with respect to the dominating measure $\mathbb{P}_{(\dot\tau,\ddot\tau)}$. The acceptance probability is

$$\begin{aligned}&\alpha_{(V_R,\Psi,Z,Y)} \\ &= 1 \wedge \frac{\kappa(v_r, \psi, z, y|v_s)}{\kappa(v^\dagger_{r^\dagger}, \psi^\dagger, z^\dagger, y^\dagger|v_s)} \frac{\pi(y^\dagger|\lambda)}{\pi(y|\lambda)} \frac{\pi(v^\dagger_{r^\dagger}, \psi^\dagger, z^\dagger, y^\dagger|v_s, \theta, \lambda)}{\pi(v_r, \psi, z, y|v_s, \theta, \lambda)} \\ &= 1 \wedge \frac{\kappa(v_r|v_s, y)}{\kappa(v^\dagger_{r^\dagger}|v_s, y^\dagger)} \frac{\prod_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger} \pi(\psi^\dagger_{(\dot\tau,\ddot\tau)}, z^\dagger_{(\dot\tau,\ddot\tau)}, v^\dagger_{\ddot\tau}|v^\dagger_{\dot\tau}, y^\dagger_{\dot\tau}, \theta)}{\prod_{(\dot\tau\sim\ddot\tau)\in\tau} \pi(\psi_{(\dot\tau,\ddot\tau)}, z_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta)} \\ &= 1 \wedge \frac{\kappa(v_r|v_s, y)}{\kappa(v^\dagger_{r^\dagger}|v_s, y^\dagger)} \frac{\prod_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger} d_\theta(v^\dagger_{\{\dot\tau,\ddot\tau\}}, y^\dagger_{\dot\tau})\bar q_\theta(\psi^\dagger_{(\dot\tau,\ddot\tau)}, z^\dagger_{(\dot\tau,\ddot\tau)}, v^\dagger_{\{\dot\tau,\ddot\tau\}}, y^\dagger_{\ddot\tau})}{\prod_{(\dot\tau\sim\ddot\tau)\in\tau} d_\theta(v_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau})\bar q_\theta(\psi_{(\dot\tau,\ddot\tau)}, z_{(\dot\tau,\ddot\tau)}, v_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau})}.\end{aligned} \tag{6.100}$$

Remark 7 (Sensitivity to Proposal)**.** *This update runs into similar proposal sensitivity issues as in Section 6.3.3, since evaluation of $\bar q_\theta(\psi^\dagger_{(\dot\tau,\ddot\tau)}, z^\dagger_{(\dot\tau,\ddot\tau)}, v^\dagger_{\{\dot\tau,\ddot\tau\}}, y^\dagger_{\ddot\tau})$ might be much more expensive for a proposed $y^\dagger_{\dot\tau} \neq y_{\dot\tau}$.*

6.4.4 Conditional Hidden Data Update

The logic of the conditional hidden data of the marginal algorithm of Section 6.3.4 carries over 1 to 1. Since the elements of τ-partitions of Ψ are a priori independent, we obtain factorized conditionals and proposal densities:

$$\begin{aligned}\pi(\psi, h, y|v_{s\cup\nu}, y_\nu, \theta, \lambda) = &\prod_{(\dot\nu\sim\ddot\nu)\in\nu} \pi(\psi_{(\dot\nu,\ddot\nu)}, h_{(\dot\nu,\ddot\nu)}, y_{(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}}, \theta, \lambda) \\ &\times \pi(\psi_{(0,\nu_1)}, h_{(0,\nu_1)}, y_{[0,\nu_1)}|v_{s\cup\{\nu_1\}}, y_{\nu_1}, \theta, \lambda) \\ &\times \pi(\psi_{(\nu_{|\nu|},\omega)}, h_{(\nu_{|\nu|},\omega)}, y_{(\nu_{|\nu|},\omega]}|v_{s\cup\{\nu_{|\nu|}\}}, y_{\nu_{|\nu|}}, \theta, \lambda),\end{aligned} \tag{6.101}$$

$$\begin{aligned}\kappa(\psi^\dagger, h^\dagger, y^\dagger|v_{s\cup\nu}, y_\nu) = &\prod_{(\dot\nu\sim\ddot\nu)\in\nu} \kappa(\psi^\dagger_{(\dot\nu,\ddot\nu)}, h^\dagger_{(\dot\nu,\ddot\nu)}, y^\dagger_{(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}}) \\ &\times \kappa(\psi^\dagger_{(0,\nu_1)}, h^\dagger_{(0,\nu_1)}, y^\dagger_{[0,\nu_1)}|v_{s\cup\{\nu_1\}}, y_{\nu_1}) \\ &\times \kappa(\psi^\dagger_{(\nu_{|\nu|},\omega)}, h^\dagger_{(\nu_{|\nu|},\omega)}, y^\dagger_{(\nu_{|\nu|},\omega]}|v_{s\cup\{\nu_{|\nu|}\}}, y_{\nu_{|\nu|}}).\end{aligned} \tag{6.102}$$

Proposals $(\psi^\dagger_{(\dot\nu,\ddot\nu)}, h^\dagger_{(\dot\nu,\ddot\nu)}, y^\dagger_{(\dot\nu,\ddot\nu)})$ are accepted with probability

$$\begin{aligned}\alpha_{(\Psi_{(\dot\nu,\ddot\nu)},H_{(\dot\nu,\ddot\nu)},Y_{(\dot\nu,\ddot\nu)})} &= 1 \wedge \frac{\kappa(\psi_{(\dot\nu,\ddot\nu)}, h_{(\dot\nu,\ddot\nu)}, y_{(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}})}{\kappa(\psi^\dagger_{(\dot\nu,\ddot\nu)}, h^\dagger_{(\dot\nu,\ddot\nu)}, y^\dagger_{(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}})} \\ &\quad\times \frac{\pi(\psi^\dagger_{(\dot\nu,\ddot\nu)}, h^\dagger_{(\dot\nu,\ddot\nu)}, y^\dagger_{(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}}, \theta, \lambda)}{\pi(\psi_{(\dot\nu,\ddot\nu)}, h_{(\dot\nu,\ddot\nu)}, y_{(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}}, \theta, \lambda)} \\ &= 1 \wedge \frac{\kappa(v_{(r\cup\check s)\cap(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{[\dot\nu,\ddot\nu]})}{\kappa(v^\dagger_{(r^\dagger\cup\check s^\dagger)\cap(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y^\dagger_{[\dot\nu,\ddot\nu]})} \\ &\quad\times \frac{\prod_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger\cap[\dot\nu,\ddot\nu]} \pi(\psi^\dagger_{(\dot\tau,\ddot\tau)}, z^\dagger_{(\dot\tau,\ddot\tau)}, v^\dagger_{\ddot\tau}|v^\dagger_{\dot\tau}, y^\dagger_{\dot\tau}, \theta)}{\prod_{(\dot\tau\sim\ddot\tau)\in\tau\cap[\dot\nu,\ddot\nu]} \pi(\psi^\dagger_{(\dot\tau,\ddot\tau)}, z_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta)},\end{aligned} \tag{6.103}$$

where the last equality applies to the edge sections $(\psi^\dagger_{[0,\nu_1)}, h^\dagger_{[0,\nu_1)}, y^\dagger_{[0,\nu_1)})$ and $(\psi^\dagger_{(\nu_{|\nu|},\omega)}, h^\dagger_{(\nu_{|\nu|},\omega)}, y^\dagger_{(\nu^\dagger_{|\nu|},\omega]})$ as well, and all elements have been defined previously. In particular, $\kappa(v_{(r\cup\check s)\cap(\dot\nu,\ddot\nu)}|v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{[\dot\nu,\ddot\nu]})$ is given by (6.87). Adaptation is done with the same scheme as in Algorithm 19.

6.5 Approximate Algorithm

We now propose an approximate Bayesian inference algorithm building on Section 5.4. Let $u_{[\dot\tau,\ddot\tau]}$ be a partition of $[\dot\tau,\ddot\tau]$ and define $\bar X_{(\dot\tau,\ddot\tau)} = X_{u_{[\dot\tau,\ddot\tau]}\setminus\{\dot\tau,\ddot\tau\}}$. We obtain the approximate density

$$\bar\pi(v_{\ddot\tau}, \bar x_{(\dot\tau,\ddot\tau)}|v_{\dot\tau}, y_{\dot\tau}, \theta) = |\eta'_\theta(v_{\ddot\tau})| \prod_{(\dot u\sim\ddot u)\in u_{[\dot\tau,\ddot\tau]}} \mathrm{N}\left[x_{\ddot\tau}; {}^{x_{\dot\tau}+(\ddot\tau-\dot\tau)\delta_\theta(x_{\dot\tau}),}_{(\ddot\tau-\dot\tau)\rho^2_\theta(y_{\dot\tau})}\right], \tag{6.104}$$

$$\lim_{\text{mesh } u_{[\dot\tau,\ddot\tau]}\to 0} \mathrm{E}_{\bar X_{(\dot\tau,\ddot\tau)}}\left[\bar\pi(v_{\ddot\tau}, \bar X_{(\dot\tau,\ddot\tau)}|v_{\dot\tau}, y_{\dot\tau}, \theta)|v_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau}, \theta\right] = \pi(v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta). \tag{6.105}$$

As in the Itō diffusion case, due to weak convergence the scheme recovers the exact transition density as the maximum discretization interval mesh $u_{[\dot\tau,\ddot\tau]} \to 0$ goes to 0. The corresponding noncentered approximate density is

$$\begin{aligned}&\bar\pi(v_{\ddot\tau}, \bar z_{(\dot\tau,\ddot\tau)}|v_{\dot\tau}, y_{\dot\tau}, \theta) \\ &\quad= |\eta'_\theta(v_{\ddot u})| \prod_{z_t\in\bar z_{(\dot\tau,\ddot\tau)}} \left|\partial_{z_t}\zeta^{-1}_\theta(z_t; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}})\right| \\ &\qquad\prod_{(\dot u\sim\ddot u)\in u_{[\dot\tau,\ddot\tau]}} \mathrm{N}\left[\zeta^{-1}_\theta(z_{\ddot u}; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}); {}^{\zeta^{-1}_\theta(z_{\dot u}; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}})+(\ddot u-\dot u)\tilde\delta_\theta(z_{\dot u}, y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}),}_{(\ddot u-\dot u)\rho^2_\theta(y_{\dot\tau})}\right].\end{aligned} \tag{6.106}$$

where we slightly abuse notation by setting $\zeta^{-1}_\theta(z_{\dot\tau}; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}) = v_{\dot\tau}$ and $\zeta^{-1}_\theta(z_{\ddot\tau}; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}) = v_{\ddot\tau}$, and the Jacobian is given by

$$\left|\partial_{z_t}\zeta^{-1}_\theta(z_t; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}})\right| = \rho_\theta(y_{\dot\tau}). \qquad (t\in(\dot\tau,\ddot\tau)) \tag{6.107}$$

This parameterization of the missing data conserves ergodicity as mesh $u_{[\dot{\tau},\ddot{\tau}]} \to 0$, and gives us a viable, approximate augmentation scheme within the same Gibbs blocking scheme as in the marginal algorithm of Section 6.3. The approximate posterior targeted by that sampler is

$$\bar{\pi}(v_r, \bar{z}, y, \theta, \lambda | v_s) \propto \pi(\theta)\pi(\lambda) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \bar{\pi}(v_{\ddot{\tau}}, \bar{z}_{(\dot{\tau},\ddot{\tau})} | v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}} | y_{\dot{\tau}}, \lambda), \tag{6.108}$$

and its Gibbs updates are

$$(V_R, \bar{Z}, Y): \quad \bar{\pi}(v_r, \bar{z}, y | v_s, \theta, \lambda) \propto \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \bar{\pi}(\bar{z}_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}} | v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}} | y_{\dot{\tau}}, \lambda), \tag{6.109}$$

$$\Theta: \quad \bar{\pi}(\theta | v_\tau, \bar{z}, y) \propto \pi(\theta) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \bar{\pi}(\bar{z}_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}} | v_{\dot{\tau}}, y_{\dot{\tau}}, \theta), \tag{6.110}$$

$$\lambda: \quad \pi(\lambda | y) \propto \pi(\lambda)\pi(y | \lambda). \tag{6.111}$$

Since the inference problem for Markov switching diffusions is harder than for Itō diffusions, warm-starting the exact algorithms from a run of the approximate algorithm is even more important. We follow that practice in our simulation studies.

6.5.1 Diffusion Parameter Update

We implement the update to $\bar{\pi}(\theta | v_\tau, z, y)$ as a Metropolis-within-Gibbs update. For a generic proposal $\kappa(\theta^\dagger | \theta)$, the acceptance probability is

$$\begin{aligned} \alpha_\Theta &= 1 \wedge \frac{\bar{\pi}(\theta^\dagger | v_\tau, \bar{z}, y)}{\bar{\pi}(\theta | v_\tau, \bar{z}, y)} \frac{\kappa(\theta | \theta^\dagger)}{\kappa(\theta^\dagger | \theta)} \\ &= 1 \wedge \frac{\kappa(\theta | \theta^\dagger)}{\kappa(\theta^\dagger | \theta)} \frac{\pi(\theta^\dagger)}{\pi(\theta)} \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \frac{\bar{\pi}(\bar{z}_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}} | v_{\dot{\tau}}, y_{\dot{\tau}}, \theta^\dagger)}{\bar{\pi}(\bar{z}_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}} | v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)}. \end{aligned} \tag{6.112}$$

6.5.2 Regime Parameter Update

The update to $\pi(\lambda | y)$ is identical to the one given in Section 6.3.2, and carried out by sampling directly from the tractable full conditional.

6.5.3 Independence Hidden Data Update

We again implement the update $\bar{\pi}(v_r, \bar{z}, y | v_s, \theta, \lambda)$ as a Metropolis-within-Gibbs update, with hierarchical proposal

$$\begin{aligned} \kappa(v_{r\dagger}^\dagger, \bar{z}^\dagger, y^\dagger | v_s) &= \kappa(v_{r\dagger}^\dagger, y^\dagger | v_s)\kappa(\bar{z}^\dagger | y^\dagger) \\ &\propto \kappa(y^\dagger)\kappa(v_r^\dagger | v_s, y^\dagger) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \kappa(\bar{z}_{(\dot{\tau},\ddot{\tau})}^\dagger), \end{aligned} \tag{6.113}$$

where $\kappa(v^\dagger_{r^\dagger}, y^\dagger | v_s)$ is constructed as in Sections 6.3.3 and 6.5.3. $\bar{Z}^\dagger_{(\dot\tau,\ddot\tau)}$ follows $\mathbb{B}_{(\dot\tau,\ddot\tau)}$ and has density with respect to $\mathrm{Leb}^{|\bar{z}^\dagger_{(\dot\tau,\ddot\tau)}|}$ given by

$$\kappa(\bar{z}^\dagger_{(\dot\tau,\ddot\tau)}) = \frac{\prod_{(\dot u \sim \ddot u) \in u_{[\dot\tau,\ddot\tau]}} \mathrm{N}\left[z^\dagger_{\ddot u}; z^\dagger_{\dot u}, \ddot u - \dot u\right]}{\mathrm{N}\left[0; 0, \ddot\tau - \dot\tau\right]}. \tag{6.114}$$

We empirically observed that setting $u_{[\dot\tau,\ddot\tau]}$ deterministically was detrimental to mixing when $u_{[\dot\tau,\ddot\tau]}$ was chosen to be dense. Randomizing $u_{[\dot\tau,\ddot\tau]}$ as a fixed-rate Poisson process somewhat alleviated that issue. We accept the proposal with probability

$$\begin{aligned}\alpha_{(H,Y)} &= 1 \wedge \frac{\kappa(v^\dagger_{r^\dagger}, y^\dagger | v_s)}{\kappa(v_r, y | v_s)} \frac{\prod_{(\dot\tau\sim\ddot\tau)\in\tau} \kappa(\bar{z}_{(\dot\tau,\ddot\tau)})}{\prod_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger} \kappa(\bar{z}^\dagger_{(\dot\tau,\ddot\tau)})} \frac{\prod_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger} \bar\pi(\bar{z}^\dagger_{(\dot\tau,\ddot\tau)} | v_{\{\dot\tau,\ddot\tau\}}, \theta)}{\prod_{(\dot\tau\sim\ddot\tau)\in\tau} \bar\pi(\bar{z}_{(\dot\tau,\ddot\tau)} | v_{\{\dot\tau,\ddot\tau\}}, \theta)} \\ &= 1 \wedge \frac{\kappa(v^\dagger_{r^\dagger}, y^\dagger | v_s)}{\kappa(v_r, y | v_s)} \frac{\prod_{(\dot\tau\sim\ddot\tau)\in\tau} \kappa(\bar{z}_{(\dot\tau,\ddot\tau)})}{\prod_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger} \kappa(\bar{z}^\dagger_{(\dot\tau,\ddot\tau)})} \frac{\prod_{(\dot\tau\sim\ddot\tau)\in\tau^\dagger} \bar\pi(\bar{z}^\dagger_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau} | v_{\dot\tau}, y_{\dot\tau}, \theta)}{\prod_{(\dot\tau\sim\ddot\tau)\in\tau} \bar\pi(\bar{z}_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau} | v_{\dot\tau}, y_{\dot\tau}, \theta)}.\end{aligned} \tag{6.115}$$

6.5.4 Conditional Hidden Data Update

While in this instance it is not strictly necessary to adhere to the virtual observation scheme of Section 6.3.4, for the purposes of warm starting the exact algorithm it is useful to follow the same approach to conditional updates. Fix a set of virtual observation times $\breve{s}$ and define

$$\bar{h} = \{v_{r\cup\breve{s}}\} \cup \left\{\bar{z}_{(\dot\tau,\ddot\tau)} : (\dot\tau \sim \ddot\tau) \in s \cup r \cup \breve{s}\right\}. \tag{6.116}$$

For some finite set of conditioning times $\nu = \breve{s} \cap \breve{s}^\dagger$, we obtain the analogous factorizations

$$\begin{aligned}\bar\pi(\bar{h}, y | v_{s\cup\nu}, y_\nu, \theta, \lambda) = &\prod_{(\dot\nu\sim\ddot\nu)\in\nu} \bar\pi(\bar{h}_{(\dot\nu,\ddot\nu)}, y_{(\dot\nu,\ddot\nu)} | v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}}, \theta, \lambda) \\ &\times \bar\pi(\bar{h}_{(0,\nu_1)}, y_{[0,\nu_1)} | v_{s\cup\{\nu_1\}}, y_{\nu_1}, \theta, \lambda) \\ &\times \bar\pi(\bar{h}_{(\nu_{|\nu|},\omega)}, y_{(\nu_{|\nu|},\omega]} | v_{s\cup\{\nu_{|\nu|}\}}, y_{\nu_{|\nu|}}, \theta, \lambda),\end{aligned} \tag{6.117}$$

$$\begin{aligned}\kappa(\bar{h}^\dagger, y^\dagger | v_{s\cup\nu}, y_\nu) = &\prod_{(\dot\nu\sim\ddot\nu)\in\nu} \kappa(\bar{h}^\dagger_{(\dot\nu,\ddot\nu)}, y^\dagger_{(\dot\nu,\ddot\nu)} | v_{s\cup\{\dot\nu,\ddot\nu\}}, y_{\{\dot\nu,\ddot\nu\}}) \\ &\times \kappa(\bar{h}^\dagger_{(0,\nu_1)}, y^\dagger_{[0,\nu_1)} | v_{s\cup\{\nu_1\}}, y_{\nu_1}) \\ &\times \kappa(\bar{h}^\dagger_{(\nu_{|\nu|},\omega)}, y^\dagger_{(\nu_{|\nu|},\omega]} | v_{s\cup\{\nu_{|\nu|}\}}, y_{\nu_{|\nu|}}).\end{aligned} \tag{6.118}$$

We accept $(\bar{h}^{\dagger}_{(\dot{\nu},\ddot{\nu})}, y^{\dagger}_{(\dot{\nu},\ddot{\nu})})$ with probability

$$\begin{aligned}\alpha_{(\bar{H}_{(\dot{\nu},\ddot{\nu})},Y_{(\dot{\nu},\ddot{\nu})})} &= 1 \wedge \frac{\kappa(\bar{h}_{(\dot{\nu},\ddot{\nu})}, y_{(\dot{\nu},\ddot{\nu})}|v_{s\cup\{\dot{\nu},\ddot{\nu}\}}, y_{\{\dot{\nu},\ddot{\nu}\}})}{\kappa(\bar{h}^{\dagger}_{(\dot{\nu},\ddot{\nu})}, y^{\dagger}_{(\dot{\nu},\ddot{\nu})}|v_{s\cup\{\dot{\nu},\ddot{\nu}\}}, y_{\{\dot{\nu},\ddot{\nu}\}})} \\ &\quad \times \frac{\bar{\pi}(\bar{h}^{\dagger}_{(\dot{\nu},\ddot{\nu})}, y^{\dagger}_{(\dot{\nu},\ddot{\nu})}|v_{s\cup\{\dot{\nu},\ddot{\nu}\}}, y_{\{\dot{\nu},\ddot{\nu}\}}, \theta, \lambda)}{\bar{\pi}(\bar{h}_{(\dot{\nu},\ddot{\nu})}, y_{(\dot{\nu},\ddot{\nu})}|v_{s\cup\{\dot{\nu},\ddot{\nu}\}}, y_{\{\dot{\nu},\ddot{\nu}\}}, \theta, \lambda)} \\ &= 1 \wedge \frac{\kappa(v_{(r\cup\check{s})\cap(\dot{\nu},\ddot{\nu})}|v_{s\cup\{\dot{\nu},\ddot{\nu}\}}, y_{[\dot{\nu},\ddot{\nu}]})}{\kappa(v^{\dagger}_{(r^{\dagger}\cup\check{s}^{\dagger})\cap(\dot{\nu},\ddot{\nu})}|v_{s\cup\{\dot{\nu},\ddot{\nu}\}}, y^{\dagger}_{[\dot{\nu},\ddot{\nu}]})} \frac{\prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau\cap[\dot{\nu},\ddot{\nu}]} \kappa(\bar{z}_{(\dot{\tau},\ddot{\tau})})}{\prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau^{\dagger}\cap[\dot{\nu},\ddot{\nu}]} \kappa(\bar{z}^{\dagger}_{(\dot{\tau},\ddot{\tau})})} \\ &\quad \times \frac{\prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau^{\dagger}\cap[\dot{\nu},\ddot{\nu}]} \bar{\pi}(\bar{z}^{\dagger}_{(\dot{\tau},\ddot{\tau})}, v^{\dagger}_{\ddot{\tau}}|v^{\dagger}_{\dot{\tau}}, y^{\dagger}_{\dot{\tau}}, \theta)}{\prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau\cap[\dot{\nu},\ddot{\nu}]} \bar{\pi}(\bar{z}_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)},\end{aligned} \tag{6.119}$$

where the last equality applies to the edge sections $(\bar{h}^{\dagger}_{[0,\nu_1)}, y^{\dagger}_{[0,\nu_1)})$ and $(\bar{h}^{\dagger}_{(\nu_{|\nu|},\omega)}, y^{\dagger}_{(\nu^{\dagger}_{|\nu|},\omega]})$ as well, and all elements have been defined previously. In particular, $\kappa(v_{(r\cup\check{s})\cap(\dot{\nu},\ddot{\nu})}|v_{s\cup\{\dot{\nu},\ddot{\nu}\}}, y_{[\dot{\nu},\ddot{\nu}]})$ is given by (6.87). Adaptation is done with the same scheme as in Section 6.3.3.

6.6 MAP Estimation

We now construct an algorithm for MAP estimation, i.e. finding the set of values $(\theta^{\ddagger}, \lambda^{\ddagger})$ such that

$$(\theta^{\ddagger}, \lambda^{\ddagger}) = \underset{\theta,\lambda}{\operatorname{argmax}}\ \pi(\theta, \lambda, v_{s\setminus\{0\}}|v_0). \tag{6.120}$$

The MAP estimator also corresponds to the maximum likelihood estimator when setting $\pi(\theta, \lambda) \propto 1$. In this section, we combine the MAP estimation algorithm from Section 5.5 with the new tools developed in this chapter to obtain a corresponding MCEM algorithm for Markov switching diffusions.

6.6.1 Log Transition Density Estimation

In order to construct a MCEM algorithm, we require an unbiased estimator of the path integral in the log complete transition density:

$$\log \pi(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta) = \log d_{\theta}(v_{\{\dot{\tau},\ddot{\tau}\}}, y_{\dot{\tau}}) - \int_{\dot{\tau}}^{\ddot{\tau}} \varphi_{\theta}(x_t, y_{\dot{\tau}})\,\mathrm{d}t. \tag{6.121}$$

Unbiased estimation is easily accomplished by uniform sampling along the path:

$$-(\ddot{\tau} - \dot{\tau})\varphi_{\theta}(x_U, y_{\dot{\tau}}), \quad U \sim \mathrm{Unif}\,[\dot{\tau}, \ddot{\tau}]\,. \tag{6.122}$$

Thus, we define the log augmented transition density estimator

$$\bar{\ell}_u(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta) = \log d_\theta(v_{\{\dot{\tau},\ddot{\tau}\}}, y_{\dot{\tau}}) - (\ddot{\tau} - \dot{\tau})\varphi_\theta(x_u, y_{\dot{\tau}}), \tag{6.123}$$

$$\mathrm{E}_U\left[\bar{\ell}_U(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\right] = \log \pi(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta). \tag{6.124}$$

The relative simplicity of the log augmented transition estimator is the main benefit of taking the MCEM approach.

6.6.2 E-Step

The E-step consists of finding a lower bound on the objective $\pi(\theta, \lambda, v_{s\setminus\{0\}}|v_0)$. It is obtained by averaging the joint density over the posterior of the latent variables, i.e.

$$\begin{aligned} Q(\theta^\dagger, \lambda^\dagger, \theta, \lambda) &= \mathrm{E}_{V_R,Z,Y}\left[\log \pi(V_R, Z, Y, \theta^\dagger, \lambda^\dagger, v_{s\setminus\{0\}}|v_0)|v_s, \theta, \lambda\right] \\ &= \mathrm{E}_{V_R,Z,Y}\left[\log \pi(V_R, Z, v_{s\setminus\{0\}}|Y, \theta^\dagger, v_0) + \log \pi(Y|\lambda^\dagger)|v_s, \theta, \lambda\right] \\ &\quad + \log \pi(\theta^\dagger) + \log \pi(\lambda^\dagger). \end{aligned} \tag{6.125}$$

where we take expectations with respect to $\pi(v_r, z, y|v_s, \theta, \lambda)$. Because the Q-function decomposes into separate functions of $\theta^\dagger$ and $\lambda^\dagger$, we define separate Q-functions

$$Q_\Theta(\theta^\dagger, \theta) = \mathrm{E}_{V_R,Z,Y}\left[\log \pi(V_R, Z, v_{s\setminus\{0\}}|v_0, y, \theta^\dagger)|v_s, \theta, \lambda\right] + \log \pi(\theta^\dagger), \tag{6.126}$$

$$Q_\Lambda(\lambda^\dagger, \lambda) = \mathrm{E}_{V_R,Z,Y}\left[\log \pi(Y|\lambda^\dagger)|v_s, \theta, \lambda\right] + \log \pi(\lambda^\dagger). \tag{6.127}$$

We may obtain samples from $\pi(v_r, z, y|v_s, \theta, \lambda)$ by iterating the corresponding MCMC update from Section 6.3.3 or 6.3.4. That results in a dependent sequence of $\hat{l}$ samples $(v_r^{(l)}, z^{(l)}, y^{(l)})$, to which we independently add the uniform variates $(u^{(l)}_{(\dot{\tau},\ddot{\tau})} : (\dot{\tau} \sim \ddot{\tau}) \in \tau^{(l)})$. Given such a sequence, we obtain the unbiased Q-estimators:

$$\bar{Q}_\Theta(\theta^\dagger) = \log \pi(\theta^\dagger) + \hat{l}^{-1} \sum_{l=1}^{\hat{l}} \sum_{(\dot{\tau}\sim\ddot{\tau})\in\tau^{(l)}} \bar{\ell}_{u^{(l)}_{\dot{\tau},\ddot{\tau}}}(z^{(l)}_{(\dot{\tau},\ddot{\tau})}, v^{(l)}_{\ddot{\tau}}|v^{(l)}_{\dot{\tau}}, y^{(l)}_{\dot{\tau}}, \theta), \tag{6.128}$$

$$\bar{Q}_\Lambda(\lambda^\dagger) = \log \pi(\lambda^\dagger) + \hat{l}^{-1} \sum_{l=1}^{\hat{l}} \log \pi(y^{(l)}|\lambda^\dagger). \tag{6.129}$$

6.6.3 M-Step

Having computed the weights in the E-step, we proceed to maximize the estimated lower bound functions by solving the optimization problems

$$\left(\operatorname*{argmax}_{\theta^\dagger} \bar{Q}_\Theta(\theta^\dagger), \quad \operatorname*{argmax}_{\lambda^\dagger} \bar{Q}_\Lambda(\lambda^\dagger)\right). \tag{6.130}$$

Analogously to the MCMC parameter update, for models in the class

$$\mathrm{d}V_t = \mu_{\theta_{Y_t}}(V_t)\,\mathrm{d}t + \sigma(V_t)\rho_{\theta_{Y_t}}\,\mathrm{d}W_t, \quad \theta = (\theta_1, \dots, \theta_{\hat{k}}), \tag{6.131}$$

with product prior $\pi(\theta) = \prod_{k=1}^{\hat{k}} \pi(\theta_k)$ the M-step decomposes into independent M-steps for each θ_k:

$$\operatorname*{argmax}_{\theta^\dagger} \bar{Q}_\Theta(\theta^\dagger) = \left\{ \operatorname*{argmax}_{\theta_k^\dagger} \bar{Q}_{\Theta_k}(\theta_k^\dagger) \right\}_{k=1}^{\hat{k}}, \tag{6.132}$$

$$\bar{Q}_{\Theta_k}(\theta_k^\dagger) = \log \pi(\theta_k^\dagger) + \hat{l}^{-1} \sum_{l=1}^{\hat{l}} \sum_{(\dot{\tau}\sim\ddot{\tau})\in\tau^{(l)}: y_{\dot{\tau}}^{(l)}=k} \bar{\ell}_{u_{\dot{\tau},\ddot{\tau}}^{(l)}}(z_{(\dot{\tau},\ddot{\tau})}^{(l)}, v_{\ddot{\tau}}^{(l)} | v_{\dot{\tau}}^{(l)}, y_{\dot{\tau}}^{(l)}, \theta_k). \tag{6.133}$$

If we assume that $\pi(\theta)$, μ_θ, σ_θ and ρ_θ are continuous in θ, as is usually the case in applications, $\bar{Q}_\Theta(\theta^\dagger)$ is also continuous and it is easily optimized with the help of a numerical optimization routine, e.g. BFGS. Conversely, we may optimize $\bar{Q}_\Lambda(\lambda^\dagger)$ exactly for a range of priors. Recall that the complete data likelihood of the realization $y^{(l)}$ may be expressed in terms of the jump counts n_{ij} from state i to j and the cumulative state holding times χ_i. As above, we assume that $\lambda_{ij} \sim \text{Gamma}(\alpha, \beta)$ a priori. The corresponding Q-estimator is

$$\bar{Q}_\Lambda(\lambda^\dagger) = \sum_{i\neq j} \left(\hat{l}^{-1} \sum_{l=1}^{\hat{l}} (n_{ij}^{(l)} \log \lambda_{ij}^\dagger + \chi_i^{(l)} \lambda_i^\dagger) + (\alpha - 1) \log \lambda_{ij}^\dagger - \beta \lambda_{ij}^\dagger \right). \tag{6.134}$$

Taking derivatives results in independent FOCs and yields the optimal value:

$$\lambda_{ij}^\dagger = \begin{cases} \dfrac{\alpha - 1 + \hat{l}^{-1} \sum_{l=1}^{\hat{l}} n_{ij}^{(l)}}{\beta + \hat{l}^{-1} \sum_{l=1}^{\hat{l}} \chi_i^{(l)}} & \text{if } \alpha - 1 + \hat{l}^{-1} \displaystyle\sum_{l=1}^{\hat{l}} n_{ij}^{(l)} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{6.135}$$

6.6.4 Standard Error Estimation

Derivation of standard error estimators follows the same trajectory as in Section 5.5.4, and we do not provide a boilerplate variation here. It is worth pointing out that the conditions guaranteeing asymptotic normality of the MLE are necessarily more complicated, so estimates need to be treated with more caution as to their theoretical validity. Within the framework of Section 5.5.4, it is easiest to provide separate estimates of $\text{Cov}\left[\theta^\ddagger\right]$ and $\text{Cov}\left[\lambda^\ddagger\right]$ separately, though the joint sampling covariance could also be estimated if needed, for a loss of elegance in notation and implementation.

6.6.5 Avoiding Absorbing States

The solution to the M-step reveals a limitation of the algorithm: If $\lambda_{ij}^\dagger = 0$, then the E-step will generate 0 transitions from i to j, permanently fixing $\lambda_{ij}^\dagger$ at 0. Such absorbing states may be avoided by choosing a value $\alpha > 1$, but this excludes the pure maximum likelihood case which corresponds to the hyperparameters $\alpha = 1$, $\beta = 0$.

If pure ML estimation is required, the absorbing states can be avoided by using a *stable* algorithm which resets the generator to a safe value when the M-step enters a forbidden, progressively vanishing set. This method was proposed by [42] for the purpose of analyzing the convergence properties of MCEM algorithms, but it also represents a convenient framework for avoiding absorbing states. By shrinking the forbidden set successively, we ensure that the MAP value is not excluded.

One such heuristic consists of checking whether $\lambda_{ij}^\dagger = 0$ after the M-step. If so, reset $\lambda_{ij}^\dagger$ to the largest value in λ. Depending on the specific setting, θ may have to be constrained as well. In the instance corresponding to (6.96), if $\lambda_{ij}^\dagger = 0$ for all i, then $\theta_j^\dagger$ reverts to its prior mode. For this value, the E-step may be very unlikely to generate transitions to state j. It is therefore preferable to reset $\theta_j^\dagger$ to $\theta_i^\dagger$ for some $i \neq j$, e.g. the state i which has the largest stationary probability under $\lambda^\dagger$.

6.7 Simulation Study

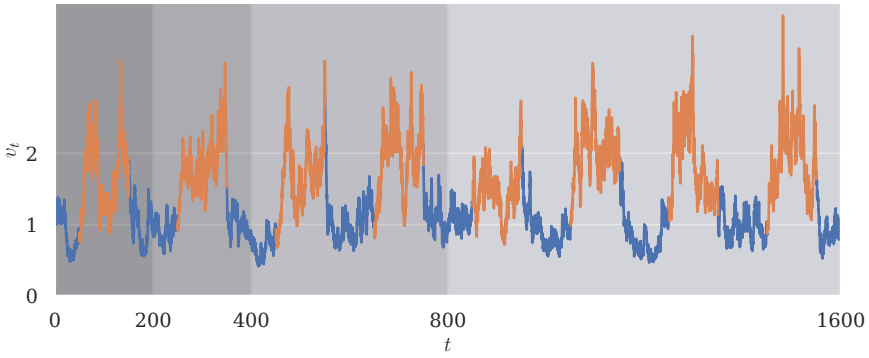


Figure 6.6: Input time series for the extension regime, generated according to the switching logistic growth model with parameters $(\beta_0, \kappa_0, \rho_0) = (1, 1/2, 1/8)$ and $(\beta_1, \kappa_1, \rho_1) = (1, 1, 1/8)$. The blue line corresponds to the trajectory of V when in state 1, and the orange to state 0. The darkest region corresponds to the smallest input series, with lighter regions being appended successively to obtain the larger input series.

In this section, we explore the scaling behavior of our methods in the same regimes we investigated in Section 6.7, i.e. the extension regime, where we append further data

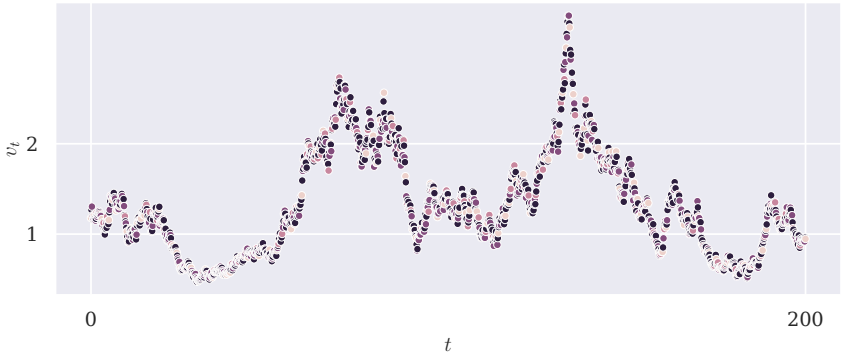


Figure 6.7: Input time series for the infill regime, generated according to the switching logistic growth model with parameters $(\beta_0, \kappa_0, \rho_0) = (1, 1/2, 1/8)$ and $(\beta_1, \kappa_1, \rho_1) = (1, 1, 1/8)$. The lightest dots correspond to the slowest observation frequency, with darker dots filled in to obtain the higher observation frequencies.

to the time series, and the infill regime, where we increase observation frequency. The input data simulation protocol uses a deterministic trajectory of y which switches states every 100 time units, ensuring that as data is added, it is taken in equal parts during the activity of each regime. Figure 6.6 illustrates the deterministic regime switching pattern of the input data. We then investigate the efficiency of the marginal and the auxiliary algorithm under both regimes.

We adopt a regime-switching version of the *logistic growth model*, defined by the SDE

$$\mathrm{d}V_t = \rho_{Y_t} V_t(\beta_{Y_t}(1 - V_t/\kappa_{Y_t})\,\mathrm{d}t + \mathrm{d}W_t), \qquad (\rho_i, \beta_i, \kappa_i > 0, \quad i = 0, 1) \tag{6.136}$$

where ρ is a scale parameter, β is the reproduction rate and κ^{-1} is the *carrying capacity* of the environment. We also set symmetrical priors for all states:

$$\log \beta_i, \log \kappa_i, \log \rho_i \sim \mathrm{N}\,[0, 1]\,, \quad \lambda_{ij} \sim \mathrm{Exp}\,[1/32]\,, \qquad (i, j = 0, 1). \tag{6.137}$$

In the instance where Y is ill-identified a posteriori, such a prior can result in substantial posterior mass allocated to the event where Y remains in a single state. This is difficult to interpret from an inference perspective, since conditional on state i being dropped, the Θ_i follows the prior, therefore posterior inference can be largely prior-driven. Besides, computational issues arise when the algorithm starts random walking on the prior of Θ_i. If Θ_i walks off to the tails to the prior, the update to (Z, Y) can become prohibitively difficult, as such values may imply unreasonable and misspecified diffusion dynamics with very large and variable drift. We choose to avoid this scenario by truncating the prior, and therefore the posterior, to trajectories of Y where both states are present. This is enforced by rejecting single-state updates to Y. This is not a perfect fix, since the algorithm can still move to trajectories of Y where very little time is spent in state i, whereupon Θ_i may largely follow the prior.

We implement the model using the following code snippet:

```
v = sp.symbols('v', positive=True)
x = sp.symbols('x', real=True)
b, k, r = sp.symbols('b k r', positive=True)
thi = sp.Array([b, k, r])
mu = b * r * v * (1 - k * v)
rho = r
sig = v
```

No further code or analysis is necessary for implementation.

We apply the same efficiency notion of average time per effective sample (T/ES) introduced in Section 2.4 and (2.45). We decompose T/ES into the average time per iteration (T/I) and the average number of iterations per effective sample (I/ES), and estimate them from the output of the MCMC algorithm. The task of assessing the effective sampling cost is further complicated by the fact that iteration times are random, so the cost per iteration must be estimated as well. As in the Itō diffusion setting, iteration costs have deterministic and random components, the former of which is linear in the number of observations, while the latter depends on the degree of parameter and regime uncertainty, among other things. Our working hypothesis is that scaling should be similar: For the extension regime, the optimistic scenario is that random costs and therefore T/I remain linear in expectation, while I/ES remains constant, giving linear T/ES. For the infill regime, we expect decreasing random costs and sublinear T/I with constant I/ES and therefore sublinear T/ES. Other than for θ and λ, we track these performance metrics for various other statistics. Similarly to the Itō diffusion case given by (5.98), we evaluate the density

$$\begin{aligned}&\pi(v_{s\setminus\{0\}}, \breve{z}^{(k)}, \breve{\psi}^{(k)}, y^{(k)}, \theta^{(k)}, \lambda^{(k)}|v_0)\\ &\quad = \pi(\theta^{(k)})\pi(\lambda^{(k)})\pi(y^{(k)}|\lambda^{(k)}) \prod_{(\dot\tau\sim\ddot\tau)\in s} \pi(v_{\ddot\tau}, \breve{z}^{(k)}_{(\dot\tau,\ddot\tau)}, \breve{\psi}^{(k)}_{(\dot\tau,\ddot\tau)}|v_{\dot\tau}, y^{(k)}_{(\dot\tau,\ddot\tau]}, \theta^{(k)}),\end{aligned} \tag{6.138}$$

where $(y^{(k)}, \theta^{(k)}, \lambda^{(k)})$ is the value at MCMC iteration k, and $\breve{z}^{(k)}_{(\dot\tau,\ddot\tau)}$ and $\breve{\psi}^{(k)}_{(\dot\tau,\ddot\tau)}$ are random samples from $\mathbb{B}_{(\dot\tau,\ddot\tau)}$ and $\mathbb{P}_{(\dot\tau,\ddot\tau)}$, rather than being taken from the MCMC chain. Therefore,

$$\mathrm{E}_{\mathbb{B}_{(\dot\tau,\ddot\tau)}\times\mathbb{P}_{(\dot\tau,\ddot\tau)}}\left[\pi(v_{s\setminus\{0\}}, \breve{Z}, \breve{\Psi}, y^{(k)}, \theta^{(k)}, \lambda^{(k)}|v_0)\right] = \pi(v_{s\setminus\{0\}}, y^{(k)}, \theta^{(k)}, \lambda^{(k)}|v_0) \tag{6.139}$$

In addition, we also track the proportion of time that Y spends in state 0, i.e.

$$\omega^{-1}\int_0^\omega 1[y_t^{(k)} = 0]\,\mathrm{d}t, \tag{6.140}$$

which can be useful to quickly detect meta-stable states, and the jump rate per time unit, i.e. $\omega^{-1}\left|r^{(k)}\right|$.

Each MCMC run that contributes to this section's results consists of 100000 iterations, of which we discard 10000 for burn-in. We precede the exact MCMC run by an approximate run of 10000 iterations according to the algorithm of Section 5.4. We target an acceptance probability of 23.4% for Metropolis-within-Gibbs steps, and 25% for Barker-within-Gibbs, with a Portkey probability of 1%. Step sizes are adapted according to the *Adapting Increasingly Rarely* (AIR) method of [25]. Notice that we do not enforce the identifiability constraint discussed above in the approximate run, since dropping state is in principle helpful to mixing, especially when initiating far from the posterior mode. For both algorithms, we use the conditional hidden data update of Section 6.3.4, setting the additional tuning parameter p_s to 1/10.

For Poisson coin simulations within the marginal algorithm we adopt the limiting batch EA version of Section 4.1.3. When the integrand bounds in Poisson estimator simulations within the auxiliary algorithm exceed 10000, the proposal is rejected to avoid memory errors. Such events occur a few times in the auxiliary simulations, and while they represent a small departure from exactness, proposals implying such large bounds would usually result in rejection even if the full simulation were to be carried out. Nevertheless, in this instance, the marginal version is fully exact as the batch EA method avoids carrying out excessively expensive simulations without loss of exactness.

6.7.1 Extension Regime

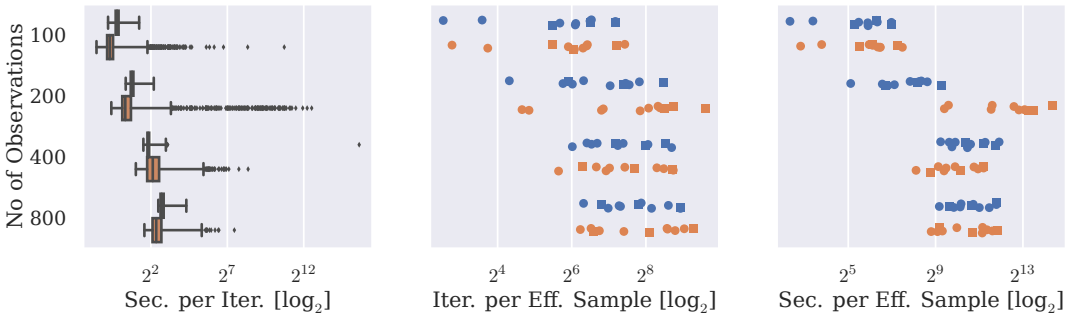


Figure 6.8: Sampling efficiency in the infill regime for the auxiliary (blue) and marginal (orange) algorithms. The left panel shows the distribution of CPU time per iteration (T/I) throughout the MCMC run. The middle panel shows estimates of the required number of MCMC iterations to generate an effective sample (I/ES), where each dot corresponds to an element of Θ and Λ, and the squares to the miscellaneous posterior summaries defined in (6.138) and following. The right panel shows estimates of the required CPU time to generate an effective sample (T/ES). Notice that the right panel is obtained by scaling the middle panel by the mean of the distributions in the left panel.

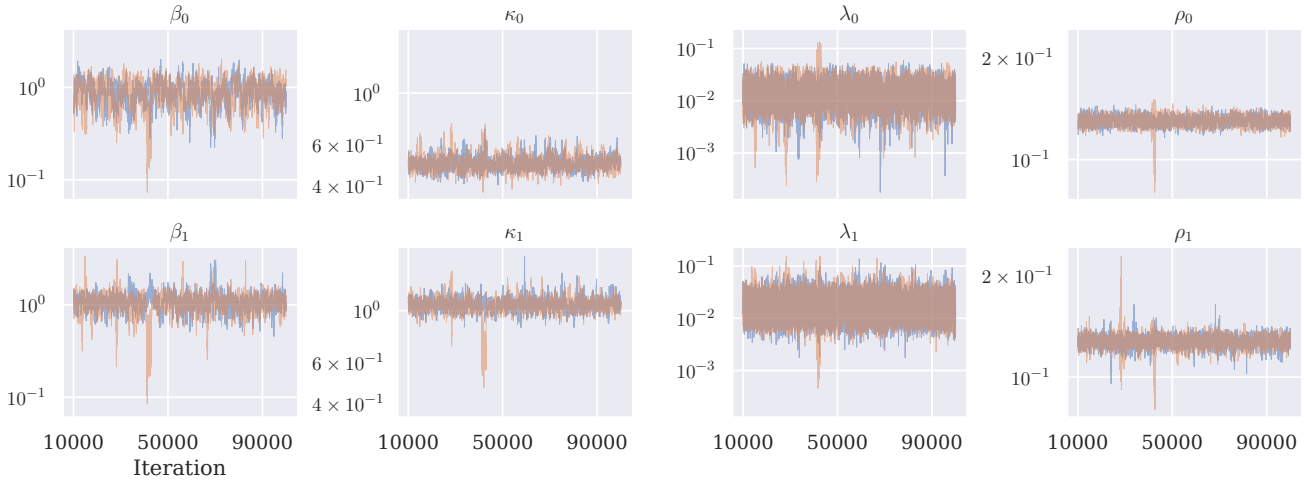


Figure 6.9: Trace plots of Θ and Λ for the 800-observation time series in the extension regime for the auxiliary (blue) and marginal (orange) algorithms. We plot the y-axis on the log scale due to the heavy tails of the posterior.

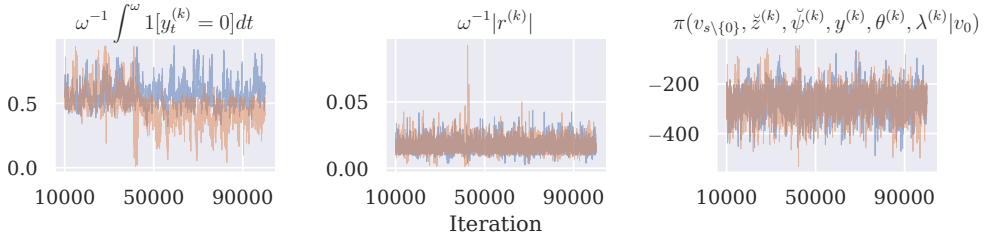


Figure 6.10: Trace plot of various posterior summaries for the 800-observation time series in the extension regime for the auxiliary (blue) and marginal (orange) algorithms.

For the extension regime, we run the marginal and augmented algorithms on a time series of 200, 400, 800 and 1600 observations respectively, with an inter-observation interval of 1. The series is generated from the switching logistic growth model with parameters $(\beta_0, \kappa_0, \rho_0) = (1, 1/2, 1/8)$ and $(\beta_1, \kappa_1, \rho_1) = (1, 1, 1/8)$ and plotted in Figure 6.6. As apparent in Figure 6.9, the model identifies a "high carrying capacity state" (small κ) and a "low carrying capacity state". We assign the label 0 to the former and 1 to the latter.

We observe very similar performance between the marginal and the auxiliary algorithms. As seen in Figure 6.8, T/I is roughly linear and I/ES roughly constant at the upper end of the observation scale, resulting in roughly linear T/ES. We again notice similar mean T/I, but substantially larger variance for the marginal algorithm, with a pronounced right tail. In particular, when Y is close to being entirely assigned to one state, the

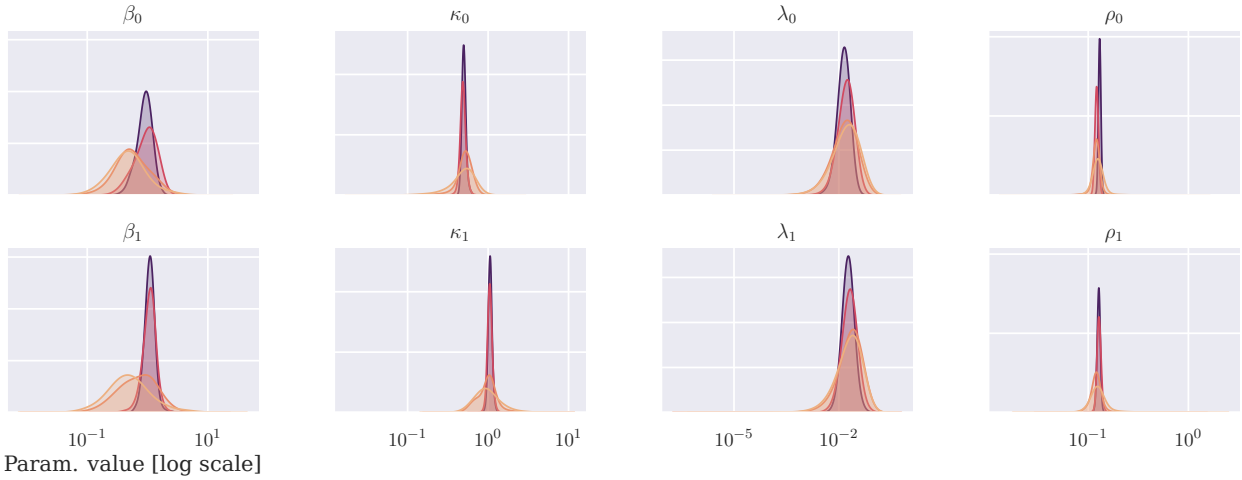


Figure 6.11: Posterior marginals of Θ and Λ in the extension regime, as estimated by a KDE. Darker shades correspond to a larger observation number. We plot the y-axis on the log scale due to the heavy tails of the posterior.

difference between Θ_0 and Θ_1 may be very large, causing very long iteration times in the marginal algorithm. This manifests itself in excursions into the tails, clearly visible in the trace plots of Figure 6.9, corresponding to the largest data scenario with the most posterior concentration. In fact, for all elements of Θ, we observe concentration of the posterior around the true simulation values, see Figure 6.11.

6.7.2 Infill Regime

For the infill regime, we interpolate the first 200 observations used in the extension experiment at frequencies 2, 4 and 8, with identical parameters $(\beta_0, \kappa_0, \rho_0) = (1, 1/2, 1/8)$ and $(\beta_1, \kappa_1, \rho_1) = (1, 1, 1/8)$. The resulting observations are plotted in Figure 6.7. Compared to the extension regimes, the regimes are not identified as clearly. In Figure 6.13 we observe a separation between a small ρ and a large ρ state for much of the run, but the model (correctly) does not rule out $\rho_0 \approx \rho_1$. It fails to clearly distinguish a small and a large carrying capacity regime. Therefore, we do not relabel regimes.

We reproduce the observations of the Itō setting in finding a factor 2 improvement of the marginal over the auxiliary algorithm in the higher observation frequency range, see Figure 6.12. The auxiliary algorithm appears linear in T/I and increasing in I/ES, resulting in superlinear T/ES. The marginal algorithm has slightly sublinear T/I and increasing I/ES, with a roughly linear T/ES overall. The increasing I/ES (i.e. degradation of mixing) in both instances stands in contrast to our observations in the Itō setting.

As seen in Figure 6.13, labels switch frequently throughout the run, and MCMC inference reflects the label switching invariance of the model, providing near identical inference

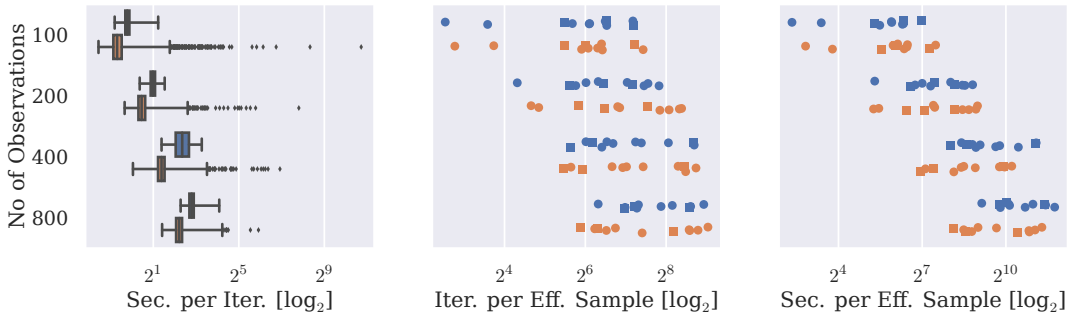


Figure 6.12: Sampling efficiency in the infill regime for the auxiliary (blue) and marginal (orange) algorithms. The left panel shows the distribution of CPU time per iteration (T/I) throughout the MCMC run. The middle panel shows estimates of the required number of MCMC iterations to generate an effective sample (I/ES), where each dot corresponds to an element of Θ and Λ, and the squares to the miscellaneous posterior summaries defined in (6.138) and following. The right panel shows estimates of the required CPU time to generate an effective sample (T/ES). Notice that the right panel is obtained by scaling the middle panel by the mean of the distributions in the left panel.

for Θ_0 and Θ_1. We observe little posterior concentration in the parameters, with the exception of the scale parameters ρ_i, see Figure 6.15.

6.8 Demonstration: Weak Mean Reversion for T-Bill Spreads

We proceed with a demonstration of the marginal algorithm on a model for US T-Bill spread, with the time series shown in Figure 6.16. To reliably sample from the corresponding posterior, we need to leverage the full extent of the methodology developed in this chapter. The data consists of weekly observations of the spread between 1-month and 3-month bills over a 2-decade time span. We model it as a 3-state Markov switching diffusion with subexponential mean reversion:

$$\mathrm{d}V_t = \rho_{Y_t}(\beta_{Y_t} \tanh\left[\mu_{Y_t} - V_t\right] \mathrm{d}t + \mathrm{d}W_t). \qquad (\beta_i, \rho_i > 0, \quad i = 1, 2, 3) \qquad (6.141)$$

Accordingly, the drift function is bounded in absolute value by $\rho_{Y_t}\beta_{Y_t}$ and by the corresponding Vasicek process drift function $\rho_{Y_t}\beta_{Y_t}(\mu_{Y_t} - V_t)$. The transition density for this model is intractable, it therefore falls within the scope of our method. We use symmetrical priors for all states:

$$\mu_i, \log\beta_i, \log\rho_i \sim \mathrm{N}\left[0, 1\right], \quad \lambda_{ij} \sim \mathrm{Exp}\left[1/730\right]. \qquad (6.142)$$

This implies a prior expectation of one state transition per year. The specification results in a posterior that is invariant to label permutations and therefore multimodal. When

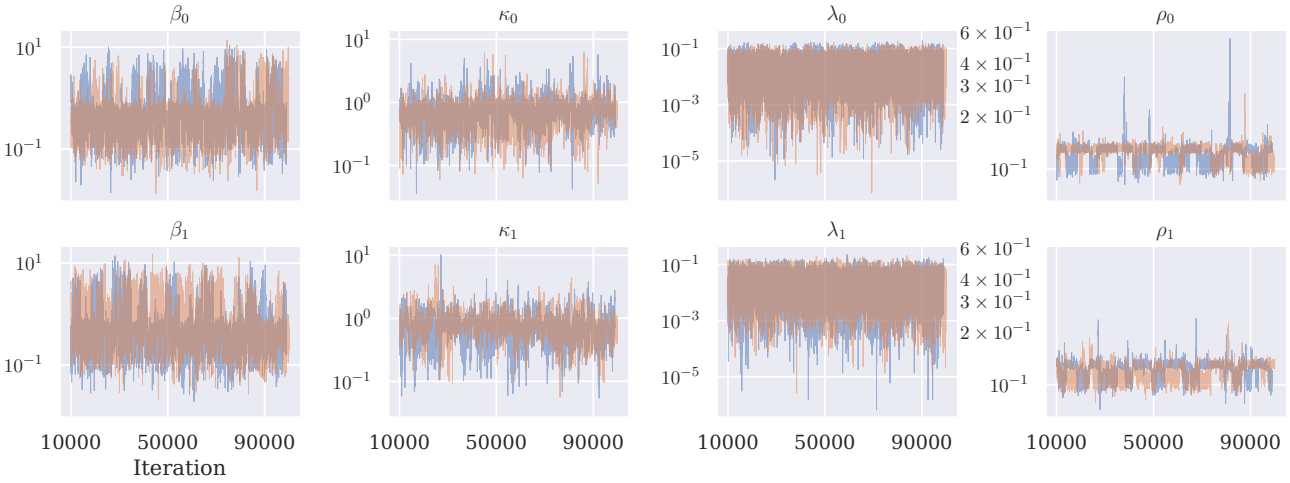


Figure 6.13: Trace plots of Θ and Λ for the 800-observation time series in the infill regime for the auxiliary (blue) and marginal (orange) algorithms. We plot the y-axis on the log scale due to the heavy tails of the posterior.

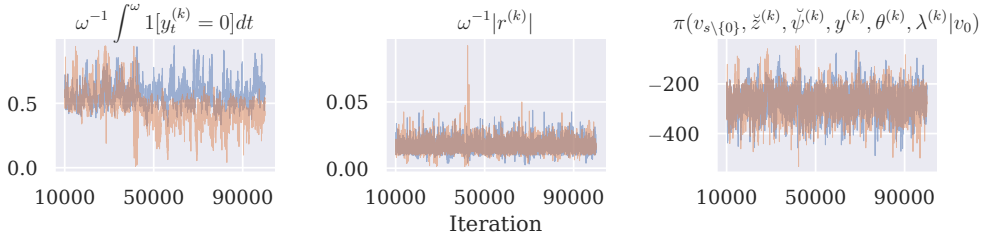


Figure 6.14: Trace plot of various posterior summaries for the 800-observation time series in the infill regime for the auxiliary (blue) and marginal (orange) algorithms.

the posterior is concentrated, the algorithm will only be able to visit one of those modes, and in this instance we do not face the identification issues pointed out in the previous section, as the algorithm never drops states after warmup.

We merely require the following code snippet to implement the model:

```
v, x = sp.symbols('v x', real=True)
b, r = sp.symbols('b r', positive=True)
m = sp.symbols('m', real=True)
thi = sp.Array([m, b, r])
mu = r * b * sp.tanh(m - v)
sig = sp.Integer(1)
rho = r
```

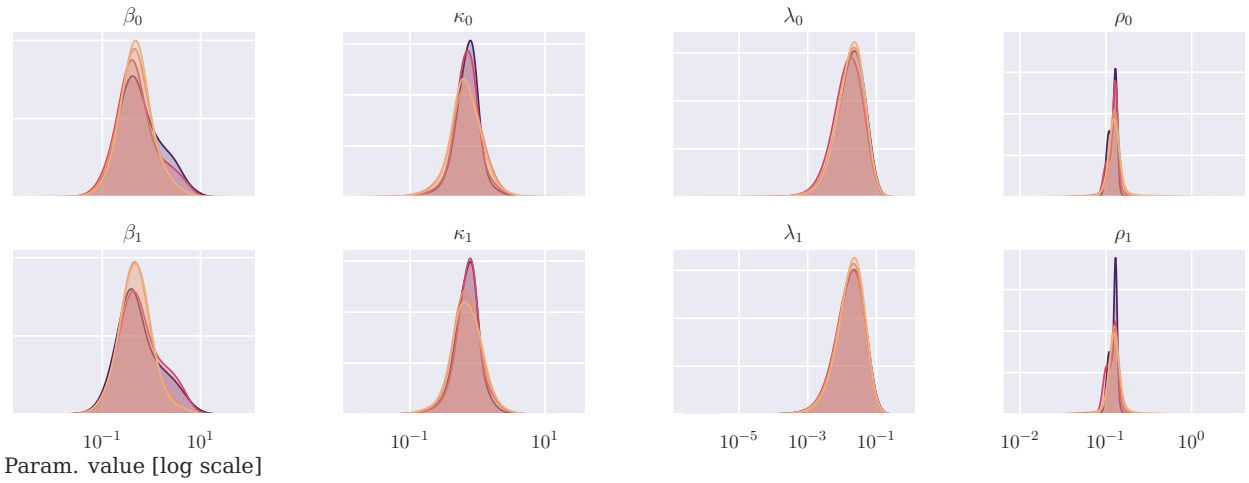


Figure 6.15: Posterior marginals of Θ and Λ in the infill regime, as estimated by a KDE. Darker shades correspond to a larger observation number. We plot the y-axis on the log scale due to the heavy tails of the posterior.

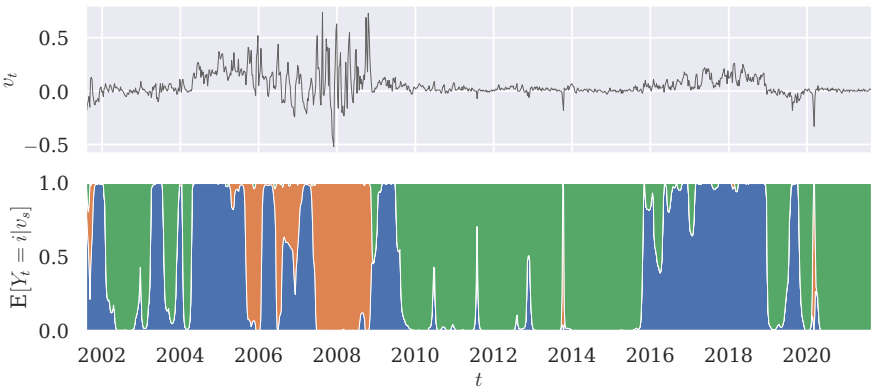


Figure 6.16: (Top) Time series of T-Bill spreads. (Bottom) Stacked posterior regime probabilities $\Pr[Y_t = i|v_s]$, as inferred by the MCMC algorithm.

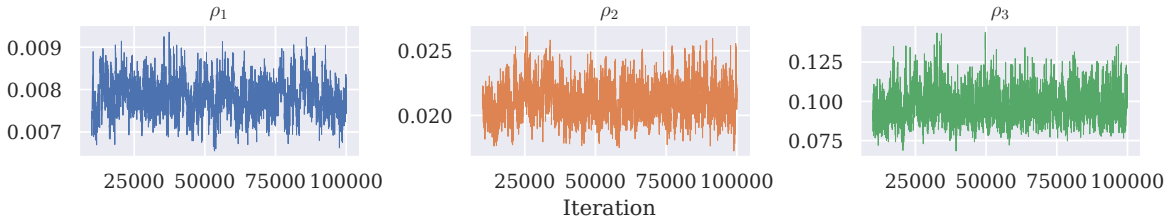


Figure 6.17: Trace plots of ρ_i for the exact MCMC algorithm. These are the parameters that mix most slowly.

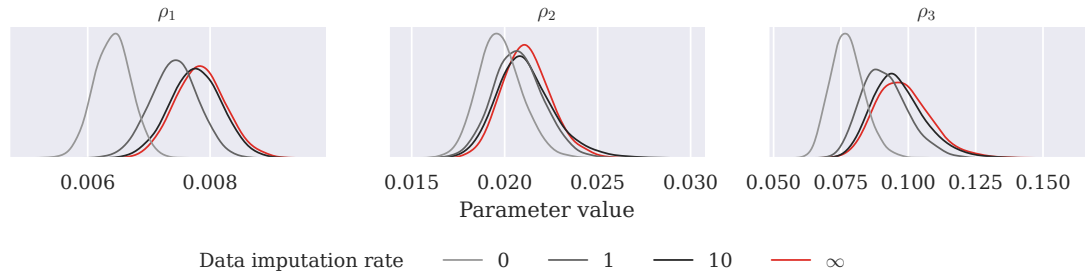


Figure 6.18: Comparison of the posterior marginals of ρ_i for the exact MCMC algorithm and the approximate algorithm with various rates of data imputation per day. These are the parameters for which the approximate algorithm exhibits the largest bias.

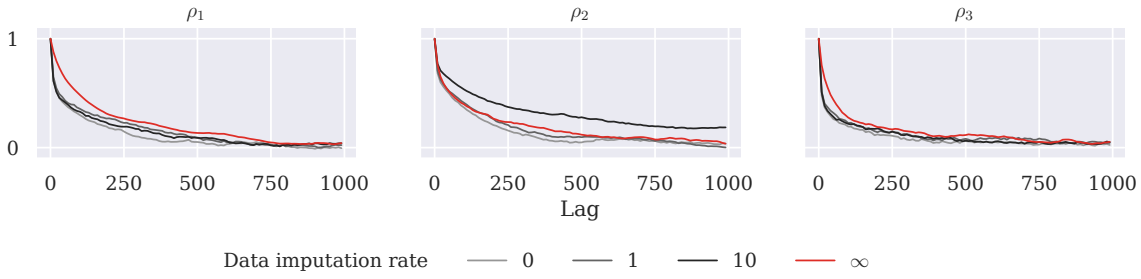


Figure 6.19: Comparison of autocorrelation functions of ρ_i for the exact MCMC algorithm and the approximate algorithm with various rates of data imputation per day. These are the parameters that mix most slowly.

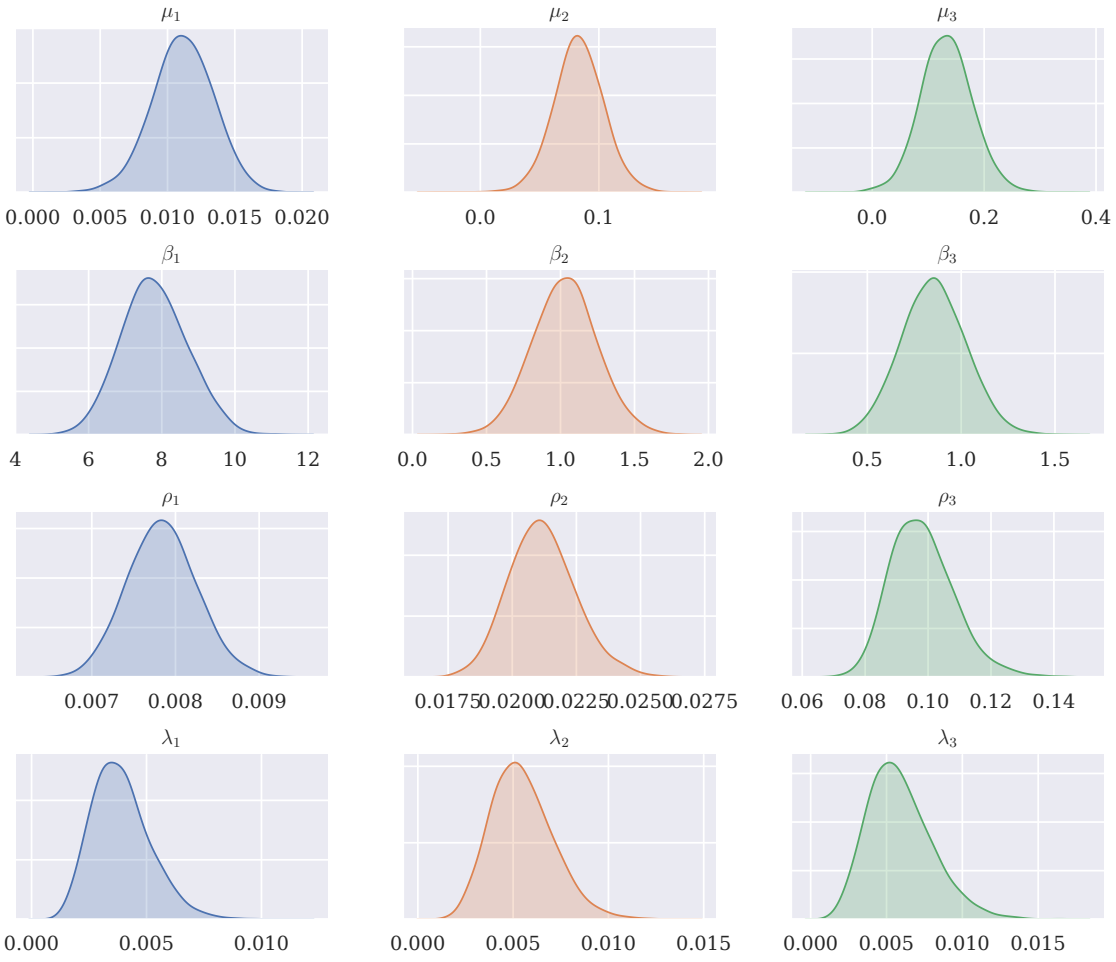


Figure 6.20: Posterior marginals of the elements of θ for the exact MCMC algorithm.

Each MCMC run that contributes to this section's results consists of 100000 iterations, of which we discard 10000 for burn-in. We precede the exact MCMC run by an approximate run of 10000 iterations according to the algorithm of Section 5.4. We target an acceptance probability of 23.4% for Metropolis-within-Gibbs steps, and 25% for Barker-within-Gibbs, with a Portkey probability of 1%. Step sizes are adapted according to the *Adapting Increasingly Rarely* (AIR) method of [25]. For Poisson coin simulations we adopt the limiting batch EA version of Section 4.1.3. We use the conditional hidden data update of Section 6.3.4, setting the additional tuning parameter p_s to 1/10.

To validate the output and quantify the benefits of the exact method, we compare the output of the exact algorithm to runs of the approximate method developed in Section 6.5. We run this algorithm with various discretization intensities and characterize the rate at which the bias decreases. Additionally, the approximate algorithm allows us to start the exact algorithm not too far from the stationary distribution. Thereby, we avoid a long transient phase for the exact algorithm, where iteration times are typically the longest and the most variable.

We observe decreasing, but substantial bias of the approximate methods for all the attempted discretization rates, as seen in Figure 6.18. Moreover, as shown in Figure 6.19, mixing starts to degrade at the higher rates, at which the average rate of diffusion imputation is much larger than for the exact algorithm.

6.9 Discussion

As in the previous chapter, the simulation results do not allow for strong recommendations for either the marginal or the auxiliary marginal approach. We again see reliable scaling in both extension and infill regimes, with no departure from exactness for the marginal algorithm, and minimal departure for the auxiliary one. The exact algorithm is also capable of generating adequate efficient sample sizes on a realistic data set with substantial drift discontinuities.

There are some important departures from previous conclusions. Both in instances where regimes are very weakly or very strongly identified, the posterior is likely to support very different values of θ_i for the different regimes. In such a situation, any proposal $y^\dagger$ is akin to a large move in parameter space, which tends to result in long and irregular iteration times, regardless of the acceptance probability of the proposal. Even in the intermediate and favorable scenario of Section 6.7, the distribution of iteration times can span many orders of magnitude, as seen in Figures 6.8 and 6.12.

Conversely, we find that approximate algorithms exhibit substantial bias even at large imputation frequencies, where we also observe degradation in MCMC mixing. At such frequencies, the runtime of exact and approximate algorithms is comparable, indicating that exact algorithms should be preferred when parameter inference is the primary goal of the modelling exercise. Nonetheless, we believe that the Euler-discretized model

could be leveraged to make more targeted proposals, especially in the Y-updates, e.g. by doing filtering-smoothing according to the discretized model along the lines of [103]. Thereby, we may be able to mostly avoid proposals with low acceptance probability, but large iteration time. This synthesis of approximate and exact approaches is already supported by the substantial observed benefits of warm-starting.

7 Approximate Inference for Stochastic Volatility Diffusions

Chapter 6 has investigated Markov Switching diffusions with dynamics that are modified by discrete, exogenous shocks. Alternatively, we can introduce continuous modifications of the dynamics by modelling the latent process as a diffusion itself. We may set up a system of SDEs of form

$$\mathrm{d}V_t = \mu_\theta(V_t)\,\mathrm{d}t + \sigma_\theta(V_t)\rho(U_t)\,\mathrm{d}W_t^V, \tag{7.1}$$

$$\mathrm{d}U_t = \beta_\xi(U_t)\,\mathrm{d}t + \gamma_\xi(U_t)\,\mathrm{d}W_t^U. \tag{7.2}$$

where V is the observed diffusion on $\mathcal{V}$, U the latent one on $\mathcal{U}$, and θ and ξ are parameter vectors affecting the observed and latent diffusion, respectively. The link function $\rho : \mathcal{U} \mapsto [0, \infty)$ with inverse ρ^{-1} ensures that the latent process is mapped to a nonnegative volatility. We may further allow for dependence between the driving Brownian motions by setting $\mathrm{Cov}\left[\mathrm{d}W_t^V, \mathrm{d}W_t^U\right] = \varrho\,\mathrm{d}t$. This type of system is known in the literature as a *(continuous) stochastic volatility* system.

Sufficient conditions for existence of a solution to such multivariate SDE systems, given e.g. by [77], are analogous to the univariate case in that the drift and volatility coefficients must satisfy Lipschitz-like continuity properties. Since we will adopt an approximation to the system, the specifics are of no further consequence - indeed, it will be sufficient for V to satisfy existence conditions with U_t being confined to a finite and bounded set.

Continuous stochastic volatility models were originally proposed in order to improve on the Black-Scholes option pricing formula [110, 66, 125]. [60] developed one of the canonical specifications, known as the *Heston Model*, where the observable process is a Geometric Brownian motion and the diffusivity process corresponds to the Cox-Ingersoll-Ross model:

$$\mathrm{d}V_t = \mu_V V_t\,\mathrm{d}t + \sigma_V V_t\sqrt{U_t}\,\mathrm{d}W_t^V, \tag{7.3}$$

$$\mathrm{d}U_t = \beta_U(\mu_U - U_t)\,\mathrm{d}t + \sigma_U\sqrt{U_t}\,\mathrm{d}W_t^U. \tag{7.4}$$

See Figure 7.1 for a sample trajectory from this model. The specification is of particular interest because the distribution of the integrated diffusivity is known [8]. By the time-change representation of the stochastic integral, there is a Brownian motion W^* such that the SDE can be solved:

$$\log V_t = \mu_V\,\mathrm{d}t + \sigma_V W^*\left(\int_0^t U_t\,\mathrm{d}t\right). \tag{7.5}$$

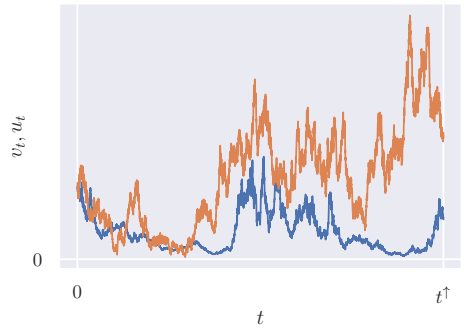


Figure 7.1: Example trajectory from the Heston model with (blue) $\mathrm{d}V_t = V_t\sqrt{U_t}\,\mathrm{d}W_t^V$ and (orange) $\mathrm{d}U_t = (1 - U_t)\,\mathrm{d}t + \sqrt{U_t}\,\mathrm{d}W_t^U$.

Therefore, the Heston model is somewhat tractable, allowing among other things for evaluation of the Moment generating function of V_t, and the pricing of some options. Under other notable specifications, such as the 3/2 model [61, 100] and the SABR model [55], the stochastic volatility model becomes nontractable. These works typically do not consider the question of parameter inference.

7.1 Inference Strategy

In greater generality, exact inference for continuous stochastic volatility models does not directly fit our framework because we cannot transform the observed diffusion into a constant volatility process. Standard practice consists of discretizing both observed and latent process, see e.g. [75] and [67, 72]. Other works such as [105] adopt specific volatility specifications driven by non-Wiener processes. As it happens, a milder form of approximation puts the diffusive volatility model squarely into the realm of Chapter 6. It consists of replacing the latent diffusion U with a continuous-state Markov jump process Y on a relatively dense state space $\mathcal{G} \subset \mathcal{U}$. If we adequately structure the generator as a function of ξ, volatility can be mimicked by frequent state changes, and drift by biasing state changes up or down. The approximation to the SDE system remains a fully continuous, Markovian system. Moreover, by increasing the density of $\mathcal{G}$, we recover the diffusion system, in a sense made precise by the theory of *local consistency*, developed by [78] and briefly presented in this chapter in Section 7.2. Figure 7.2 shows how as $\mathcal{G}$ becomes very dense, the effect on the dynamics of V is eventually negligible. In fact, the setup in this chapter straddles the spectrum between the fully discrete Markov switching of Chapter 6, and the fully continuous exogenous variation of the diffusion system. Markov jump process approximations to stochastic volatility systems have previously been proposed by [26, 28], though those works do not consider the matter of inference.

Before investigating the specifics of the approximation, we must fix an appropriate model class for which we may hope for effective inference methods. This becomes apparent when orthogonalizing the diffusion system. We define

$$\eta_\theta(a) = \int_{v^*}^{a} \frac{1}{\sigma_\theta(c)}\,\mathrm{d}c, \qquad (a, v^* \in \mathcal{V}) \tag{7.6}$$

$$\upsilon_\xi(b) = \int_{u^*}^{b} \frac{\rho(c)}{\gamma_\xi(c)}\,\mathrm{d}c, \qquad (b, u^* \in \mathcal{U}) \tag{7.7}$$

and the auxiliary process $X = \eta_\theta(V) - \varrho\upsilon_\xi(U)$ to obtain the orthogonal system

$$\mathrm{d}X_t = \delta_{\theta,\xi}(X_t, U_t)\,\mathrm{d}t + \rho(U_t)\sqrt{1-\varrho^2}\,\mathrm{d}W_t^X, \tag{7.8}$$

$$\delta_{\theta,\xi}(a,b) = \left(\frac{\mu_\theta}{\sigma_\theta} - \rho^2(b)\frac{\sigma_\theta'}{2}\right) \circ (\eta_\theta)^{-1}(a + \varrho\upsilon_\xi(b)) + \varrho(\beta_\xi\upsilon_\xi' + \gamma_\xi\upsilon_\xi''/2)(b), \tag{7.9}$$

for which $W_t^X \perp W_t^U$. Therefore, the system (X, U) is the natural system on which to approximate U, rather than (V, U). The presence of both θ and ξ in the specification of the auxiliary process X results in a rather dense conditional independence graph, complicating the design of good Gibbs blocking schemes and probably necessitating a joint update. While this is by no means infeasible if the parameter dimensionality is moderate, in this chapter we limit ourselves to investigating the no-correlation case $\varrho = 0$, where

$$\mathrm{d}X_t = \delta_\theta(X_t, U_t)\,\mathrm{d}t + \rho(U_t)\,\mathrm{d}W_t^X, \tag{7.10}$$

$$\delta_\theta(a,b) = \left(\frac{\mu_\theta}{\sigma_\theta} - \rho^2(b)\frac{\sigma_\theta'}{2}\right) \circ (\eta_\theta)^{-1}(a), \qquad (a \in \mathcal{X}, \quad b \in \mathcal{U}) \tag{7.11}$$

and changes in ξ do not affect X. This allows for conditionally independent updates to Θ and Ξ.

The rest of the chapter proceeds as follows. Section 7.2 introduces a diffusion approximation scheme by a continuous-time Markov jump process Y, and states the mode of convergence to the diffusion. Section 7.3 briefly adapts the familiar data augmentation approach from Section 6.1.1. Section 7.5 presents a marginal Gibbs sampler in the style of Section 6.3 that targets

$$\pi(v_r, z, y, \theta, \xi | v_s) \propto \pi(\theta)\pi(\xi) \prod_{(\dot\tau \sim \ddot\tau) \in \tau} \pi(z_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau} | v_{\dot\tau}, y_{\dot\tau}, \theta)\pi(y_{\ddot\tau} | y_{\dot\tau}, \xi), \tag{7.12}$$

where Θ and Ξ are a priori independent, R corresponds to the jump times of Y, $T = R \cup S$, and Z results from noncentering X, as usual. Section 7.6 investigates numerical performance of the algorithm, particularly in the regime where the approximation to U is refined.

We note that there are complications and simplifications compared to the general Markov switching diffusion setting of Chapter 6. On the one hand, Y is an approximation to a

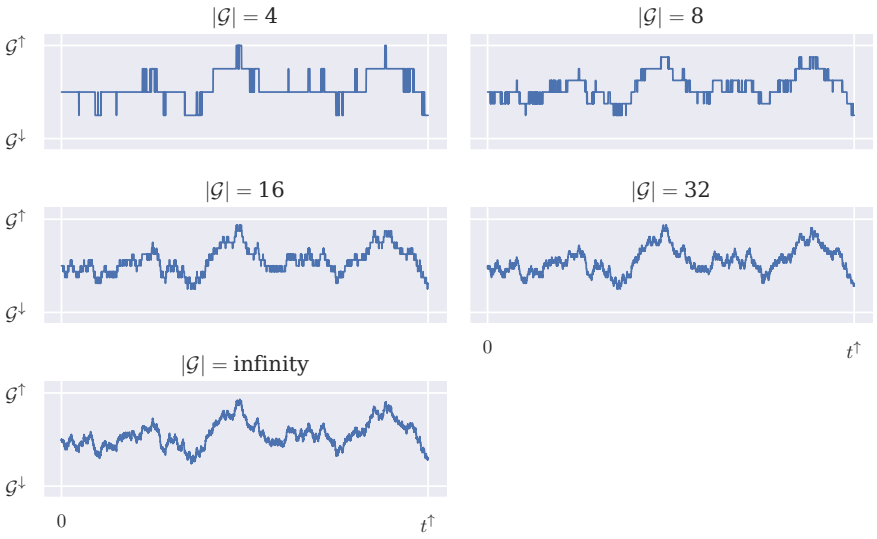


Figure 7.2: Illustration of approximating a trajectory of U ($|\mathcal{G}| = \infty$) by a step function on $\mathcal{G}$ for different resolutions.

process that affects the SDE of V continuously, and it does not affect the drift at all. This substantially alleviates the issue of proposal sensitivity raised e.g. in Section 6.9. On the other hand, posterior dependence between Y and Ξ is potentially large, with no obvious reparameterization available to alleviate the associated slowdown of Gibbs sampling. Other challenges around the simulation of continuous-time Markov jump processes with large state spaces are addressed in Section 7.4

7.2 Latent Diffusion Approximation and Local Consistency

We will carry out inference for an approximate version of the stochastic volatility model, where U is approximated by a piecewise constant process. The key notion to such an approximation is the concept of *local consistency*, developed by [78].

Definition 6 (Local consistency [78])**.** *Let Y be a continuous time Markov jump process on the grid $\mathcal{G} \subset \mathcal{U}$ with generator λ. Suppose that it satisfies* local consistency *with respect to the diffusion process U, i.e. β_ξ and γ_ξ are Lipschitz-continuous for any ξ and*

$$\mathrm{E}\left[Y_{t+\epsilon} - Y_t | Y_t\right] = \beta_\xi(Y_t)\epsilon + \mathcal{O}(\epsilon), \tag{7.13}$$

$$\mathrm{E}\left[(Y_{t+\epsilon} - Y_t)^2 | Y_t\right] = \gamma_\xi(Y_t)\gamma'_\xi(Y_t)\epsilon + \mathcal{O}(\epsilon). \tag{7.14}$$

Then, as $\mathcal{G}$ tends to cover $\mathcal{U}$, Y weakly converges to U.

Figure 7.2 qualitatively illustrates how a finer grid affects the diffusion approximation. The theory of Markov chain approximations for diffusions provides various ways of parameterizing λ in terms of ξ such that local consistency applies. For a grid $\mathcal{G}$ with constant spacing d, local consistency is ensured by setting

$$\lambda_{ij} = \begin{cases} \left(\frac{\gamma_\xi(u_j)}{\sqrt{2}d}\right)^2 + \frac{|0 \wedge \beta_\xi(u_j)|}{d} & j = i-1 \\ \left(\frac{\gamma_\xi(u_j)}{\sqrt{2}d}\right)^2 + \frac{|0 \vee \beta_\xi(u_j)|}{d} & j = i-1 \\ -(\lambda_{i,j-1} + \lambda_{i,j+1}) & i = j \\ 0 & \text{otherwise} \end{cases}. \tag{7.15}$$

It is apparent from the formulation that states where U has large volatility promote moves to nearby states, while drift promotes moves up or down depending on size. A smaller grid spacing also promotes state changes.

Similar, but nonuniformly spaced grids also have the local consistency property. [83] point out how a nonuniform grid results in a better approximation for a given grid size. Furthermore, the algorithm that we will propose is more stable when the volatility cannot change by multiple orders of magnitude for each state change in Y, implying that the volatility spacings $\rho(\mathcal{G})$ should follow a log scale. This is violated for example when U is the CIR process approximated with a uniform grid, resulting in volatility spacings $\sqrt{\mathcal{G}}$ that can vary by many degrees of magnitude, even for a high grid resolution. Nonetheless, nonuniform grids must obey certain relationships that may be fulfilled for some ξ but not for others, which is inconvenient if we wish to infer ξ on an unbounded set. Therefore, we persist with the simple uniform grid. We can still obtain a grid for which $\rho(\mathcal{G})$ follows the log scale by applying Itō's formula to any starting specification such that $\rho(U_t) = e^{cU_t}$.

The remaining, and arguably more decisive degree of freedom is to set the width of the grid. This should ideally be done such that Y only visits the boundaries of $\mathcal{G}$ with moderate probability, since otherwise the volatility process is being constrained unduly, and the model is misspecified. On the other hand, if the probability of visiting the edges of $\mathcal{G}$ is very low, then the span of the grid should be reduced to save computational resources and invest them in a denser grid. For known ξ, we can adopt the framework of [92], who investigate the containment probability of Markov jump processes in order to price barrier options. In particular, defining $\breve{\lambda}$ as the submatrix of λ excluding the first and last rows and columns, $\breve{\mathcal{G}}$ as the interior of $\mathcal{G}$ excluding $\breve{\mathcal{G}}$ and $\hat{\mathcal{G}}$, and ν as the first escape time from $\breve{\mathcal{G}}$ onto the boundary of $\mathcal{G}$, they find that $\breve{\lambda}$ is sufficient to evaluate

$$\mathrm{E}\left[f(Y_\omega)\, 1_{\nu>\omega} \,|y_0, \xi\right] = e(y_0)^\mathsf{T} e^{\omega\breve{\lambda}} f(\breve{\mathcal{G}}), \qquad (y_0 \in \breve{\mathcal{G}}, f : \breve{\mathcal{G}} \to \mathbf{R}) \tag{7.16}$$

where $e(y_0)$ is the vector which is 1 at y_0 and 0 otherwise, such that the RHS is a quadratic form. We point out the special case where f is the identity function, yielding $\Pr[\nu > \omega | y_0, \xi]$. If our prior on ξ is not too diffuse, we can use this formula to make sure that the a priori containment probability $\int \Pr[\nu > \omega | y_0, \xi]\, \pi(\xi)\, \mathrm{d}\xi$ is close to 1.

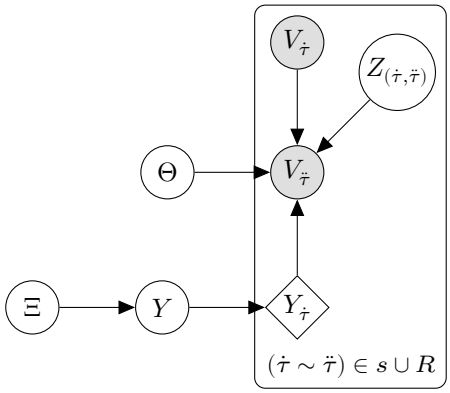


Figure 7.3: Plate diagram for the approximate Stochastic volatility class considered in this chapter. $V_{\dot\tau}$ and $V_{\ddot\tau}$ may be observed or latent, depending on whether $\dot\tau, \ddot\tau \in s$.

Conversely, if our prior is too diffuse to control the containment probability, we resort to a simple heuristic where we use the observations v_s to obtain a rough estimate of the range of Y. We do so by postulating that V is an accelerated Brownian motion following $\rho(U_t)\,\mathrm{d}W_t$, giving the simple volatility estimate of

$$\left|v_{\ddot s} - v_{\dot s}\right| / (\ddot s - \dot s), \tag{7.17}$$

in between observations $v_{\dot s}$ and $v_{\dot s}$. We then cover the $100(1 - 1/|\mathcal{G}|)\%$-interval of the sample

$$\left\{\rho^{-1}(\left|v_{\ddot s} - v_{\dot s}\right| / (\ddot s - \dot s))\right\}_{(\dot s \sim \ddot s) \in s}, \tag{7.18}$$

with an equidistant grid of $|\mathcal{G}|$ points.

7.3 Complete Transition Density

After replacing U with its approximation Y in (7.10), we obtain the model

$$\mathrm{d}X_t = \delta_\theta(X_t, Y_t)\,\mathrm{d}t + \rho(Y_t)\,\mathrm{d}W_t^X, \tag{7.19}$$

$$\delta_\theta(a,b) = \left(\frac{\mu_\theta}{\sigma_\theta} - \rho^2(b)\frac{\sigma_\theta'}{2}\right) \circ (\eta_\theta)^{-1}(a), \qquad (a \in \mathcal{X}, \quad b \in \mathcal{U}) \tag{7.20}$$

and observe that X is now a Markov switching diffusion and a special case of the specification of Section 6.1.1. Hence, we can derive a complete transition density along essentially identical lines. We require for any $\theta \in \mathcal{T}$, $b \in \mathcal{G}$ and $\dot\tau < \ddot\tau$ that

- $\delta_\theta(a,b)$ is continuously differentiable in a.
- The *Novikov condition* applies, i.e. $\mathrm{E}_{X_{(\dot\tau,\ddot\tau]}}\left[\exp\left[\int_{\dot\tau}^{\ddot\tau} \delta_\theta^2(X_t,b)\,\mathrm{d}t\right] | x_{\dot\tau}, \{y_{\dot\tau} = b\}, \theta\right] < \infty$. This is sufficient, albeit not necessary.

Then, the centered complete transition density immediately follows from Section 6.1.1 and is given by

$$\begin{aligned}\pi(x_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta) &= |\eta'_\theta(v_{\ddot\tau})|\,\mathrm{N}\left[\eta_\theta(v_{\ddot\tau}); \eta_\theta(v_{\dot\tau}), (\ddot\tau - \dot\tau)\rho^2(y_{\dot\tau})\right] \\ &\quad \times \frac{\mathrm{d}\mathbb{X}|(X_{\dot\tau} = \eta_\theta(v_{\dot\tau}), y_{\dot\tau}, \theta)}{\mathrm{d}\mathbb{M}|(X_{\dot\tau} = \eta_\theta(v_{\dot\tau}), y_{\dot\tau})}(x_{(\dot\tau,\ddot\tau)}, \eta_\theta(v_{\ddot\tau})),\end{aligned} \tag{7.21}$$

$$\frac{\mathrm{d}\mathbb{X}|(x_{\dot\tau}, y_{\dot\tau}, \theta)}{\mathrm{d}\mathbb{M}(x_{\dot\tau}, y_{\dot\tau})}(x_{(\dot\tau,\ddot\tau]}) = \exp\left[\frac{\Delta_\theta(x_{\ddot\tau}, y_{\ddot\tau}) - \Delta_\theta(x_{\dot\tau}, y_{\dot\tau})}{\rho^2(y_{\dot\tau})} - \int_{\dot\tau}^{\ddot\tau} \varphi_\theta(x_t, y_{\dot\tau})\,\mathrm{d}t\right], \tag{7.22}$$

$$\Delta_\theta(a, b) = \int \delta_\theta(c, b)\,\mathrm{d}c, \tag{7.23}$$

$$\varphi_\theta(a, b) = \frac{1}{2}\left(\frac{\delta^2_\theta(a, b)}{\rho^2(b)} + \partial_a\delta_\theta(a, b)\right), \tag{7.24}$$

where the dominating measure is $\mathbb{M}|(X_{\{\dot\tau,\ddot\tau\}} = \eta_\theta(v_{\{\dot\tau,\ddot\tau\}}), y_{\dot\tau}) \times \mathrm{Leb}$, $\mathbb{X}|(x_{\dot\tau}, y_{\dot\tau}, \theta)$ is induced by $X_{(\dot\tau,\ddot\tau]}$ and $\mathbb{M}|(x_{\dot\tau}, y_{\dot\tau})$ is induced by $\mathrm{d}X_t = \rho(y_{\dot\tau})\,\mathrm{d}W_t$. In addition, if we define noncentering and centering functions ζ_θ and ζ_θ^{-1} by

$$\zeta_\theta(x_t; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}) = \frac{x_t - \eta_\theta(v_{\dot\tau}) - (\eta_\theta(v_{\ddot\tau}) - \eta_\theta(v_{\dot\tau}))\frac{t-\dot\tau}{\ddot\tau-\dot\tau}}{\rho(y_{\dot\tau})}, \qquad (t \in (\dot\tau, \ddot\tau)) \tag{7.25}$$

and set $Z_{(\dot\tau,\ddot\tau)} = \zeta_\theta(X_{(\dot\tau,\ddot\tau)}; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}})$, the noncentered complete transition density is given by

$$\begin{aligned}\pi(z_{(\dot\tau,\ddot\tau)}, v_{\ddot\tau}|v_{\dot\tau}, y_{\dot\tau}, \theta) &= |\eta'_\theta(v_{\ddot\tau})|\,\mathrm{N}\left[\eta_\theta(v_{\ddot\tau}); \eta_\theta(v_{\dot\tau}), (\ddot\tau - \dot\tau)\rho^2(y_{\dot\tau})\right] \\ &\quad \times \underbrace{\frac{\mathrm{d}\mathbb{X}|(X_{\dot\tau} = \eta_\theta(v_{\dot\tau}), y_{\dot\tau}, \theta)}{\mathrm{d}\mathbb{M}|(X_{\dot\tau} = \eta_\theta(v_{\dot\tau}), y_{\dot\tau})}(\zeta_\theta^{-1}(z_{(\dot\tau,\ddot\tau)}; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}), \eta_\theta(v_{\ddot\tau}))}_{d_\theta(v_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau})} \\ &= \left|\eta'_\theta(v_{\ddot\tau})\right|\mathrm{N}\left[\eta_\theta(v_{\ddot\tau}); \eta_\theta(v_{\dot\tau}), (\ddot\tau - \dot\tau)\rho^2(y_{\dot\tau})\right] e^{\Delta_\theta(\eta_\theta(v_{\ddot\tau}), y_{\dot\tau}) - \Delta_\theta(\eta_\theta(v_{\dot\tau}), y_{\dot\tau})} \\ &\quad \times \underbrace{\exp\left[-\int_{\dot\tau}^{\ddot\tau} \varphi_\theta(\zeta_\theta^{-1}(z_t; y_{\dot\tau}, v_{\{\dot\tau,\ddot\tau\}}), y_{\dot\tau})\,\mathrm{d}t\right]}_{q_\theta(z_{(\dot\tau,\ddot\tau)}, v_{\{\dot\tau,\ddot\tau\}}, y_{\dot\tau})},\end{aligned} \tag{7.26}$$

where the dominating measure is $\mathbb{B}_{(\dot\tau,\ddot\tau)} \times \mathrm{Leb}$.

7.4 Markov Jump Processes with Tridiagonal Generators

Compared to the Markov jump processes that typically arise in the setting of Chapter 6, our current setting differs in two computationally relevant ways:

1. Y has a much bigger state space of size $|\mathcal{G}|$.
2. λ is *tridiagonal*, i.e. all entries except for the main diagonal and its adjacent diagonals are 0.

The larger state space represents an impediment to applying the methods of Section 6.2 naively - for example, the more robust bridge simulation algorithms have complexity $\mathcal{O}(|\mathcal{G}|^3)$, which could be prohibitive for fine approximations. Furthermore, the number of jumps typically increases as $\mathcal{G}$ is refined, which in practice makes the scaling even less favorable.

Fortunately, by exploiting the tridiagonal structure of λ, we can reduce the complexity of all simulation algorithms by at least an order of magnitude. Forward simulation drops in complexity from linear to constant, the linear solve for computing the stationary distribution drops from cubic to linear, and the eigendecomposition required for bridge sampling algorithms drops from cubic to quadratic. The improvement for forward simulation is due to the fact that only 2 states can be accessed from any current state, regardless of $|\mathcal{G}|$. In the following subsections, we give a brief description of the relevant linear algebra of tridiagonal matrices that enables the other improvements.

7.4.1 Linear Solve and Stationary Distribution

In order to compute the stationary vector $\bar{p}$ corresponding to λ, we solve the linear system

$$\lambda^{\mathsf{T}}\bar{p} = 0, \quad 1^{\mathsf{T}}\bar{p} = 1. \tag{7.27}$$

which generally comes at computational cost $\mathcal{O}(|\mathcal{G}|^3)$. For tridiagonal λ, we can solve the system in linear time by applying the detailed balance equations

$$\frac{\bar{p}_{i+1}}{\bar{p}_i} = \frac{\lambda_{i\to i+1}}{\lambda_{i+1\to i}}. \qquad (i = 1, \dots, |\mathcal{G}| - 1) \tag{7.28}$$

For a given $\bar{p}_1$, we solve for the unnormalized stationary vector by cumulatively multiplying the balance equations:

$$\frac{\bar{p}_i}{\bar{p}_1} = \prod_{j=1}^{i-1} \frac{\bar{p}_{j+1}}{\bar{p}_j} = \prod_{j=1}^{i-1} \frac{\lambda_{j\to j+1}}{\lambda_{j+1\to j}} \tag{7.29}$$

The corresponding normalizing constant then becomes $\sum_{k=1}^{|\mathcal{G}|} \frac{\bar{p}_k}{\bar{p}_1}$, and the entries of the stationary vector are given by

$$\bar{p}_i = \frac{\prod_{j=1}^{i-1} \frac{\lambda_{j\to j+1}}{\lambda_{j+1\to j}}}{\sum_{k=1}^{|\mathcal{G}|} \prod_{j=1}^{k-1} \frac{\lambda_{j\to j+1}}{\lambda_{j+1\to j}}}, \tag{7.30}$$

all of which can be computed at linear cost in one sweep.

7.4.2 Eigendecomposition and Bridge Simulation

While the direct sampling and uniformization algorithms are essentially interchangeable for many scenarios, in the context of this chapter the choice is highly consequential. In practice, the elements of $\operatorname{diag}\lambda$ often differ by multiple orders of magnitude, especially when the grid $\mathcal{G}$ is not optimally set. As a consequence, the vast majority of the transitions in the uniformization algorithm are virtual. Therefore, the direct algorithm is the much more robust choice.

Simulating bridges according to the direct sampling algorithm of Section 6.2.4 principally requires 2 operations: eigendecomposing λ upfront at cost $\mathcal{O}(|\mathcal{G}|^3)$, and at each iteration computing $\pi(y_{\ddot{\tau}}|y_{\{\dot{\tau},\omega\}})$ at cost $\mathcal{O}(|\mathcal{G}|)$ for $|\mathcal{G}|$ possible states. Therefore, the latter in general requires an expenditure of order $\mathcal{O}(|\mathcal{G}|^2)$ per iteration. Conversely, in the tridiagonal case, $\pi(y_{\ddot{\tau}}|y_{\{\dot{\tau},\omega\}})$ is only nonzero for $j \subseteq \{y_{\dot{\tau}}-1, y_{\dot{\tau}}+1\}$, and thus at most 2 terms have to be computed, lowering the complexity at each iteration to linear. As for the eigendecomposition, the approach is to first symmetrize λ by way of the *similarity transformation*

$$\tilde{\lambda} = (\operatorname{diag} d)^{-1}\lambda(\operatorname{diag} d), \tag{7.31}$$

$$d_i = \prod_{j=1}^{i-1}\sqrt{\lambda_{j+1\to j}/\lambda_{j\to j+1}}, \tag{7.32}$$

which yields the symmetric tridiagonal matrix $\tilde{\lambda}$. Various off-the-shelf algorithms are available for eigendecomposing tridiagonal matrices, see for example [27] for a log-linear algorithm. The eigenvalues of λ and $\tilde{\lambda}$ coincide, and the eigenvectors of λ are obtained by left-multiplying $(\operatorname{diag} d)$ onto the eigenvectors of $\tilde{\lambda}$, at quadratic cost.

7.5 Marginal Algorithm

We now proceed with the development of a Bernoulli factory MCMC algorithm targeting

$$\pi(v_r, z, y, \theta, \xi|v_s) \propto \pi(\theta)\pi(\xi)\prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau}\pi(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}}|y_{\dot{\tau}}, \lambda) \tag{7.33}$$

by way of the updates

$$(\Theta, \xi): \quad \pi(\theta, \xi|v_\tau, z, y) \propto \pi(\theta)\pi(\xi)\prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau}\pi(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}}|y_{\dot{\tau}}, \lambda), \tag{7.34}$$

$$(V_R, Z, Y): \quad \pi(v_r, z, y|v_s, \theta, \lambda) \propto \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau}\pi(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta)\pi(y_{\ddot{\tau}}|y_{\dot{\tau}}, \lambda), \tag{7.35}$$

keeping in mind that λ is a deterministic function of ξ. As in Section 6.3, the dominating measure of the latter full conditional is

$$\mathbb{L}(\mathrm{d}y) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \mathbb{B}_{(\dot{\tau},\ddot{\tau})}(\mathrm{d}z_{(\dot{\tau},\ddot{\tau})}) \prod_{\dot{r}\in r} \mathrm{Leb}(\mathrm{d}v_{\dot{r}}). \tag{7.36}$$

We exploit the $\varrho = 0$ case by decomposing the former full conditional into the independent updates

$$\Theta: \quad \pi(\theta|v_\tau, z, y) \propto \pi(\theta) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(z_{(\dot{\tau},\ddot{\tau})}, v_{\ddot{\tau}}|v_{\dot{\tau}}, y_{\dot{\tau}}, \theta), \tag{7.37}$$

$$\xi: \quad \pi(\xi|y) \propto \pi(\xi) \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \pi(y_{\ddot{\tau}}|y_{\dot{\tau}}, \lambda). \tag{7.38}$$

The main difference to the algorithm of Section 6.3 is that the ξ-update is not conjugate. Nevertheless, it may be addressed rather conventionally within the Metropolis-within-Gibbs framework.

Remark 8 (Alternative algorithms)**.** *Notice that we could develop an auxiliary and an approximate MCMC algorithm, as well as a MCEM algorithm for MAP estimation by making minor modifications to the algorithms of Chapter 6.*

7.5.1 Diffusion Parameter Update

We carry out the diffusion parameter update by way of a Barker-within-Gibbs step with generic proposal density $\kappa(\theta^\dagger|\theta)$. The formulae and mechanics are exactly as in Section 6.3.1, keeping in mind that some symbols are defined slightly differently.

7.5.2 Regime Parameter Update

We carry out the regime parameter update by way of a Metropolis-within-Gibbs step with generic proposal density $\kappa(\xi^\dagger|\xi)$. Let λ and $\lambda^\dagger$ be the corresponding generators. The acceptance probability is

$$\begin{aligned} \alpha_\Xi &= 1 \wedge \frac{\pi(\xi^\dagger|y)}{\pi(\xi|y)} \frac{\kappa(\xi|\xi^\dagger)}{\kappa(\xi^\dagger|\xi)} \frac{\pi(y|\lambda^\dagger)}{\pi(y|\lambda)} \\ &= 1 \wedge \frac{\kappa(\xi|\xi^\dagger)}{\kappa(\xi^\dagger|\xi)} \frac{\pi(\xi^\dagger)}{\pi(\xi)} \frac{\pi(y|\lambda^\dagger)}{\pi(y|\lambda)} \\ &= 1 \wedge \frac{\kappa(\xi|\xi^\dagger)}{\kappa(\xi^\dagger|\xi)} \frac{\pi(\xi^\dagger)}{\pi(\xi)} \prod_{(\dot{\tau}\sim\ddot{\tau})\in\tau} \frac{\pi(y_{\ddot{\tau}}|y_{\dot{\tau}}, \lambda^\dagger)}{\pi(y_{\ddot{\tau}}|y_{\dot{\tau}}, \lambda)}, \end{aligned} \tag{7.39}$$

where $\pi(y|\lambda)$ is the complete data likelihood of Y and evaluated as in Section 6.3.2.

7.5.3 Hidden Data Update

We carry out the update to $\pi(v_r, z, y | v_s, \theta, \lambda)$ by way of a Barker-within-Gibbs step. For both the independence and the conditional update, the formulae and mechanics are exactly as in Sections 6.3.3 and 6.3.4, keeping in mind that some symbols are defined differently, and with the simplification that ρ does not depend on θ.

7.6 Simulation Studies and Discussion

The goal of this section is to analyze the scaling behavior of our method in the grid resolution ($\mathcal{G}$) regime, where we both extend and refine the discrete approximation to the volatility process. We keep the performance criterion of effective sampling rate set out in Section 2.4 and previously applied in Sections 5.8 and 6.7.

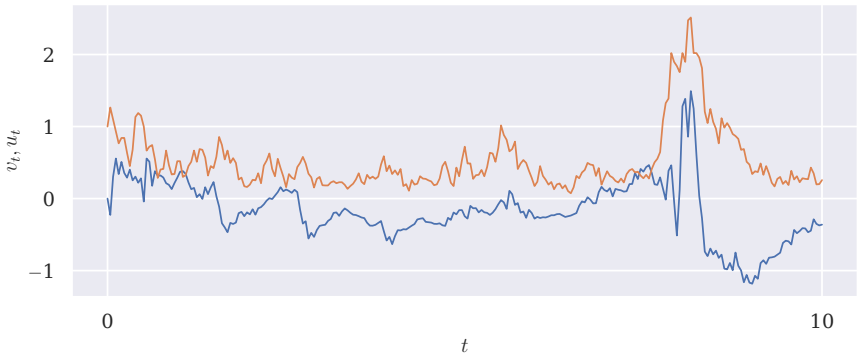


Figure 7.4: Input time series v_s (blue) and unknown volatility series u (orange) for the grid resolution ($|\mathcal{G}|$) regime, generated according to the Tanh-gCIR model with parameters $(\mu_V, \beta_V) = (0, 1)$ and $(\mu_U, \beta_U, \rho_U, \gamma_U) = (1, 1, 1, .75)$.

Our test case is the "Tanh-gCIR" process, consisting of a weakly mean-reverting specification for the observable process V, and a diffusivity specification that generalizes on the square-root specification of the Heston model given by (7.3):

$$\mathrm{d}V_t = \rho_V \tanh\left[\mu_V - V_t\right] \mathrm{d}t + \sqrt{e^{U_t}}\, \mathrm{d}W_t, \qquad (\beta_V > 0) \tag{7.40}$$

$$\mathrm{d}e^{U_t} = \beta_U(\mu_U - V_t)\, \mathrm{d}t + \rho_U e^{U_t(1-\gamma_U)}\, \mathrm{d}W_t. \qquad (\mu_U, \beta_U, \rho_U > 0, \quad \gamma_U \in (0, 1/2)) \tag{7.41}$$

The diffusivity specification is variably known as the *generalized Cox-Ingersoll Ross model* (gCIR) or *Constant elasticity of variance model* (CEV), and reduces to the CIR process for $\gamma_U = 1/2$. For $\gamma_U = 0$, we recover the Vasicek process. For $\gamma_U \in (0, 1/2)$, standard scale analysis shows that the process is confined to $(0, \infty)$, with somewhat heaver tails than the CIR process. We transform the diffusivity specification to the

log scale to better accommodate the fixed-interval grid defined in Section 7.2. By Itō's formula, U follows the SDE

$$\mathrm{d}U_t = (\beta_U(\mu_U e^{-U_t} - 1) - (\rho_U^2 e^{2U_t(\gamma_U - 1)})\,\mathrm{d}t + \rho_U e^{U_t(\gamma_U - 1)}\,\mathrm{d}W_t. \tag{7.42}$$

We complete the specification of the model with the following prior distribution on the unknown parameters:

$$\mu_V, \log\beta_V, \log\mu_U, \log\beta_U, \log\rho_U, \log\frac{1-2\gamma_U}{2\gamma_U - 2} \sim \mathrm{N}\left[0, 1\right]. \tag{7.43}$$

This may be implemented with a similar symbolic preprocessor as for the other model classes. The specification above is implemented by the following, slightly more extensive code snippet:

```
v, x = sp.symbols('v x', real=True)
u = sp.symbols('u', real=True)
m_v = sp.symbols('m_v', real=True)
b_v, m_u, b_u, s_u, g_u = sp.symbols('b_v m_u b_u s_u g_u', positive=
    True)
thi = sp.Array([m_v, b_v])
xi = sp.Array([m_u, b_u, s_u, g_u])
mu_v = b_v * sp.tanh(m_v - v)
sig_v = sp.Integer(1)
gg_u = (1 + 2 * g_u) / (2 + 2 * g_u)
mu_u = b_u * (m_u * sp.exp(-u) - 1) - (s_u * sp.exp((gg_u - 1) * u))
    ** 2
sig_u = s_u * sp.exp((gg_u - 1) * u)
rho_u = sp.exp(u / 2)
```

Our experimental protocol is as follows. Using the specification above, we approximately simulate U on the time interval $[0, 10]$ using a large resolution $|\mathcal{G}|$, and generate 256 equidistant observations from V according to the conditionally correct specification. The resulting time series is shown in Figure 7.4 and exhibits the interesting feature of large transitory volatility spikes causing fast deviation from the stationary mean 0, while only slowly reverting back to 0 after volatility dies back down. The series is shown in Figure 7.4.

Using the time series as input to the resulting specification, we run the MCMC algorithm of Section 7.5 for 100000 iterations, including 10000 for burn-in, with the grid resolution setting $|\mathcal{G}| = 8, 16, 32, 64$. This is preceded by 10000 iterations using an approximate algorithm similar to the one described in Section 6.5. We target an acceptance probability of 23.4% for Metropolis-within-Gibbs steps, and 25% for Barker-within-Gibbs, with a Portkey probability of 1%. Step sizes are adapted according to the *Adapting Increasingly Rarely* (AIR) method of [25]. For Poisson coin simulations we adopt the limiting batch EA version of Section 4.1.3. We use a conditional hidden data update akin to Section 6.3.4, setting the additional tuning parameter p_s to 1/10.

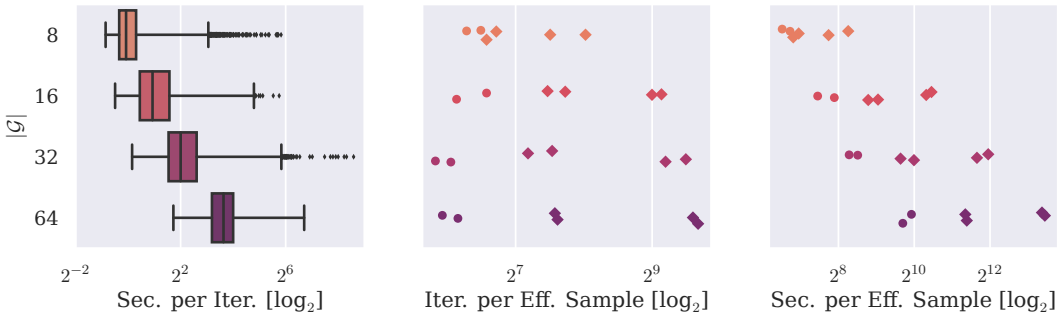


Figure 7.5: Sampling efficiency in the grid resolution regime. The left panel shows the distribution of CPU time per iteration (T/I) throughout the MCMC run. The middle panel shows estimates of the required number of MCMC iterations to generate an effective sample (I/ES), where dots refer to elements of Θ and diamonds to Ξ. Notice that the right panel is obtained by scaling the middle panel by the mean of the distributions in the left panel.

We adopt the same efficiency notion of average time per effective sample (T/ES) introduced in Section 2.4 and (2.45). We decompose T/ES into the average time per iteration (T/I) and the average number of iterations per effective sample (I/ES), and estimate them from the output of the MCMC algorithm. We then investigate the efficiency of the algorithm as $|\mathcal{G}|$ increases.

Efficiency measurements in Figure 7.5 indicate that both T/I and T/ES increase superlinearly in $|\mathcal{G}|$, with only a minor slowdown in I/ES. We deem the T/I slowdown to be largely due to the higher jump frequency in Y as $|\mathcal{G}|$ increases. The I/ES increase at the lower end is largely due to the larger span of $\mathcal{G}$, upon which the prior $\pi(y|\xi)$ becomes a worse proposal for the full conditional, slowing down the hidden data update of Section 7.5.3. Fortuitously, as foreshadowed in Section 7.1, the distribution of T/I is much better behaved than in the vanilla Markov switching diffusion simulations of Section 6.7. Unfortunately, we also observe substantial posterior dependence between Ξ and Y, which results in the slow mixing of Ξ observed in Figures 7.5 and 7.6, particularly for the parameters affecting the meta-volatility γ_ξ.

Figure 7.5 shows that parameter inference is relatively insensitive when $|\mathcal{G}| \geq 16$, except for the case of γ_U which seems better identified, though we note that the estimated effective sample size is only 100 in that instance, and the corresponding KDE is rather heavily smoothed. Figure 7.8 reveals that the automatically chosen grid $\mathcal{G}$ unduly restricts volatility inference, which may drive the discrepancy in parameter estimates. The misspecification manifests as a low-uncertainty plateau at the upper end of $\mathcal{G}$ in the inferred volatility trajectory. Nonetheless, for moderate resolutions, the automatically chosen grid is adequate, and the model correctly captures the volatility spike around $t = 8$.

In the light of those results, we recommend an iterative approach to estimation, starting

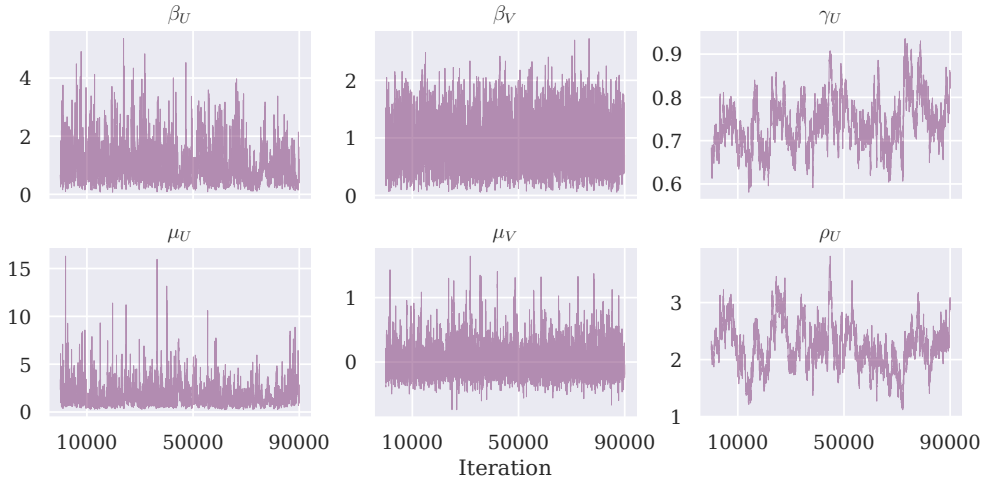


Figure 7.6: Trace plots of Θ and Ξ for $|\mathcal{G}| = 64$.

with moderate values of $|\mathcal{G}|$, and increasing when identifying low-uncertainty plateaus at the boundaries, such as in Figure 7.8. In this instance, $|\mathcal{G}| = 32$ adequately covers the volatility range, and more expensive simulation could be avoided. Larger values of $|\mathcal{G}|$ are inherently computationally challenging, as jumps occur more often, and the algorithms of Section 7.4.2 are quadratic in $|\mathcal{G}|$, even for fixed jump counts. This could be mitigated for more restrictive specifications, for which Ξ is typically better identified a posteriori.

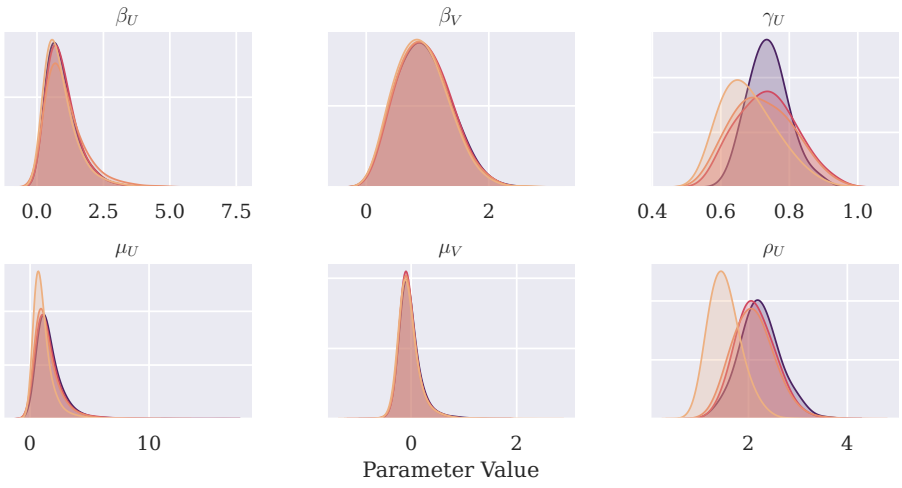


Figure 7.7: Posterior marginals of Θ and Ξ in the grid resolution regime, as estimated by a KDE. Darker shades correspond to a larger $|\mathcal{G}|$.

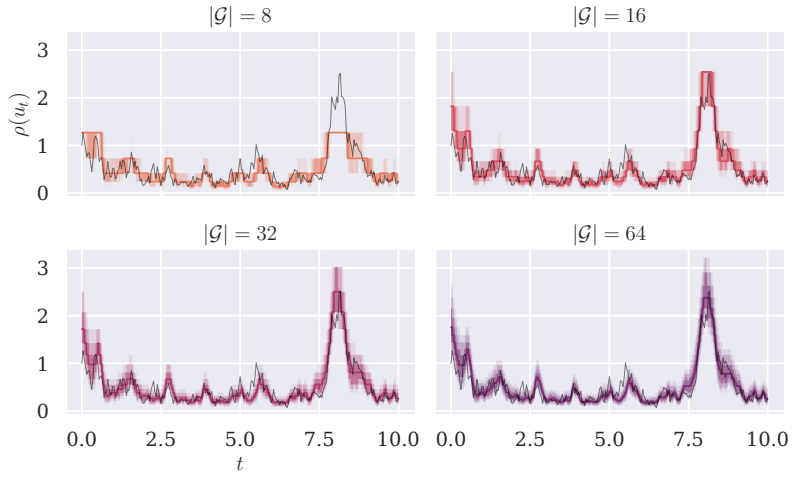


Figure 7.8: Posterior marginals of $\rho(Y_t)$ in the grid resolution regime. The solid colored lines denote the posterior median, and shading in decreasing opacity denotes 50%, 75% and 87.5% credibility intervals, respectively. The solid black line corresponds to the ground truth volatility trajectory.

8 Automatic Implementation of Retrospective Algorithms

Throughout this thesis, we have sought to develop algorithms for generic specifications, e.g. an Itō diffusion model with drift and volatility functions μ and σ. Notice that we are suppressing dependence on parameters θ, even though our framework easily accommodates those. In practice, these algorithms require the evaluation of various transformations of those functions, and bounds thereon. For the purpose of illustration, the pivotal function φ involves the computation of

$$\eta(a) = \int_{v^*}^{a} \frac{\mathrm{d}b}{\sigma(b)}, \quad \delta(a) = \left(\frac{\mu}{\sigma} - \frac{\sigma'}{2}\right) \circ \eta^{-1}(a), \quad \varphi(a) = \frac{1}{2}\left(\delta^2 + \delta'\right)(a), \tag{8.1}$$

which is error prone and time consuming to carry out manually, even for rather simple specifications such as the CEV model with SDE $\mathrm{d}V_t = \beta(\mu - V_t)\,\mathrm{d}t + \rho V_t^{\gamma}\,\mathrm{d}W_t$. More challenging is the need for bounds

$$\varphi^{\downarrow}(a^{\downarrow}, a^{\uparrow}) \leq \inf_{a \in [a^{\downarrow}, a^{\uparrow}]} \varphi(a), \quad \varphi^{\uparrow}(a^{\downarrow}, a^{\uparrow}) \geq \sup_{a \in [a^{\downarrow}, a^{\uparrow}]} \varphi(a), \tag{8.2}$$

because the roots of φ' are often not analytically available, and could exist or not depending on the parameters θ. Therefore, an implementation that requires the manual specification of all required quantities requires substantial effort on the part of the user, far above the mere elementary building blocks of the model, μ and σ. Automating away such obstacles allows for far more flexible experimentation, and can substantially accelerate adoption - for example, Bayesian statistics has gained popularity due the availability of black box probabilistic programming engines such as *BUGS* [49], *JAGS* [101] and *Stan* [45]. Therefore, we deem it useful to provide simple frontends, requiring as little as the specification of μ, σ, the observable diffusion support $\mathcal{V}$, and the parameter space $\mathcal{T}$. The main complication compared to existing engines is that nonconvex functions cannot reliably be bounded numerically. Therefore, we take a *symbolic* approach to automation.

8.1 A Very Short Introduction to Symbolic Computation

Symbolic computation, most prominently implemented in the *Wolfram Mathematica* engine, is the manipulation of mathematical expressions through computers. Mathematical

expressions are usually represented as trees, with nodes corresponding to variables or operators, and with human rules and heuristics formalized and adapted to operate on those trees. Tasks such as differentiation are fulfilled by rather straightforward *substitution rules*, such as

$$\frac{\mathrm{d}\log x}{\mathrm{d}x} \to \frac{1}{x}. \tag{8.3}$$

Rather more artfully, a *simplification* rule will automatically transform

$$2 \times a/2 \to a \tag{8.4}$$

by cancelling the fraction. Simplification is key to counteracting the tendency of programs to proliferate expressions, known as *expression swell*. *Expansion* rules break parenthesized expressions into their constitutions, as in the polynomial expansion rule

$$(a+b)^2 \to a^2 + b^2 + 2ab. \tag{8.5}$$

A further critical feature of symbolic libraries such as *sympy* [90] is the translation of symbolic expressions into numerical code. For example, such a generator would translate the symbolic expression $f(a,b) = a + b$ into the Python function

```
lambda a, b: a + b
```

Accordingly, once an adequate expression has been derived by symbolic manipulation, it can be converted into code that provides efficient numerical evaluations within another Python routine, such as an MCMC algorithm. This generation step is potentially expensive, but only needs to be carried out once at the start of the program.

Armed with those tools, we can attempt to automatically generate symbolic representations of the required functions, and translate those into numerical code. The main limitation of the tools at hand concerns discontinuities in representation. For example, $\int a^{-b}\,\mathrm{d}a$, $a > 0$ is discontinuous at $b = 1$, in the sense that it is represented as

$$\int a^{-b}\,\mathrm{d}a = \begin{cases} \log a & (b = 1) \\ -ba^{b-1} & (\text{otherwise}) \end{cases}. \tag{8.6}$$

This could in principle be resolved by supplying and processing sufficient domain information, such as $b < 1$, but such restrictions are not always supported by symbolic libraries. Nonetheless, there is usually an appropriate reparameterization that avoids the discontinuity, e.g. $c = 1 - b$ such that a^{1-c}, with the more straightforward positivity constraint on c. Assuming that these issues have been resolved, we now investigate the slightly more intricate matter of analytically bounding a function f above and below.

8.2 A Simple Recursive Bound Generator

We now specify a simple recursive algorithm that typically succeeds in bounding general functions f. We note that the reliability of the algorithm partially depends on the level

of sophistication of the symbolic computation library, and the resulting bounds are not necessarily optimal. Loosely speaking, the method we present succeeds if there is an expansion $f \to \sum_i f_i$ such that the library can solve $f_i'(a) = 0$ for all i. Nonetheless, if the algorithm terminates without error, the resulting bounds are guaranteed to be correct, and we have not encountered any failures to terminate successfully in practice.

Given any differentiable expression $f(a)$ defined on an interval $[a^\downarrow, a^\uparrow]$, its minimum f must either lie at one of the roots $f'(a) = 0$, or at the boundaries of the interval. Hence, when the roots of f' are analytically available, we apply a differentiation rule to f, yielding f', and a solution rule to obtain all a such that $f'(a) = 0$. We then return the minimum of

$$\left\{f(a^\downarrow), f(a^\uparrow)\right\} \cup \left\{f(a) : f'(a) = 0\right\} \tag{8.7}$$

as our solution for $f^\downarrow$. When the roots of f' are not available, e.g. for

$$a(a + e^a) \tag{8.8}$$

the solution rule fails. The most straightforward relaxation of the problem, consists of expanding $f(a)$ to a sum $g(a) + h(a)$, independently bounding each of the constituents, i.e.

$$\min_{a\in[a^\downarrow,a^\uparrow]} g(a) + \min_{a\in[a^\downarrow,a^\uparrow]} h(a) \leq \inf_{a\in[a^\downarrow,a^\uparrow]} \left[g(a) + h(a)\right], \tag{8.9}$$

In the above example, we expand $a(a + e^a) \to a^2 + ae^a$, and independently minimize a^2 and ae^a. We obtain the interior minima $0 = \min_{a\in[-1,1]} a^2$ and $-e^{-1} = \min_{a\in[-1,1]} ae^a$ for an overall bound of $-e^{-1} < a(a + e^a)$. If one of the constituents still cannot be bounded, the algorithm could attempt a further expansion step.

The resulting algorithm may be specified as follows:

Algorithm 20 Bounding algorithm (infimum case), mapping a function $f(a)$ to a bound $f^\downarrow(a^\downarrow, a^\uparrow)$. The routine roots $[f]$ solves for the roots of f and also returns the boolean variable success, indicating whether the solver succeeded in finding all the roots. The routine expand fully expands an expression f into a sum of expressions $\sum_i f_i$, where f_i is a pure product.

```
function BoundBelow(f)
    success, a* ← roots [f']
    if success then
        f↓(a↓, a↑) ← min [{f(a↓), f(a↑)} ∪ {f(a) : a ∈ a*}]
        return simplify [f↓]
    for f_i ∈ expand [f] do
        f_i↓ → BoundBelow [f_i]
    f↓ ← Σ_i f_i↓
    return simplify [f↓]
```

Notice that the algorithm generates $f^\downarrow$ as a function of $(a^\downarrow, a^\uparrow)$, since those are not known before runtime. Moreover, there is no sense in which the resulting $f^\downarrow$ is guaranteed to be optimal. In the next section, we show how to express bounds on the functions of interest in terms of bounds on more simple functions, which we have been able to obtain reliably through Algorithm 20.

8.3 Bounding the Path Integrand

We now return to the specific problem of bounding the path integrand

$$\varphi(a) = \frac{1}{2}\left(\delta^2 + \delta'\right)(a). \qquad (a \in \mathcal{X}) \tag{8.10}$$

While this may in principle be generically addressed by Algorithm 20, applying it to φ typically results in looser bounds, and is less reliable, than expressing a bound in terms of more simple functions. We paraphrase $\varphi = \tilde{\varphi} \circ \eta^{-1}$, where

$$\tilde{\varphi}(b) = \frac{1}{2}\left(\tilde{\delta}^2 + \tilde{\delta}'\right)(b), \quad \tilde{\delta}(b) = \left(\frac{\mu}{\sigma} - \frac{\sigma'}{2}\right)(b), \qquad (b \in \mathcal{V}) \tag{8.11}$$

and observe that we can express bounds on φ in terms of bounds on $\tilde{\varphi}$ and η^{-1}, i.e.

$$\underbrace{\tilde{\varphi}^\downarrow((\eta^{-1})^\downarrow(a^\downarrow, a^\uparrow), (\eta^{-1})^\uparrow(a^\downarrow, a^\uparrow))}_{\varphi^\downarrow(a^\downarrow, a^\uparrow)} \leq \varphi(a) \leq \underbrace{\tilde{\varphi}^\uparrow((\eta^{-1})^\downarrow(a^\downarrow, a^\uparrow), (\eta^{-1})^\uparrow(a^\downarrow, a^\uparrow))}_{\varphi^\uparrow(a^\downarrow, a^\uparrow)} \tag{8.12}$$

for $a \in [a^\downarrow, a^\uparrow]$. We define bounds on $\tilde{\varphi}$ by way of the more elementary bounds

$$\underbrace{(\tilde{\delta}')^\downarrow(b^\downarrow, b^\uparrow)}_{2\tilde{\varphi}^\downarrow(b^\downarrow, b^\uparrow)} \leq 2\tilde{\varphi}(b) \leq \underbrace{\tilde{\delta}^\downarrow(b^\downarrow, b^\uparrow)^2 \vee \tilde{\delta}^\uparrow(b^\downarrow, b^\uparrow)^2 + (\tilde{\delta}')^\uparrow(b^\downarrow, b^\uparrow)}_{2\tilde{\varphi}^\uparrow(b^\downarrow, b^\uparrow)}. \qquad (b \in [b^\downarrow, b^\uparrow]) \tag{8.13}$$

Therefore, we have obtained bounds on φ in terms of bounds on the comparatively simple functions $\tilde{\delta}$, $\tilde{\delta}'$ and η^{-1}, which we can reliably obtain by Algorithm 20.

8.4 Specifying the CIR Process in Sympy

Having discussed how to assemble the building blocks of a retrospective algorithm from the elementary diffusion specification, we now give an example of how to provide the elementary diffusion specification in the symbolic Python library *sympy*, namely for the CIR process with SDE

$$\mathrm{d}V_t = \underbrace{b(m - V_t)}_{\mu_\theta}\,\mathrm{d}t + \underbrace{r\sqrt{V_t}}_{\sigma_\theta}\,\mathrm{d}W_t. \qquad (m, b, r > 0) \tag{8.14}$$

The process and its reduced analogue $\eta_\theta(V)$ both have support $\mathcal{V} = \mathcal{X} = \mathbf{R}$, and the parameter vector is $\theta = (m, b, r)$. The full specification can be expressed in *sympy* as follows:

```
v, x = sp.symbols('v x', positive=True)
b, m, r = sp.symbols('b m r', positive=True)
thi = sp.Array([b, m, r])
drift = b * r * (m - v)
vol = r * sp.sqrt(v)
```

Notice that the appropriate domain is passed as a keyword argument. The variables (`v`, `x`, `mu`, `sig`, `th`) fully specify the model, and are provided to a backend which provides the retrospective algorithm with the necessary functions and bounds. Hence, the overhead to adapting an inference or forward simulation algorithm to a new diffusion specification is minimal.

Bibliography

[1] Sanket Agrawal, Dootika Vats, Krzysztof Łatuszyński, and Gareth O Roberts. "Optimal scaling of MCMC beyond Metropolis". In: *Advances in Applied Probability* 55.2 (2023), pp. 492–509.

[2] Yacine Aït-Sahalia. *Closed-form likelihood expansions for multivariate diffusions.* 2002.

[3] Yacine Aït-Sahalia. "Maximum likelihood estimation of discretely sampled diffusions: a closed-form approximation approach". In: *Econometrica* 70.1 (2002), pp. 223–262.

[4] Christophe Andrieu, Gareth O Roberts, et al. "The pseudo-marginal approach for efficient Monte Carlo computations". In: *The Annals of Statistics* 37.2 (2009), pp. 697–725.

[5] Christophe Andrieu and Johannes Thoms. "A tutorial on adaptive MCMC". In: *Statistics and computing* 18.4 (2008), pp. 343–373.

[6] Soren Asmussen, Peter Glynn, Jim Pitman, et al. "Discretization error in simulation of one-dimensional reflecting Brownian motion". In: *The Annals of Applied Probability* 5.4 (1995), pp. 875–896.

[7] Søren Asmussen, Peter W Glynn, and Hermann Thorisson. "Stationarity detection in the initial transient problem". In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 2.2 (1992), pp. 130–157.

[8] Clifford A Ball and Antonio Roma. "Stochastic volatility option pricing". In: *Journal of Financial and Quantitative Analysis* 29.4 (1994), pp. 589–607.

[9] Av A Barker. "Monte carlo calculations of the radial distribution functions for a proton? electron plasma". In: *Australian Journal of Physics* 18.2 (1965), pp. 119–134.

[10] Gopal K Basak, Arnab Bisi, and Mrinal K Ghosh. "Stability of a random diffusion with linear drift". In: *Journal of Mathematical Analysis and Applications* 202.2 (1996), pp. 604–622.

[11] Mark A Beaumont. "Estimation of population growth or decline in genetically monitored populations". In: *Genetics* 164.3 (2003), pp. 1139–1160.

[12] Jean Bérard, Pierre Del Moral, and Arnaud Doucet. "A lognormal central limit theorem for particle approximations of normalizing constants". In: *Electronic Journal of Probability* 19 (2014), pp. 1–28.

[13] Jean Bertoin and Jim Pitman. "Path transformations connecting Brownian bridge, excursion and meander". In: *Bulletin des sciences mathématiques* 118.2 (1994), pp. 147–166.

[14] Alexandros Beskos, Omiros Papaspiliopoulos, Gareth Roberts, et al. "Monte Carlo maximum likelihood estimation for discretely observed diffusion processes". In: *The Annals of Statistics* 37.1 (2009), pp. 223–245.

[15] Alexandros Beskos, Omiros Papaspiliopoulos, and Gareth O Roberts. "A factorisation of diffusion measure and finite sample path constructions". In: *Methodology and Computing in Applied Probability* 10.1 (2008), pp. 85–104.

[16] Alexandros Beskos, Omiros Papaspiliopoulos, Gareth O Roberts, et al. "Retrospective exact simulation of diffusion sample paths with applications". In: *Bernoulli* 12.6 (2006), pp. 1077–1098.

[17] Alexandros Beskos, Omiros Papaspiliopoulos, Gareth O Roberts, and Paul Fearnhead. "Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion)". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.3 (2006), pp. 333–382.

[18] Alexandros Beskos, Gareth O Roberts, et al. "Exact simulation of diffusions". In: *The Annals of Applied Probability* 15.4 (2005), pp. 2422–2444.

[19] Fischer Black and Myron Scholes. "The pricing of options and corporate liabilities". In: *Journal of political economy* 81.3 (1973), pp. 637–654.

[20] PG Blackwell. "Bayesian inference for Markov processes with diffusion and discrete components". In: *Biometrika* 90.3 (2003), pp. 613–627.

[21] Mogens Bladt and Michael Sørensen. "Simple simulation of diffusion bridges with application to likelihood inference for diffusions". In: *Bernoulli* 20.2 (2014), pp. 645–675.

[22] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

[23] Jun Cai. "A Markov model of switching-regime ARCH". In: *Journal of Business & Economic Statistics* 12.3 (1994), pp. 309–316.

[24] George Casella and Edward I George. "Explaining the Gibbs sampler". In: *The American Statistician* 46.3 (1992), pp. 167–174.

[25] Cyril Chimisov, Krzysztof Latuszynski, and Gareth Roberts. "Air Markov chain Monte Carlo". In: *arXiv preprint arXiv:1801.09309* (2018).

[26] Kyriakos Chourdakis. "Continuous time regime switching models and applications in estimating processes with stochastic volatility and jumps". In: *U of London Queen Mary Economics Working Paper* 464 (2002).

[27] Ed S Coakley and Vladimir Rokhlin. "A fast divide-and-conquer algorithm for computing the spectra of real symmetric tridiagonal matrices". In: *Applied and Computational Harmonic Analysis* 34.3 (2013), pp. 379–414.

[28] Zhenyu Cui, J Lars Kirkby, and Duy Nguyen. "A general valuation framework for SABR and stochastic local volatility models". In: *SIAM Journal on Financial Mathematics* 9.2 (2018), pp. 520–563.

[29] Didier Dacunha-Castelle and Danielle Florens-Zmirou. "Estimation of the coefficients of a diffusion from discrete observations". In: *Stochastics: An International Journal of Probability and Stochastic Processes* 19.4 (1986), pp. 263–284.

[30] George Deligiannidis, Arnaud Doucet, and Michael K Pitt. "The correlated pseudomarginal method". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 80.5 (2018), pp. 839–870.

[31] Luc Devroye. "Sample-based non-uniform random variate generation". In: *Proceedings of the 18th conference on Winter simulation*. ACM. 1986, pp. 260–265.

[32] Joseph L Doob. "Heuristic approach to the Kolmogorov-Smirnov theorems". In: *The Annals of Mathematical Statistics* (1949), pp. 393–403.

[33] Arnaud Doucet, Michael K Pitt, George Deligiannidis, and Robert Kohn. "Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator". In: *Biometrika* 102.2 (2015), pp. 295–313.

[34] Garland B Durham and A Ronald Gallant. "Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes". In: *Journal of Business & Economic Statistics* 20.3 (2002), pp. 297–338.

[35] Ola Elerian, Siddhartha Chib, and Neil Shephard. "Likelihood inference for discretely observed nonlinear diffusions". In: *Econometrica* 69.4 (2001), pp. 959–993.

[36] Charles Engel and James D Hamilton. "Long swings in the dollar: Are they in the data and do markets know it?" In: *The American Economic Review* (1990), pp. 689–713.

[37] Bjørn Eraker. "MCMC analysis of diffusion models with application to finance". In: *Journal of Business & Economic Statistics* 19.2 (2001), pp. 177–191.

[38] Paul Fearnhead and Zhen Liu. "On-line inference for multiple changepoint problems". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69.4 (2007), pp. 589–605.

[39] Paul Fearnhead, Omiros Papaspiliopoulos, and Gareth O Roberts. "Particle filters for partially observed diffusions". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.4 (2008), pp. 755–777.

[40] Danielle Florens. "Estimation of the diffusion coefficient from crossings". In: *Statistical Inference for Stochastic Processes* 1.2 (1998), pp. 175–195.

[41] Wai Mun Fong and Kim Hock See. "A Markov switching model of the conditional volatility of crude oil futures prices". In: *Energy Economics* 24.1 (2002), pp. 71–95.

[42] Gersende Fort and Eric Moulines. "Convergence of the Monte Carlo expectation maximization for curved exponential families". In: *The Annals of Statistics* 31.4 (2003), pp. 1220–1259.

[43] Alan E Gelfand, Sujit K Sahu, and Bradley P Carlin. "Efficient parametrisations for normal linear mixed models". In: *Biometrika* 82.3 (1995), pp. 479–488.

[44] Alan E Gelfand and Adrian FM Smith. "Sampling-based approaches to calculating marginal densities". In: *Journal of the American statistical association* 85.410 (1990), pp. 398–409.

[45] Andrew Gelman, Daniel Lee, and Jiqiang Guo. "Stan: A probabilistic programming language for Bayesian inference and optimization". In: *Journal of Educational and Behavioral Statistics* 40.5 (2015), pp. 530–543.

[46] Stuart Geman and Donald Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images". In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), pp. 721–741.

[47] Charles J Geyer. "Practical markov chain monte carlo". In: *Statistical science* (1992), pp. 473–483.

[48] Mrinal K Ghosh, Aristotle Arapostathis, and Steven I Marcus. "Optimal control of switching diffusions with application to flexible manufacturing systems". In: *SIAM Journal on Control and Optimization* 31.5 (1993), pp. 1183–1204.

[49] Wally R Gilks, Andrew Thomas, and David J Spiegelhalter. "A language and program for complex Bayesian modelling". In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 43.1 (1994), pp. 169–177.

[50] Daniel T Gillespie. "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions". In: *Journal of computational physics* 22.4 (1976), pp. 403–434.

[51] Tilmann Gneiting and Adrian E Raftery. "Strictly proper scoring rules, prediction, and estimation". In: *Journal of the American statistical Association* 102.477 (2007), pp. 359–378.

[52] Flávio B Gonçalves, Krzysztof Łatuszyński, Gareth O Roberts, et al. "Barker's algorithm for Bayesian inference with intractable likelihoods". In: *Brazilian Journal of Probability and Statistics* 31.4 (2017), pp. 732–745.

[53] Flávio B Gonçalves, Krzysztof Łatuszyński, and Gareth O Roberts. "Exact Monte Carlo likelihood-based inference for jump-diffusion processes". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* (Apr. 2023). qkad022. ISSN: 1369-7412. DOI: 10.1093/jrsssb/qkad022. eprint: https://academic.oup.com/jrsssb/advance-article-pdf/doi/10.1093/jrsssb/qkad022/49737503/qkad022.pdf. URL: https://doi.org/10.1093/jrsssb/qkad022.

[54] Peter J Green, Krzysztof Łatuszyński, Marcelo Pereyra, and Christian P Robert. "Bayesian computation: a summary of the current state, and samples backwards and forwards". In: *Statistics and Computing* 25.4 (2015), pp. 835–862.

[55] Patrick S Hagan, Deep Kumar, Andrew S Lesniewski, and Diana E Woodward. "Managing smile risk". In: *The Best of Wilmott* 1 (2002), pp. 249–296.

[56] James D Hamilton. "A new approach to the economic analysis of nonstationary time series and the business cycle". In: *Econometrica: Journal of the Econometric Society* (1989), pp. 357–384.

[57] James D Hamilton and Raul Susmel. "Autoregressive conditional heteroskedasticity and changes in regime". In: *Journal of econometrics* 64.1-2 (1994), pp. 307–333.

[58] W Keith Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: (1970).

[59] Hans Hersbach. "Decomposition of the continuous ranked probability score for ensemble prediction systems". In: *Weather and Forecasting* 15.5 (2000), pp. 559–570.

[60] Steven L Heston. "A closed-form solution for options with stochastic volatility with applications to bond and currency options". In: *The review of financial studies* 6.2 (1993), pp. 327–343.

[61] Steven L Heston. "A simple new formula for options with stochastic volatility". In: (1997).

[62] Max Hird, Samuel Livingstone, and Giacomo Zanella. "A fresh take on 'Barker dynamics' for MCMC". In: *Monte Carlo and Quasi-Monte Carlo Methods: MCQMC 2020, Oxford, United Kingdom, August 10–14*. Springer, 2022, pp. 169–184.

[63] Asger Hobolth. "A Markov chain Monte Carlo expectation maximization algorithm for statistical analysis of DNA sequence evolution with neighbor-dependent substitution rates". In: *Journal of Computational and Graphical Statistics* 17.1 (2008), pp. 138–162.

[64] Asger Hobolth and Eric A Stone. "Simulation from endpoint-conditioned, continuous-time Markov chains on a finite state space, with applications to molecular evolution". In: *The annals of applied statistics* 3.3 (2009), p. 1204.

[65] MEA Hodgson. "A Bayesian restoration of an ion channel signal". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.1 (1999), pp. 95–114.

[66] John Hull and Alan White. "The pricing of options on assets with stochastic volatilities". In: *The journal of finance* 42.2 (1987), pp. 281–300.

[67] Eric Jacquier, Nicholas G Polson, and Peter E Rossi. "Bayesian analysis of stochastic volatility models". In: *Journal of Business & Economic Statistics* 20.1 (2002), pp. 69–87.

[68] Paul A Jenkins and Dario Spano. "Exact simulation of the Wright–Fisher diffusion". In: *The Annals of Applied Probability* 27.3 (2017), pp. 1478–1509.

[69] Christopher S Jones. *Bayesian analysis of the short-term interest rate*. Tech. rep. Working paper, The Wharton School, University of Pennsylvania, 1997.

[70] Galin L Jones. "On the Markov chain central limit theorem". In: *Probability surveys* 1 (2004), pp. 299–320.

[71] Ioannis Karatzas and Steven E Shreve. "Brownian motion and stochastic calculus Springer-Verlag". In: *New York* (1991).

[72] Gregor Kastner, Sylvia Frühwirth-Schnatter, and Hedibert Freitas Lopes. "Efficient Bayesian inference for multivariate factor stochastic volatility models". In: *Journal of Computational and Graphical Statistics* 26.4 (2017), pp. 905–917.

[73] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. "Optimal detection of changepoints with a linear computational cost". In: *Journal of the American Statistical Association* 107.500 (2012), pp. 1590–1598.

[74] Chang-Jin Kim and Charles R Nelson. "Has the US economy become more stable? A Bayesian approach based on a Markov-switching model of the business cycle". In: *Review of Economics and Statistics* 81.4 (1999), pp. 608–616.

[75] Sangjoon Kim, Neil Shephard, and Siddhartha Chib. "Stochastic volatility: likelihood inference and comparison with ARCH models". In: *The review of economic studies* 65.3 (1998), pp. 361–393.

[76] Peter Eris Kloeden, Eckhard Platen, and Henri Schurz. *Numerical solution of SDE through computer experiments*. Springer Science & Business Media, 2012.

[77] Nikolaj Vladimirovič Krylov. *Controlled diffusion processes*. Vol. 14. Springer Science & Business Media, 2008.

[78] Harold J Kushner and Paul G Dupuis. *Numerical methods for stochastic control problems in continuous time*. Vol. 24. Springer Science & Business Media, 2001.

[79] Krzysztof Łatuszyński, Ioannis Kosmidis, Omiros Papaspiliopoulos, and Gareth O Roberts. "Simulating events of unknown probabilities via reverse time martingales". In: *Random Structures & Algorithms* 38.4 (2011), pp. 441–452.

[80] Krzysztof Łatuszyński and Gareth O Roberts. "CLTs and asymptotic variance of time-sampled Markov chains". In: *Methodology and Computing in Applied Probability* 15.1 (2013), pp. 237–247.

[81] John C Liechty and Gareth O Roberts. "Markov chain Monte Carlo methods for switching diffusion models". In: *Biometrika* 88.2 (2001), pp. 299–315.

[82] Jun S Liu. "The fraction of missing information and convergence rate for data augmentation". In: *Computing Science and Statistics* (1994), pp. 490–490.

[83] Chia Chun Lo and Konstantinos Skindilias. "An improved Markov chain approximation methodology: Derivatives pricing and model calibration". In: *International Journal of Theoretical and Applied Finance* 17.07 (2014), p. 1450047.

[84] Thomas A Louis. "Finding the observed information matrix when using the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 44.2 (1982), pp. 226–233.

[85] Xuerong Mao and Chenggui Yuan. *Stochastic differential equations with Markovian switching*. Imperial college press, 2006.

[86] Harley H McAdams and Adam Arkin. "Stochastic mechanisms in gene expression". In: *Proceedings of the National Academy of Sciences* 94.3 (1997), pp. 814–819.

[87] Margaret M McConnell and Gabriel Perez-Quiros. "Output fluctuations in the United States: What has changed since the early 1980's?" In: *American Economic Review* 90.5 (2000), pp. 1464–1476.

[88] Kerrie L Mengersen and Richard L Tweedie. "Rates of convergence of the Hastings and Metropolis algorithms". In: *The annals of Statistics* 24.1 (1996), pp. 101–121.

[89] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, et al. "Equation of state calculations by fast computing machines". In: *The journal of chemical physics* 21.6 (1953), pp. 1087–1092.

[90] Aaron Meurer, Christopher P Smith, Mateusz Paprocki, Ondřej Čertík, et al. "SymPy: symbolic computing in Python". In: *PeerJ Computer Science* 3 (2017), e103.

[91] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[92] Aleksandar Mijatović and Martijn Pistorius. "Continuously monitored barrier options under Markov processes". In: *Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics* 23.1 (2013), pp. 1–38.

[93] Timothy D Mount, Yumei Ning, and Xiaobin Cai. "Predicting price spikes in electricity markets using a regime-switching model with time-varying parameters". In: *Energy Economics* 28.1 (2006), pp. 62–80.

[94] Şerban Nacu, Yuval Peres, et al. "Fast simulation of new coins from old". In: *The Annals of Applied Probability* 15.1A (2005), pp. 93–115.

[95] Rasmus Nielsen. "Mapping mutations on phylogenies". In: *Systematic biology* 51.5 (2002), pp. 729–739.

[96] Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[97] Omiros Papaspiliopoulos, Gareth O Roberts, and Martin Sköld. "A general framework for the parametrization of hierarchical models". In: *Statistical Science* (2007), pp. 59–73.

[98] Asger Roer Pedersen. "A new approach to maximum likelihood estimation for stochastic differential equations based on discrete observations". In: *Scandinavian journal of statistics* (1995), pp. 55–71.

[99] Peter H Peskun. "Optimum monte-carlo sampling using markov chains". In: *Biometrika* 60.3 (1973), pp. 607–612.

[100] Eckhard Platen. *A non-linear stochastic volatility model*. Centre for Mathematics and Its Applications, Australian National University, 1998.

[101] Martyn Plummer et al. "JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling". In: *Proceedings of the 3rd international workshop on distributed statistical computing*. Vol. 124. 125.10. Vienna, Austria. 2003, pp. 1–10.

[102] M Pollock, AM Johansen, and GO Roberts. "On the exact and e-strong simulation of (jump) diffusions". In: *Bernoulli* (2016).

[103] Vinayak Rao and Yee Teh. "MCMC for continuous-time discrete-state systems". In: *Advances in Neural Information Processing Systems* 25 (2012).

[104] Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*. Vol. 2. Springer, 1999.

[105] Gareth O Roberts, Omiros Papaspiliopoulos, and Petros Dellaportas. "Bayesian inference for non-Gaussian Ornstein–Uhlenbeck stochastic volatility processes". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 66.2 (2004), pp. 369–393.

[106] Gareth O Roberts and Jeffrey S Rosenthal. "Optimal scaling for various Metropolis-Hastings algorithms". In: *Statistical science* 16.4 (2001), pp. 351–367.

[107] Gareth O Roberts and Jeffrey S Rosenthal. "Variance bounding Markov chains". In: *The Annals of Applied Probability* 18.3 (2008), pp. 1201–1214.

[108] Gareth O Roberts and Osnat Stramer. "On inference for partially observed nonlinear diffusion models using the Metropolis–Hastings algorithm". In: *Biometrika* 88.3 (2001), pp. 603–621.

[109] Pedro Santa-Clara. "Simulated Likeliehood Estimation of Diffusions With an Application to the Short Tem Interest Rate". In: (1997).

[110] Louis O Scott. "Option pricing when the variance changes randomly: Theory, estimation, and an application". In: *Journal of Financial and Quantitative analysis* 22.4 (1987), pp. 419–438.

[111] Giorgos Sermaidis, Omiros Papaspiliopoulos, Gareth O Roberts, Alexandros Beskos, et al. "Markov chain Monte Carlo for exact inference for diffusions". In: *Scandinavian Journal of Statistics* 40.2 (2013), pp. 294–321.

[112] Chris Sherlock, Alexandre H Thiery, Gareth O Roberts, and Jeffrey S Rosenthal. "On the efficiency of pseudo-marginal random walk Metropolis algorithms". In: *The Annals of Statistics* 43.1 (2015), pp. 238–275.

[113] Tokuzo Shiga, Akinobu Shimizu, et al. "Infinite dimensional stochastic differential equations and their applications". In: *Journal of Mathematics of Kyoto University* 20.3 (1980), pp. 395–416.

[114] Isao Shoji and Tohru Ozaki. "Estimation for nonlinear stochastic differential equations by a local linearization method". In: *Stochastic Analysis and Applications* 16.4 (1998), pp. 733–752.

[115] Adrian FM Smith and Gareth O Roberts. "Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 55.1 (1993), pp. 3–23.

[116] Alan Sokal. "Monte Carlo methods in statistical mechanics: foundations and new algorithms". In: *Functional integration*. Springer, 1997, pp. 131–192.

[117] Robert H Swendsen and Jian-Sheng Wang. "Nonuniversal critical dynamics in Monte Carlo simulations". In: *Physical review letters* 58.2 (1987), p. 86.

[118] Luke Tierney. "A note on Metropolis-Hastings kernels for general state spaces". In: *Annals of applied probability* (1998), pp. 1–9.

[119] Luke Tierney. "Introduction to general state-space Markov chain theory". In: *Markov chain Monte Carlo in practice* (1996), pp. 59–74.

[120] Luke Tierney. "Markov chains for exploring posterior distributions". In: *the Annals of Statistics* (1994), pp. 1701–1728.

[121] NG Van Kampen. "Stochastic processes in chemistry and physics". In: *Chaos* (1981).

[122] D Vats, FB Gonçalves, K Łatuszyński, and GO Roberts. "Efficient Bernoulli factory Markov chain Monte Carlo for intractable posteriors". In: *Biometrika* 109.2 (2022), pp. 369–385.

[123] Dootika Vats, James M Flegal, and Galin L Jones. "Multivariate output analysis for Markov chain Monte Carlo". In: *Biometrika* 106.2 (2019), pp. 321–337.

[124] Greg CG Wei and Martin A Tanner. "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms". In: *Journal of the American statistical Association* 85.411 (1990), pp. 699–704.

[125] James B Wiggins. "Option values under stochastic volatility: Theory and empirical estimates". In: *Journal of financial economics* 19.2 (1987), pp. 351–372.

[126] Yaming Yu and Xiao-Li Meng. "To center or not to center: That is not the question—an Ancillarity–Sufficiency Interweaving Strategy (ASIS) for boosting MCMC efficiency". In: *Journal of Computational and Graphical Statistics* 20.3 (2011), pp. 531–570.