A Study on the Properties and Applications of Advanced MCMC Methods for Diffusion Models

For the degree of Doctor of Philosophy in Statistical Science

University College London

Erik Pazos

September 2014

I confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Name: ERIK PAZOS

Signature:

Date:

I dedicate this thesis to all the friends and family who helped me during its construction. Special thanks to my father for helping me with the final proofread. And finally, my deepest gratitude to my supervisor Alex Beskos, whose expert advice and guidance made this thesis possible.

Abstract

The aim of this thesis is to find ways to make advanced Markov Chain Monte Carlo (MCMC) algorithms more efficient. Our framework is relevant for target distributions defined as change of measures from Gaussian laws; we use this definition because it provides the flexibility to apply our methods to a wider range of problems –including models driven by Stochastic Differential Equations (SDE). The advanced MCMC algorithms presented in this thesis are well-defined on the infinite-dimensional path-space and exhibit superior properties in terms of computational complexity. The consequence of the well-definition of these algorithms is that they have mesh-free mixing properties and their convergence time does not deteriorate when the dimension of the path increases. The contributions we make in this thesis are in four areas: First, we present a new proof for the well-posedness of the advanced Hybrid Monte Carlo (HMC) algorithm; this proof allows us to verify the validity of the required assumptions for well-posedness in several practical applications. Second, by comparing analytically and with numerical examples the computational costs of different algorithms, we show that the advanced Random Walk Metropolis and the Metropolis-adjusted Langevin algorithm (MALA) have similar complexity when applied to 'long' diffusion paths, whereas the HMC algorithm is more efficient than both. Third, we demonstrate that the Golightly-Wikinson transformation can be applied to a wider range of applications – than the typically used Lamperti– when using HMC algorithms to sample from complex target distributions such as SDEs with general diffusion coefficients. Four, we implemented a novel joint update scheme to sample from a path observed with error, where the path itself was driven by a fractional Brownian motion (fBm) instead of a Wiener process. Here HMC's scaling properties proved desirable, since, the non-Markovian properties of fBm made techniques like blocking overly expensive. We achieved this by a well-planned use of the Davies-Harte algorithm to provide the mapping between fBm and uncorrelated white noise that we used to decouple the a-priori involved model parameters from the high-dimensional latent variables. Finally, we showed numerically that our proposed algorithm works efficiently and provided ample comparisons to corroborate it.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Markov Chain Monte Carlo (MCMC) methods have been around for more than two decades now and provide an intuitive and powerful mechanism to sample from complex posterior distributions where other methods may prove difficult to work[1]. Due to the high versatility and power inherent of the MCMC methodology, the range of applications has been increasing rapidly; this has in turn created a need for the rapid advancement of MCMC methods. Not surprisingly, this need has extended to high-dimensional distributions, where standard MCMC methods could break down, and some advanced MCMC methods are needed to improve efficiency. In this thesis we will focus on studying such advanced algorithms as well as proposing a few new ones. First, in this Chapter, we present the main standard MCMC methods used in applications and that are relevant for the advanced methodology shown in the next Chapter. Also, we present some introductory material for Stochastic Differential Equations (SDEs), focusing on the aspects of these processes which are important when developing the advanced MCMC algorithm on the diffusion pathspace as presented in the next chapter.

1.1 Markov Chain Monte Carlo

In this section we provide a brief introduction to the basic concepts underpinning the MCMC methodology as well as some more modern developments in terms of new advanced methods that are robust in high-dimensions and that have provided the motivation for this thesis.

[1] See e.g. [41, 14] for good summaries on the algorithms and applications

1.1.1 Monte Carlo Integration

All MCMC algorithms stem from the same basic idea of Monte Carlo integration, which is summarised in [41] and we include here for the sake of completion. The main idea stems from adopting a Bayesian framework and we will be taking such a Bayesian stance when explaining all these concepts.

Lets assume that we have observed data Y with unknown parameters θ, so that the likelihood of the data is $L(Y \mid \theta)$. A Bayesian framework treats both parameters and data as random variables, therefore using Bayes rule we obtain the posterior distribution of main interest:

$$P(\theta \mid Y) \propto P(\theta)L(Y \mid \theta). \tag{1.1.1}$$

Consider the case we are given a function $f = f(\theta)$ and we are interested in evaluating the posterior expectation:

$$E[f(\theta) \mid Y] = \int f(\theta)P(\theta \mid Y)d\theta. \tag{1.1.2}$$

The problem here is that in most cases (especially in high-dimensions or in the presence of latent variables) the analytic evaluation of this integral is impossible. The solution is to perform some sort of numerical integration using samples from the posterior.

Now that we have shown the Bayesian motivation, we move towards a more generic framework and also of more general interest. Assume that we are interested in computing the following integral:

$$E[f(X)] = \int f(X)\Pi(X)dX \tag{1.1.3}$$

for some given $f = f(X)$ and distribution Π. Notice that in a Bayesian setting $\Pi(X)$ could correspond to posterior distribution as in (1.1.1). Lets also assume, for the time being, than we can sample $x_1, x_2, \ldots, x_n$ directly from $\Pi(X)$. Then, under some conditions on the distribution of $f(x_i)$ (e.g. finite L_1-norm) it follows that:

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} f(x_i) \longrightarrow E[f(X)], \quad \text{almost surely} \tag{1.1.4}$$

via the Strong Law of Large Numbers (see e.g. [45]). This very important result could be labeled Monte Carlo Integration and allows for an accurate estimate of the unknown expectation when using large values of n. Assuming a finite second moment, the Central Limit Theorem dictates that the rate of convergence is of the order $\mathcal{O}(1/\sqrt{N})$.

1.1.2 Standard MCMC Algorithms

When discussing the Monte Carlo Integration principle in the previous section we made the implicit assumption that one could easily generate samples from $\Pi(X)$. Unfortunately, this is not typically the case in a large variety of real life problems. The principle behind Markov Chain Monte Carlo (MCMC) is simple: we are given a maybe complex, high-dimensional distribution $\Pi(X)$, which we will refer to from now on as the 'target' distribution, and we want to generate samples $\{x_t\}_{t\geq 1}$ from this target $\Pi(X)$. We assume that we can't sample from the target directly using standard methods (e.g. inverse cdf or rejection sampling [67]). An effective solution is to set up a Markov chain whose stationary distribution is $\Pi(\cdot)$. Once we have this Markov chain at hand, we can sample its trajectory $\{x_0, x_1, ..., x_n\}$ using iteratively it's transition dynamics $P(X_{t+1} \mid X_t)$, which depend on the current state. Assuming we reach equilibrium, samples of the Markov chain can then be treated as a correlated sample from the invariant (thus also equilibrium) distribution Π. Typically, a first part of the Markov chain trajectory is discarded, i.e. a so-called 'burn-in' period must be taken under consideration.

Perhaps the most well-known MCMC algorithms correspond to following the Metropolis-Hastings (MH) methodology. This methodology was originally developed by Metropolis [58] and later expanded and generalised by Hasting [48]. Due to its general nature, the Metropolis-Hastings forms the basis for many of the specific MCMC algorithms used in practice. The MH algorithm develops a Markov transition kernel $P(\cdot \mid \cdot)$ that has as equilibrium distribution the target $\Pi(\cdot)$, by employing candidates from a simple proposal distribution $Q(\cdot \mid \cdot)$. The algorithm works as follows:

1. A new candidate value Y is suggested from the proposal distribution given the current value X_t i.e. $Y \sim Q(\cdot \mid X_t)$.

2. This value is accepted with probability $\alpha(X_t, Y)$ where:

$$\alpha(X_t, Y) = \min\left(1, \frac{\Pi(Y)Q(X_t|Y)}{\Pi(X_t)Q(Y|X_t)}\right). \tag{1.1.5}$$

If accepted, then the next Markov chain value is set equal to the proposed value $X_{t+1} = Y$;

If rejected, then the next Markov chain value is set equal to the current value $X_{t+1} = X_t$.

3. Repeat until a long enough trajectory $\{X_0, X_1, \ldots, X_t, \ldots\}$ has been drawn.

Given a sufficiently long burn-in time, say k, then $\{x_k, x_{k+1}, ..., x_n\}$ will be treated as correlated samples with marginal distribution $\Pi(X)$[2]. The main reason that the MH algorithm is so powerful is that $Q(\cdot \mid \cdot)$ can take, in principle, any form its user wants[3].

Later, we continue with the idea that the MH algorithm presented forms a general framework that can provide a number of specific algorithms. The particular type of MH algorithm chosen will depend on the choice of the proposal distribution $Q(\cdot \mid \cdot)$. Consider, for example, using the following proposal distribution:

$$Q(X_{t+1} \mid X_t) \; : \;\; X_{t+1} = X_t + \xi,$$

where, ξ is distributed as a symmetric distribution, say a Gaussian one:

$$\xi \sim N(0, \Sigma),$$

with Σ being a variance parameter which can be tuned further to improve the mixing properties of the algorithm. This is equivalent to saying $Q(\cdot \mid \cdot)$ is determined as follows:

$$Q(X_{t+1} \mid X_t) \sim N(X_t, \Sigma). \tag{1.1.6}$$

This specification of MH corresponds to the Random Walk Metropolis (RWM) with its main characteristic being that the proposal kernel $Q(\cdot|\cdot)$ cancels out from the acceptance probability (1.1.5) due to having $Q(Y|X_t) = Q(X_t|Y)$.

Another specification of MH is the one corresponding to the Independent Sampler. In this case we propose a move independent of its current position, that is:

$$Q(X_{t+1}|X_t) \equiv Q(X_{t+1}). \tag{1.1.7}$$

We will be using a version of this algorithm in the context of the SDE applications of interest in this thesis.

Then we move on to a different kind of MCMC algorithm called the Gibbs sampler. Suppose we are interested in sampling a posterior on d parameters of in-

[2] There are now well-studied methods for determining the burn-in period, see e.g. [22], [13]

[3] Of course, some regularity conditions are needed to enforce convergence in equilibrium, see e.g. [68]

terest, say $\Theta = \{\theta_1, \theta_2, \dots, \theta_d\}$, from the joint distribution $P(\theta_1, \theta_2, \dots, \theta_d)$. Then, the algorithm works like follows:

1. Choose suitable starting values for the parameters $\{\theta_1, \theta_2, \dots, \theta_d\}$.
2. Alternate sampling all parameters, each one from its full conditional distribution $P(\theta_i \mid \text{all other variables})$ i.e. sample each univariate variable from the conditional distribution given all the other variables.
3. Repeat previous step until a sufficient large trajectory is drawn.

Then, given a sufficiently large burn-in time, the algorithm will converge to the target distribution $\Pi(\Theta)$. Again, convergence requires some regularity condition on $\Pi(\Theta)$ as there will be instances when the algorithm will fail to convergence to equilibrium or will converge very slowly for practical purposes. In the latter case of slow convergence, a standard context where it arises is for Gibbs samplers in the presence of strong correlation among variables[4]. We will return to this point in more detail in subsequent sections.

It is important to notice that different algorithms can be derived by combining the general direction within MH and Gibbs samplers. For instance, the Gibbs sampler provides a way of sampling parameters according to a target joint probability distribution, but it requires being able to sample from the full conditional distributions easily, e.g. the first one $P(\theta_1 \mid \theta_2, \dots, \theta_k)$– something that may not be possible for many problems[5]. Luckily, the MH methodology provides us a method to overcome this problem: one simply applies the MH correction (i.e. proposal and accept/reject rule) to each of the full conditionals if needed; the final algorithm will always have the correct invariant distribution. This briefly describes the so-called Metropolis-within-Gibbs methodology which is typically used in practice. The conclusion is that we can nest MH algorithms within a Gibbs sampler to obtain the final algorithm.

1.1.3 Derivative-Driven MCMC Algorithms

In the previous section, we looked at MCMC algorithms such as Random Walk Metropolis (RWM) and the Independent Sampler[6]. RWM is often referred to as a blind algorithm since the proposal kernel, in principle, does not depend on the target distribution (in practice, for the algorithm to be effective, step-sizes in

[4] see [41]

[5] See [29] for some early attempts to overcome this problem using iterated rejection sampling

[6] See [41] for details on other algorithms

different direction will have to somewhat adapt to the covariance structure of the target).

In this section we look at ways of improving the proposal distribution by using information –in the form of the gradient of the log-target density– in the proposal mechanism. That is, we look at derivative-driven MCMC algorithms. Derivative-driven methods use the first derivative of the target distribution to attempt to produce proposals towards the center of the domain of the target distribution. Our aim is to create better proposals that lead to a higher acceptance probability and better mixing. It is important to notice that it is not necessarily the case that derivative-driven algorithms are better than non-derivative-driven methods, as the latter ones will always have the advantage that little needs to be known about the target distribution derivatives to develop the algorithm.

The first derivative-driven method we cover is known as the Metropolis Adjusted Langevin Algorithm (MALA)[7]. MALA is based on the stationary properties of the so-called Langevin Stochastic Differential Equation (SDE). In the equation below, W_t denotes a standard Brownian motion on $\mathbb{R}^d$. Also, for arbitrary function $f = f(x_1, x_2, x_3, \dots, x_d)$ we define the gradient as the vector of first-order partial derivatives, so that for an abstract function $f = f(x)$ we have $\nabla f(x_1, x_2, x_3 \dots, x_d) = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}, \dots, \frac{\partial f}{\partial x_d})^\top$.

Theorem 1.1.1. Langevin SDE*:*

Let Π *be an arbitrary target distribution on* $\mathbb{R}^d$*. Then, given regularity conditions on* Π *(see e.g. [72]), the solution to the following SDE:*

$$dX_t = \tfrac{1}{2}\,\mathcal{C}\,\nabla \log \Pi(X_t)dt + \sqrt{\mathcal{C}}\,dW_t \tag{1.1.8}$$

with $\mathcal{C}$ *a positive-definite matrix on* $\mathbb{R}^{d\times d}$*, has invariant distribution* $\Pi(\cdot)$*.*

Proof. This theorem can be proven using standard Fokker-Planck equations see e.g. [38, 78] for a complete proof. □

One intuitive reason why (1.1.8) provides some effective mechanism for sampling from its equilibrium distribution $\Pi(x)$, is that the drift in its expression represents the direction of steepest ascent of the target distribution. From standard geometry arguments, $\nabla \log \Pi(X_t)$ is a vector perpendicular to the contour of Π at X; this is known as the direction of steepest ascent (see [3]). In practice, this should push the SDE process in the direction of the mode of the target distribution. It is important to notice that, if these dynamics could be implemented

[7] Which was originally proposed in [72]

exactly in continuous or discrete time, then, because they have the correct invariant distribution, a no accept/reject mechanism would be required (all moves would be accepted with probability one).

Most frequently, it is not possible to simulate directly from this SDE (apart from some very simple cases, for instance when Π is Gaussian). As we will show in the next section, for computational purposes all such SDEs must be discritized in time, so invariance with respect to Π will be lost and a MH correction must be applied. Nevertheless, even with this approximation, the proposals that the Langevin SDE produces should still be very effective for sampling Π. The simplest discretization method is the explicit Euler scheme. Indeed, using this scheme in (1.1.8) for a mesh $h > 0$ gives the following proposed transition:

$$Y = X + \tfrac{1}{2}\,\mathcal{C}\,\nabla \log \Pi(X)h + \sqrt{\mathcal{C}}\sqrt{h}\,\xi. \tag{1.1.9}$$

with $\xi \sim N(0, I)$. Here, both h and $\mathcal{C}$ should be viewed as tuning parameters for the user to specify. So, we have now specified a proposal distribution $Q(\cdot \mid \cdot)$, based on the Euler approximation of the Langevin SDE. By construction, we have that:

$$Q(Y \mid X) \sim N(X + \tfrac{1}{2}\mathcal{C}\nabla \log \Pi(X)h, \mathcal{C}h). \tag{1.1.10}$$

Since, the proposal is just a linear transformation of a random normal variable, we can now plug in this proposal in the main MH framework to obtain MALA.

Both RWM and MALA are examples of 'local-move' algorithms, as typically a small enough h will be chosen (especially in moderate to high dimension sizes) to deliver good-enough acceptance probabilities. A common issue with such methods is their 'random walk behaviour', that is: one typically needs $1/h$ steps to explore the state space even if the standard deviation of each step is in fact $\mathcal{O}(\sqrt{h})$.

We expand on the MALA algorithm and present now the Hybrid Monte Carlo (HMC)[8] method which is a fairly unique example of a non-local method that generates large steps in the state space. To simplify our exposition we consider the case where the target distribution is defined by its density in the form:

$$\Pi(x) \propto \exp\{-\Phi(x)\}\ , \tag{1.1.11}$$

for some differentiable function $\Phi : \mathbb{R}^d \mapsto \mathbb{R}$. The development of HMC can be summarised into three separate components:

[8] see [34]

i) a Hamiltonian flow,

ii) a numerical integrator, and

iii) an accept/reject rule.

HMC expands the phase space of the target distribution by adding an auxiliary variable v of the same size as x such that we have the joint distribution on (x, v):

$$\Pi(x, v) \propto \exp\{-H(x, v)\} \tag{1.1.12}$$

where

$$H(x, v) = \tfrac{1}{2}\langle v, Mv\rangle + \Phi(x) \tag{1.1.13}$$

for a positive-definite matrix $M \in \mathbb{R}^{d\times d}$. The function $H = H(x, v)$ is known as the Hamiltonian ‘Energy’ function. The auxiliary variable v should be thought of as the ‘velocity’ variable, the original variable x is the ’location’ variable and M as the ‘mass’ matrix to be specified by the user. So, in this setting, $\Phi(x)$ is the ‘potential’ energy and $\frac{1}{2}\langle v, Mv\rangle$ the ‘kinetic’ energy. We can use the standard Hamiltonian equations to describe the evolution in time of the above system within an energy-preserving environment. In particular, we introduce a time index $t \geq 0$, so that the Hamiltonian equations are as follows:

$$\begin{aligned}\frac{dx}{dt} &= M^{-1}\frac{\partial H}{\partial v} = v,\\ M\frac{dv}{dt} &= -\frac{\partial H}{\partial x} = -\nabla\Phi(x).\end{aligned} \tag{1.1.14}$$

These equations are simply an application of Newton’s first law of motion. The Hamiltonian differential equations give rise to a semigroup solution operator Ξ^t which maps:

$$(x(t), v(t)) = \Xi^t(x(0), v(0)) \tag{1.1.15}$$

and has three important properties[9]:

1. Conservation of Energy: $H(\Xi^t(x(0), v(0))) = H((x(0), v(0)))$.

2. Conservation of Volume: $\Xi^t(dx, dv) = (dx, dv)$.

3. Time Symmetricity: $\Xi^t(x(t), -v(t)) = (x(0), -v(0))$.

Reminiscent to the MALA case, where it is possible to apply the Langevin dynamics perfectly, the transition from the current position $x(0) = x$ to $x(t)$ using

[9] Which will be relevant when defining HMC, see [7]

a sampled $v(0) \sim N(0, M^{-1})$ from its marginal distribution, would give rise to a Markov transition in x-space that preserves $\Pi(x)$. Hence, no accept/reject rule would be needed to obtain Markov dynamics with the correct equilibrium distribution. We are aware, though, that in practice we cannot typically obtain analytic solutions for the Hamiltonian equations.

As with MALA, we must resort to the use of a time-discretization method. The integrator of choice here is the so-called leapfrog integrator, since this scheme maintains both the time reversibility and conservation of volume properties of the original Hamiltonian equations[10]. The leapfrog integrator splits the Hamiltonian dynamics (1.1.14) into two steps. Consider separating the Hamiltonian energy function $H(x, v)$ into two parts: H_1 the potential energy and H_2 the kinetic energy. That is:

$$H = H_1 + H_2, \quad H_1 = \Phi(x), \quad H_2 = \tfrac{1}{2}\langle v, Mv\rangle. \tag{1.1.16}$$

Now we look at Hamiltonian flows Ξ_1^t and Ξ_2^t as arising from applying Hamiltonian equations to each of the energies H_1 and H_2 separately. That is, we have the differential equations:

$$\frac{dx}{dt} = M^{-1}\frac{\partial H_1}{\partial v} = 0, \quad M\frac{dv}{dt} = -\frac{\partial H_1}{\partial x} = -\nabla\Phi(x), \tag{1.1.17}$$

and

$$\frac{dx}{dt} = M^{-1}\frac{\partial H_2}{\partial v} = v, \quad M\frac{dv}{dt} = 0. \tag{1.1.18}$$

Now, it follows that these two separate flows can be solved explicitly giving:

$$\Xi_1^t(x, v) = (x, v - tM^{-1}\nabla\Phi(x)), \quad \Xi_2^t(x, v) = (x + tMv, v). \tag{1.1.19}$$

One complete step of the leapfrog integration involves a half step of Ξ_1, a full step of Ξ_2, followed by another half step of Ξ_1. That is, we have the synthesized operator:

$$\Psi_h = \Xi_1^{\frac{h}{2}} \circ \Xi_2^h \circ \Xi_1^{\frac{h}{2}}. \tag{1.1.20}$$

Ψ_h will now be applied for a number of times, say, $I = \lfloor \frac{T}{h} \rfloor$, so that the discretized dynamics will approximate the Hamiltonian flow on the time interval $[0, T]$, for some time horizon[11] $T > 0$. Then we obtain the following synthesis:

$$(x_T, v_T) = \Psi_h(\Psi_h(\Psi_h \dots \Psi_h(x_0, v_0) \dots)), \quad \text{applied } \lfloor T/h \rfloor \text{ times.}$$

[10] For a proof, see e.g. [60]

[11] This is a free parameter to be specified by the user

We will refer to this synthesis as the following mapping:

$$(x_T, v_T) = \Psi_h^T(x_0, v_0).$$

We can observe in Figure 1.1 a graphical illustration of the the leapfrog integration:

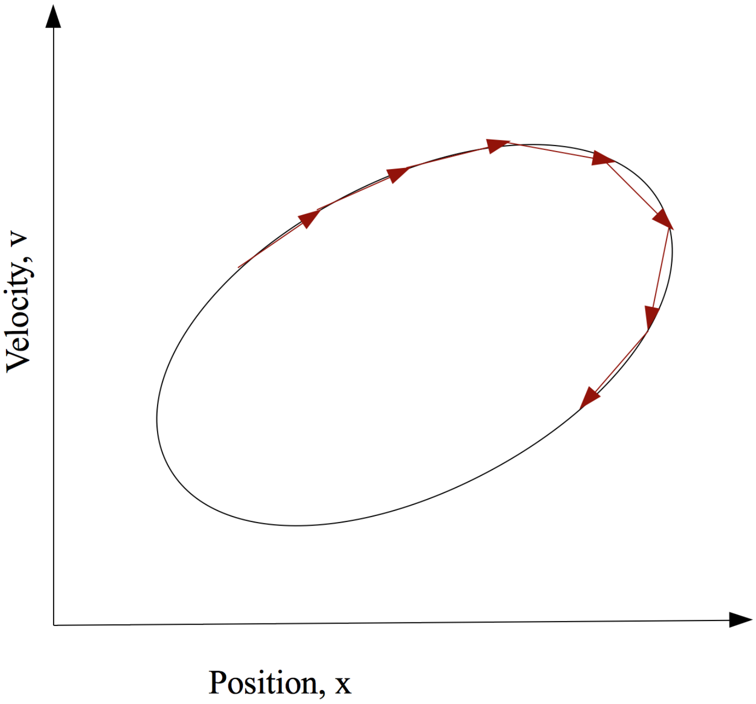

Figure 1.1: Graphical illustration of a synthesis of leapfrog steps.

each arrow in the graph represents a single leapfrog step Ψ_h, whereas the end point represents $\Psi^{(T)}$ for a chosen time horizon T. We can also see in the graph, how the terminal position of the algorithm ends up and how it depends on the size of each leapfrog step h and the time horizon T. Both parameters are user-specified, and some fine tuning is required, since for instance: if T is too small the algorithm may exhibit random walk behaviour whereas if it is too big the Hamiltonian trajectory may double-back on itself wasting computational resources. Another remark motivated by Figure 1.1 , is that the mapping Ψ_T is exploring a single contour which is linked to the initial choice of velocity v_0. The final algorithm, in fact, explores many contours as at each step it will sample a new $v_0 \sim N(0, M^{-1})$ before applying the leapfrog mapping $\Psi_h^{(T)}$ step of the algorithm.

One topic of debate may be the choice of the mass matrix M. If the target was a Gaussian measure $N(0, \Sigma)$ for some covariance matrix Σ, then an optimal choice is to set the mass function equal to the inverse of the covariance function i.e. $M = \Sigma^{-1}$. Intuitively, this means that areas of low variance will be assigned a large mass and vice-versa. We can think about the effect of this choice in terms of the corresponding HMC dynamics for describing the movement of particles in the phase space. Since the auxiliary variable v is interpreted as velocity, particles (by the conservation of momentum) will move slower in areas of low variance and faster in those of high variance. This means that the x-trajectory, which can be thought as the location of particles, will explore better the state space of the target distribution. Probabilistically, this is equivalent to transforming the target distribution to a product of d univariate $N(0,1)$ Gaussians. For a non-Gaussian target, it still makes sense to try to adjust M to the inverse of the covariance of the target distribution.

```
Standard HMC:
  (i) Assume current position $x^{(n)} = x_0$. Sample $v_0 \sim N(0, M^{-1})$.
  (ii) (a) update $v_{h/2} = v_0 - \frac{h}{2} M^{-1} \nabla\Phi(x_0)$;
       (b) update $x_h = x_0 + h v_{h/2}$;
       (c) update $v_h = v_{h/2} - \frac{h}{2} M^{-1} \nabla\Phi(x_h)$;
       (d) Repeat steps $(a) - (c)$ over $\lfloor \frac{T}{h} \rfloor$ times.
  (iii) Set $x^{(n+1)} = x_T$ with probability $1 \wedge \exp\{H(x_0, v_0) - H(\Psi_h^T(x_0, v_0)\}$,
        otherwise set $x^{(n+1)} = x_0 = x^{(n)}$.
  (iv) Repeat for $n = 1, 2, 3, \ldots$.
```

Table 1.1: Specification of standard HMC.

Table 1.1 presents the standard HMC algorithm. Notice that the MH acceptance probability– that corrects the discretized dynamics so that the final method has the correct invariant distribution– involves the changes in the energy function between the initial configuration in the phase space and the final configuration after synthesizing the leapfrog steps. For completeness we present, in Theorem 1.1.2 below, a simple proof that this particular acceptance probability provides a correct algorithm[12]. Below we denote by $\mathcal{P}_x$ the projection on the x component.

Theorem 1.1.2. *Consider any one-step mapping Ψ_h with the following properties:*

- *it is time-reversible, that is, $\Psi_h(x_h, -v_h) = (x_0, -v_0)$.*
- *it is volume-preserving, that is, $\Psi_h(dx, dv) = (dx, dv)$.*

[12] Versions of this proof can be found in several works, for example in the seminal paper of [34]

Consider the composite map Ψ_h^T, that synthesizes $\lfloor T/h \rfloor$ applications of Ψ_h. Assume current position $x \in \mathbb{R}^d$ and sample $v \sim N(0, M^{-1})$. Consider the Markov transition so that the trajectory moves to $x' = \mathcal{P}_x(\Psi_h^T(x,v))$ according to the acceptance probability:

$$\alpha(x,v) = 1 \wedge \exp\{H(x,v) - H(\Psi_h(x,v))\}$$

otherwise it stays at x (so $x' = x$). Then, the Markov transition $x \mapsto x'$ has invariant distribution Π.

Proof. To simplify the presentation of the formulae, we omit the subscript h from operators Ψ_h and Ψ_h^T. Assume, $(x_0, v_0) \sim \exp\{-H(x,v)\}$ and $(x_T, v_T) = \Psi^{(T)}(x_0, v_0)$ where Ψ is both volume-preserving and time-reversible. Then,

$$\begin{aligned} P((x_T, v_T)\epsilon(dx,dv)) &= P\big((x_0,v_0)\epsilon(dx,dv) \bigcap \text{reject move}\big) \\ &\quad + \overbrace{P\big((x_0,v_0)\epsilon\Psi^{-1}(dx,dv) \bigcap \text{accept move}\big)}^{B} \\ &= \exp\{-H(x,v)\}dxdv \cdot (1 - 1 \wedge \exp\{H(x,v) - H(\Psi(x,v))\}) \qquad (1.1.21) \\ &\quad + \underbrace{\int_{\Psi^{-1}(dx,dv)} \exp\{-H(r,w))\}drdw}_{I} \cdot \Big(1 \wedge \exp\{H(\Psi^{-1}(x,v)) - H(x,v)\}\Big). \end{aligned}$$

Now, making the change of variables $(r', w') = \Psi(r,w)$ within the integral I defined above together with the volume preservation of Ψ gives that:

$$I = \int_{dxdv} \exp\{-H(\Psi^{-1}(r,w))\}drdw \equiv \exp\{-H(\Psi^{-1}(x,v))\}dxdv.$$

Therefore, continuing from (1.1.21) we have that:

$$\begin{aligned} B = dxdv(\exp\{-H(\Psi^{-1}(x,v))\} \wedge \exp\{-H(x,v)\}) = \\ \int_{dxdv} \exp\{-H(\Psi^{-1}(r,w))\} \wedge \exp\{-H(r,w)\}drdw \\ = -\int_{dxd(-v)} \exp\{-H(\Psi^{-1}(r,-w))\} \wedge \exp\{-H(r,w)\}drdw, \end{aligned}$$

where in the last equation we have applied the change of variables $w \leftrightarrow -w$ and used the fact that $H(r,w) \equiv H(r,-w)$. Because of the time-reversibility property of Ψ and the fact again that $H(x,v) \equiv H(x,-v)$, we obtain that:

$$B = \exp\{-H(\Psi(x,v))\} \wedge \exp\{-H(x,v)\}dxdv \equiv \\ \exp\{-H(x,v)\}dxdv \cdot 1 \wedge \exp\{H(x,v) - H(\Psi(x,v))\}.$$

Plugging this final expression into (1.1.21) gives us:

$$P((x_T, v_T)\epsilon(dx, dv)) = \exp\{-H(x,v)\}dxdv.$$

Thus, marginally we have that $x_T \sim \exp\{-\Phi(x)\}$, and this completes the proof.

□

1.2 Introduction to Diffusion Processes

The target distributions of interest for our advanced algorithms are defined on the pathspace of diffusion processes, i.e. solutions of Stochastic Differential Equations (SDEs). Thus, the distributions of interest are defined on the infinite-dimensional Hilbert space of squared integrable paths, on some interval of interest $[0,\ell]$ for path-length $\ell > 0$. Thus, we denote this separable Hilbert space as follows:

$$\mathcal{H} = L^2([0,\ell], \mathbb{R}).$$

A big challenge here is the high-dimensionality of the state space (in theory infinite-dimensional, in practice some finite-dimensional approximation, say on $\mathbb{R}^N$ is used, for some large enough $N \geq 1$). We cover here briefly some of the basics aspects on the theory of SDEs[13].

The most basic diffusion process is known is the Wiener process (or Brownian motion), which is defined as follows: Let W_t, $t \geq 0$, be a Wiener process. Then this process is uniquely specified by the following properties:

1. $W_0 = 0$;

2. With probability 1, the sample path $t \to W_t$ is everywhere continuous and nowhere differentiable;

3. W_t has independent increments on disjoint time intervals, with distribution for $0 \leq s < t$, $W_t - W_s \sim N(0, t-s)$.

[13] The reader is referred e.g. in [65] for a more rigorous approach and extensive details

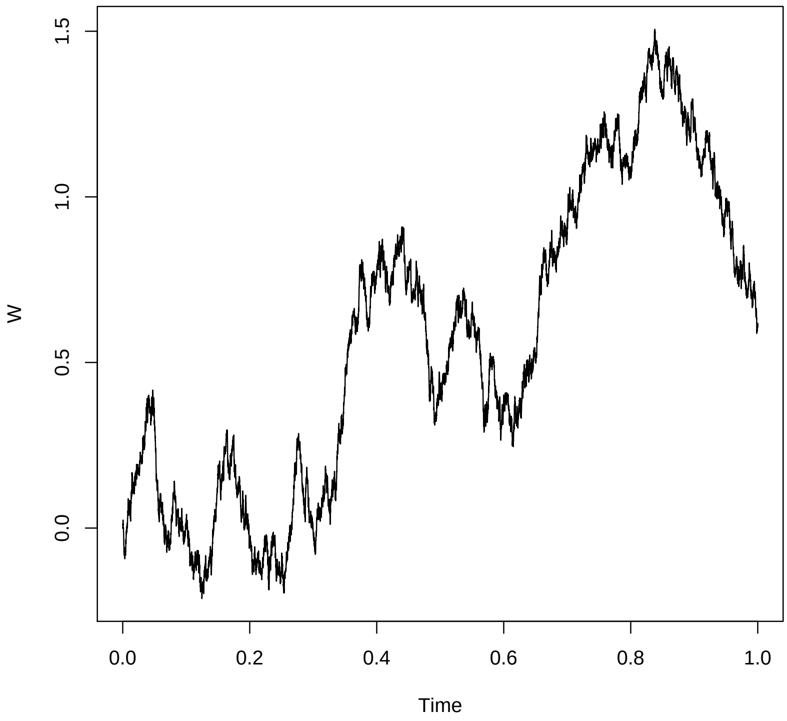

Figure 1.2: An example of a Wiener process sample path.

In Figure 1.2 we can see an entire Wiener process W_t sample path plotted against time.

The Wiener process forms the basis for most of the stochastic processes used in this thesis. The standard method to construct a continuous-time diffusion process is by using a SDE that resembles an ordinary differential equation which, apart for the involvement of a stochastic component, is of the following general form:

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t, \tag{1.2.1}$$

for some drift function $b : \mathbb{R} \mapsto \mathbb{R}$ and diffusion coefficient $\sigma : \mathbb{R} \mapsto \mathbb{R}$. The drift and the diffusion coefficient must satisfy the standard regulatory condition to guarantee the existence and uniqueness of a global solution for (1.2.1)[14]. Typically, it is required that b is Lipschitz with a linear growth condition; similar conditions must hold for σ. Notice that one can trivially extend (1.2.1) to a time-inhomogeneous setting by allowing b and σ to also depend on time; for simplicity, we have introduced the SDE in a time-homogeneous context.

[14] See e.g. [65] for details

The solution X_t of (1.2.1) is typically impossible to obtain analytically. In a numerical setting, one can use a multitude of approximate methods[15] for the simulation of sample paths of X_t. For the purposes of this thesis we will limit ourselves to the simplest of methods, i.e. the standard Euler-Maruyama scheme. When applied to (1.2.1), the Euler scheme requires a discretized grid of times $0 = t_0 < t_1 < \cdots t_N$, and provides the sampling iteration:

$$X_{t_i} = X_{t_{i-1}} + b(X_{t_{i-1}})\Delta_i + \sigma(X_{t_{i-1}})(W_{t_i} - W_{t_{i-1}}), \tag{1.2.2}$$

where $\Delta_i = t_i - t_{i-1}$. This scheme allows for easy generation of sample paths when used on a personal computer. The size of the approximation error depends on the smallness [16] of Δ_t.

Another important process, which we will be involved in when discussing advanced algorithms, is the so-called Brownian bridge. In a similar way that random variables can be conditioned on events or on other random variables, this can also happen for stochastic process. In particular, given the Wiener process W_t, we can condition it to start at point x at time 0 and to end at point y at time ℓ. Then, it can be shown either from first principles ([52]) in the Brownian motion case or by using Doob h-transform (this provides bridges for general SDEs, [73]) that the process $X_t = W_t \mid W_0 = x, W_\ell = y$ can be defined as the solution of the SDE:

$$dX_t = \frac{y - X_t}{\ell - t}dt + dW_t, \quad X_0 = x. \tag{1.2.3}$$

This is precisely the definition of a Brownian bridge. A number of sample paths of a Brownian bridge starting and finishing at 0 are shown in Figure 1.3 to give a visual impression of Brownian bridge characteristics. Conditioned diffusion bridges, such as the Brownian bridge, are later on shown to be useful for missing data problems where we are interested in filling the gaps between observed data points. Notice that, typically, there is not a simple explicit SDE expression for general conditioned diffusion processes, so their sampling is a non-trivial problem.

1.2.1 SDE as Change of Measure from Gaussian Law

We now describe some expression for the distribution of the solution of an SDE, and that later on will be used in the development of advanced MCMC methods

[15] See e.g. [53] and the references therein

[16] Typically, and under regulatory conditions, the weak error is $\mathcal{O}(\Delta)$ with $\Delta = \sup_i \Delta_i$, with the strong error being $\mathcal{O}(\Delta^{1/2})$, see [53] for proofs and details

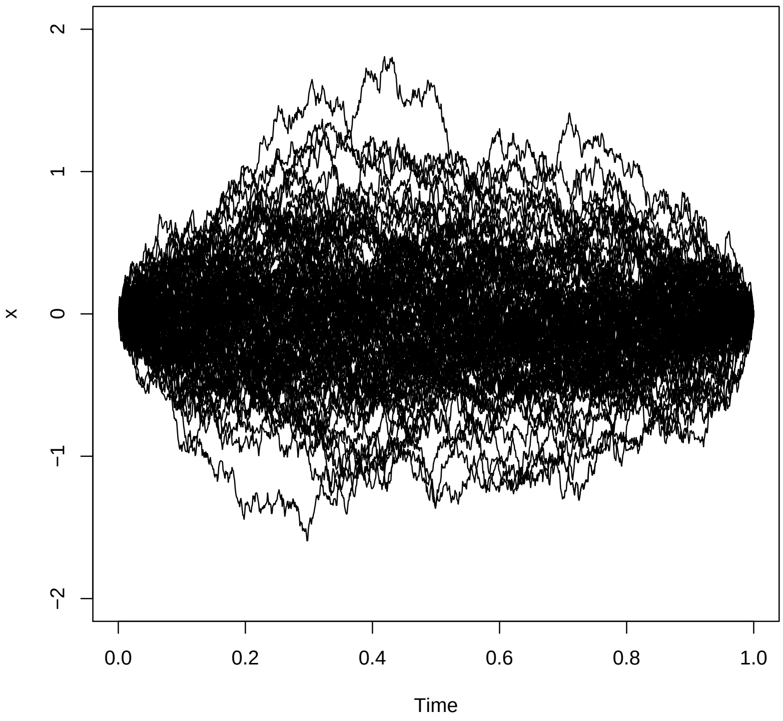

Figure 1.3: Brownian bridge sample paths starting at 0 and ending at 0.

for sampling from such SDEs and related processes. As we have seen in (1.1.5), the Metropolis-Hastings algorithm requires the probability density function of the target and proposal distribution. In finite dimensions such densities are typically obtained with respect to the standard Lebesque measure on $\mathbb{R}^d$. In the infinite-dimensional Hilbert space $\mathcal{H}$ the role of the reference measure will be taken by the law of the Brownian motion (sometimes called the Wiener measure).

Consider the following SDEs:

$$\Pi : \quad dX_t = b(X_t)dt + dW_t, \quad t \in [0, \ell] \tag{1.2.4}$$

$$\Pi_0 : \quad dX_t = dW_t, \quad t \in [0, \ell]. \tag{1.2.5}$$

We have denoted by Π and Π_0 as the probability measures on $\mathcal{H}$ determined by the above two processes. We are interested in obtaining here the density of Π with respect to the Gaussian law[17] Π_0. Using simple terms: assume we observe a sample path X_t, $0 \leq t \leq \ell$, and we want to know if it belongs to the diffusion process (1.2.4) or to the Brownian motion process (1.2.5). Then, the density $d\Pi/d\Pi_0$ between

[17] This is also called the Radon-Nikodym derivative $d\Pi/d\Pi_0$, see e.g. [10]

the relevant probability measures provides essentially the likelihood of the given sample path coming from Π versus the path coming from Π_0. Hence, the reference measure we are using here, Π_0, is simply the Wiener measure. A general result here is that, under some regulatory conditions on b, any process of the form (1.2.4) is absolutely continuous with respect to the relevant Wiener measure, and the probability density function $d\Pi/d\Pi_0$ is provided by the Girsanov's theorem[18]:

$$\frac{d\Pi}{d\Pi_0}(X) = \exp\left[\int_0^\ell b(X_t)dX_t - \frac{1}{2}\int_0^\ell b^2(X_t)dt\right]. \tag{1.2.6}$$

So far we have restricted our attention to unit diffusion coefficients. In the general case, we will be interested on an SDE of the form:

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t, \tag{1.2.7}$$

with $\sigma(X_t)$ being a non-constant function of X_t. It is well-known in the theory of SDEs, that different diffusion coefficient functions lead to singular diffusion probability laws, thus, we cannot expect to use the Wiener measure as a reference measure for (1.2.7). The Girsanov density defined below generalizes for the case of general diffusion coefficient. In this case, we look at the processes:

$$\Pi: \quad dX_t = b(X_t)dt + \sigma(X_t)dW_t, \tag{1.2.8}$$

$$\Pi_0: \quad dX_t = \sigma(X_t)dW_t, \quad t \in [0,\ell], \tag{1.2.9}$$

in which case we have the Girsanov density being equal to:

$$\frac{d\Pi}{d\Pi_0}(X) = \exp\left[\int_0^\ell \frac{b(X_t)}{\sigma(X_t)}dX_t - \frac{1}{2}\int_0^\ell \frac{b^2(X_t)}{\sigma^2(X_t)}dt\right] =: \mathcal{G}(X). \tag{1.2.10}$$

In what follows below we focus on conditioned diffusion processes. In the previous section we encountered ‘diffusion bridges’, that is, diffusion processes that have been conditioned to start and end at some specific points. Assume now that we are given a diffusion process X_t, defined by an SDE with general diffusion coefficient as in (1.2.8). Then, we define the corresponding target diffusion bridge starting at point x and ending at point y at time ℓ in the standard way as:

$$X_t \mid X_0 = x,\ X_\ell = y. \tag{1.2.11}$$

[18] See e.g. [65]

We are interested in sampling from such diffusion bridges. Notice, that standard discretization schemes used in the unconditional setting are not useful when conditioning is on the terminal position of the SDE, this is because it is typically impossible to obtain the SDE expression for the conditioned[19]. Since, most typically, we will not have an analytical expression for the bridge, we will look at MCMC methods for solving this sampling problem.

For this, we require a simpler process that generates proposals for the target diffusion bridge, and thus, also the related density between probability measures of two diffusion processes conditioned to end at the same point. This will involve coming up with a new version of Girsanov's theorem that will apply to diffusion bridges, this is because the standard Girsanov's theorem given in (1.2.10) involves unconditional dynamics. Using Bayes' rule for the Π and Π_0 as defined in (1.2.8) and (1.2.9) one can obtain the following expression[20]:

$$\frac{d\Pi}{d\Pi_0}(X|X_0 = x, X_\ell = y) = \frac{\Pi(X_\ell = y|X)\Pi(dX)/\Pi(X_\ell \in dy)}{\Pi_0(X_\ell = y|X)\Pi_0(dX)/\Pi_0(X_\ell \in dy)}. \tag{1.2.12}$$

We now briefly consider each term individually to try to gain some intuition about its significance. The first ratio $\Pi(X_\ell = y|X)/\Pi_0(X_\ell = y|X)$ can simply be replaced by 1, as we are considering sample paths X which are constrained to have $X_\ell = y$ by definition. This involves the marginal distribution at time ℓ of the two diffusion processes. The next fraction $\Pi(dX)/\Pi_0(dX)$ corresponds to the unconditional Girsanov density given in (1.2.10). Summarising, we have that:

$$\frac{d\Pi}{d\Pi_0}(X|X_\ell = y) = \mathcal{G}(X) \times \frac{\Pi_0(X_\ell \in dy|X_0 = x)}{\Pi(X_\ell \in dy|X_0 = x)}.$$

Notice now that $\Pi(X_\ell \in dy|X_0 = x)$ corresponds to a transition probability for the unconditional Markov process (1.2.8). We denote the transition density of (1.2.8) as:

$$\Pi(X_\ell \in dy|X_0 = x) = p(\ell; x, y)dy. \tag{1.2.13}$$

Then, we make a similar definition for the reference SDE in (1.2.9). That is, we set:

$$\Pi_0(X_\ell \in dy|X_0 = x) = q(\ell; x, y)dy. \tag{1.2.14}$$

Bringing everything together, we have the following Girsanov density for diffusion bridges:

[19] A notable exception as we have seen, is the case of the Brownian bridge

[20] Many times we suppress reference to the initial position $X_0 = x$ as it is easy to enforce

$$\frac{d\Pi}{d\Pi_0}(X|X_\ell = y) = \mathcal{G}(X) \times \frac{q(\ell; x, y)}{p(\ell; x, y)}. \tag{1.2.15}$$

We have achieved an intuitive derivation of the conditional version of Girsanov's theorem, which fits the practical algorithmic investigations of the thesis, while avoiding stating technical assumptions and mathematical conditions[21]. A timely comment here is that the fraction $q(\ell; x, y)/p(\ell; x, y)$ is a constant when sampling from the conditional distribution of X, thus, it will not be involved when setting up MCMC algorithms as it cancels out when calculating the relevant MH acceptance probability.

Now we focus on the practical problem of sampling from the diffusion bridge, specified via the dynamics in (1.2.8) and the constraints in (1.2.11). The reference measure $\Pi_0(X|X_\ell = y)$ used in (1.2.15) is not useful for giving candidate paths, generally it is not possible to generate sample paths from that distribution. Instead, following e.g. [44] we build an alternative diffusion bridge that can be easily simulated. Earlier on we described the Brownian bridge and we wrote down the equivalent SDE expression of it, i.e. $W_t \mid W_\ell = y$ which is described by the SDE equation in (1.2.3). Notice that the drift function in the expression is:

$$b^*(X_t, t) = \frac{y - X_t}{\ell - t}, \tag{1.2.16}$$

and it ensures that the diffusion process is 'pushed' towards the terminal position y as $t \to \ell$. This motivates us to use the following SDE when generating candidate paths for the target diffusion bridge:

$$dX_t = b^*(X_t, t)dt + \sigma(X_t)dW_t, \tag{1.2.17}$$

so that, due to the particular form of drift, X_t will indeed be a bridge that ends on $X_\ell = y$. This is a useful result. For instance, if we were to carry out an Independent Sampler, then (1.2.17) could be used as a proposal. We can also find the density of our target distribution with respect to this proposal SDE in (1.2.17). Let $\Pi^{x,y}$ be the probability measure corresponding to the diffusion bridge of interest and let $Q^{x,y}$ be the probability measure of the reference diffusion process above, that is:

$$\Pi^{x,y} : \quad dX_t = b(X_t)dt + \sigma(X_t)dW_t, \quad X_0 = x, \;\; X_\ell = y, \tag{1.2.18}$$

$$Q^{x,y} : \quad dX_t = b^*(X_t, t)dt + \sigma(X_t)dW_t, \quad X_0 = x, \;\; X_\ell = y \;. \tag{1.2.19}$$

[21] For a complete rigorous treatment the reader is referred to [30]

We define the following function:

$$\begin{aligned}\mathcal{D}(X) = &\exp\Bigg[\int_0^T \frac{b(X_t)}{\sigma(X_t)^2}dX_t - \frac{1}{2}\int_0^T \frac{b^2(X_t)}{\sigma(X_t)^2}dt \\ &\qquad - \int_0^T \frac{b^*(X_t)}{\sigma(X_t)^2}dX_t + \frac{1}{2}\int_0^T \frac{(b^*(X_t,t))^2}{\sigma(X_t)^2}\Bigg].\end{aligned} \tag{1.2.20}$$

which resembles a Girsanov density between the unconditional original SDE and the reference one in (1.2.17). We can now use Bayes' theorem from first principles as before, or follow the works in [30] and [18] to obtain that:

$$\begin{aligned}\frac{d\Pi^{x,y}}{dQ^{x,y}}(X) &= \frac{\Pi(dX \mid X_\ell = y)}{Q^{x,y}(dX)} \\ &= \frac{\Pi(dX)}{Q^{x,y}(dX)p(\ell;x,y)} = \frac{1}{p(\ell;x,y)} \times \mathcal{D}(X).\end{aligned} \tag{1.2.21}$$

This density can now be used, say within the context of an Independent Sampler, to determine the acceptance probability.

1.2.2 Path Transformations

The density functions found above can provide an Independent Sampler algorithm targeting the diffusion bridge of interest. However, for some of the advanced MCMC algorithms to be presented in the sequel, it will be necessary that the reference measure is Gaussian; in general, this is certainly not the case for above proposal $Q^{x,y}$, when σ is non-constant. Another issue is that when there are unknown parameters present in the diffusion coefficient function to be inferred by some observations, then the reference measure $Q^{x,y}$ will also depend on these parameters, and this can provide unsatisfying singularities when setting up MCMC methods. Indeed, by using clever use of a transformation we can decouple the latent parameters from the reference measure by mapping $Q^{x,y} \to \Pi_0$. We use this concept later on in the thesis, specifically in chapter 5 and a bit in section 3.3. To enforce Gaussianity for the reference measure, we are obligated to transform the target bridge into a process whose law is indeed absolutely continuous with respect to a Gaussian measure[22]. There are two main methods in the literature for achieving such an effect, which we briefly present next.

Consider the target distribution $\Pi^{x,y}$ in (1.2.18). A standard direction to obtain a 1-1 transformation to produce a modified stochastic process Y_t with unit

[22] Typically a Brownian bridge or a Brownian motion

diffusion coefficient is via the Lamperti transformation. Assuming X_t is defined by a SDE (1.2.18), then, we define the new process Y_t as follows:

$$Y_t \equiv \gamma(X_t) := \int_x^{X_t} \frac{du}{\sigma(u)}, \tag{1.2.22}$$

Then, with a straightforward application of Itô's lemma, we obtain that Y_t will itself solve the following SDE:

$$dY_t = \mu_Y(Y_t)dt + dW_t, \quad Y_0 = 0,\ Y_\ell = \gamma(y). \tag{1.2.23}$$

for drift function:

$$\mu_Y(Y_t) = \frac{b(\gamma^{-1}(Y_t))}{\sigma(\gamma^{-1}(Y_t))} - \frac{1}{2}\sigma(\gamma^{-1}(Y_t)). \tag{1.2.24}$$

One can now easily obtain the density of the distribution of Y with respect to a Brownian bridge as given by the conditional version of the Girsanov density in (1.2.15), after appropriately adjusting to the new starting and ending points. This is precisely the Lamperti transformation, widely referenced in the literature for SDEs[23]. A diffusion process that can be transformed to one with unit-diffusion coefficient with Lamperti's transformation (1.2.22), and its generalization to a non-scalar case, is called reducible. For one dimensional diffusion processes such as (1.2.7), it is, in principle, always possible to use the transformation (1.2.22). However, multivariate diffusions are not always reducible. A negative and commonly used example is the standard bivariate stochastic volatility model[24].

Because sometimes it is not possible to use the Lamperti transformation in non-scalar SDEs, we now look at another transformation considered in the literature. This is the Golightly-Wilkinson transformation[25], and which only requires the existence of an inverse, $\sigma^{-1}(X_t)$ for the diffusion coefficient $\sigma(X_t)$. This is a much weaker condition than the conditions of reducibility required by the Lamperti transformation. It is important to note that the Golightly-Wilkinson transformation as specified in this section is only relevant for diffusion bridges. Later on in the thesis, in section 4.1, we provide a similar mapping to Golightly-Wilkinson that is relevant to cases beyond diffusion bridges.

Hence, we can consider the SDE in (1.2.17) as a mapping which projects the

[23] A very thorough explanation of the transformation can be found in [2] and the very straightforward derivation using Itô's formula can be found in [36]

[24] The work in [2] presents necessary and sufficient conditions for identifying reducible diffusions in general dimension

[25] The Golightly-Wilkinson transformation was first seen in [44] and is based on the bridged SDE (1.2.17)

Brownian motion path W_t onto the actual bridge path X_t. That is, the very definition of the SDE (1.2.17) gives rise to a mapping, say $\Psi(\cdot)$ such that:

$$X = \Psi(W). \tag{1.2.25}$$

Now recall that the actual target SDE is the bridge $\Pi^{x,y}$ defined in (1.2.18). With X now referring to a path of this target SDE, we will transform it into:

$$\tilde{X} = \Psi^{-1}(X) \ . \tag{1.2.26}$$

This is precisely the Golightly-Wilkinson transform. That is, we have defined the connection between the target X and the transformation $\tilde{X}$:

$$dX_t = \frac{y - X_t}{\ell - t} dt + \sigma(X_t) d\tilde{X}_t. \tag{1.2.27}$$

The idea is that, X being a path of the reference measure $Q^{x,y}$, then $\Psi^{-1}(X)$ would deliver a Brownian motion. Now that X is a path from $\Pi^{x,y}$, the transform $\Psi^{-1}(X)$ does not give a Brownian motion, but a process that will be absolutely continuous with respect to Brownian motion, due to the absolute continuity of the original measures $\Pi^{x,y}$ and $Q^{x,y}$. Indeed, from standard results on 1-1 transformations of probability measures, we can find the relevant density. Let $\tilde{\Pi}^{x,y}$ denote the law of $\Psi^{-1}(X)$ for $X \sim \Pi^{x,y}$, and Π_0 the law of a standard Brownian motion on $[0, \ell]$. Then we have that:

$$\frac{d\tilde{\Pi}^{x,y}}{d\Pi_0}(\tilde{X}) = \frac{d\Pi^{x,y}}{dQ^{x,y}}(\Psi(\tilde{X}))$$

with the latter density being given in (1.2.21).

The Golightly-Wilkinson transformation will prove handy when we move onto more complex algorithms. In practice it will allow us to make proposals like $\tilde{X}_t$ which resemble brownian motion, and then transform it into X_t which looks similar to the target distribution and has a known density (1.2.21). Notice that the mapping $\tilde{X} \mapsto X$ obtained via (1.2.27) cannot be determined exactly, it will have to be found numerically. That is, we can construct an approximation using Euler's method (for some chosen step-size $\Delta t > 0$) that gives us:

$$\begin{aligned} X_{t+\Delta t} &= X_t + \frac{y - X_t}{\ell - t}\Delta t + \sigma(X_t)(\tilde{X}_{t+\Delta t} - \tilde{X}_t), \\ X_0 &= x. \end{aligned} \tag{1.2.28}$$

Equation (1.2.28) will be the one used in practice to obtain $X = \Psi(\tilde{X})$ when given $\tilde{X}$ or the inverse (this is why we need $\sigma(X)$ to be invertible) $\tilde{X} = \Psi^{-1}(X)$, when given X.

1.2.3 Independent Sampler: A Numerical Example

We will now apply the results from the previous Section to run an Independent sampler on the path space of diffusion processes. The Independent Sampler[26] (IS) is a very simple sampler and, as we have mentioned, is a specific version of the general Metropolis-Hastings algorithm. The name arises from the fact that each proposal is sampled independently from the current value. Assume that the target distribution is the diffusion bridge $\Pi^{x,y}$ in (1.2.18) for $\sigma \equiv 1$ (maybe after applying the Lamperti transformation). With a slight abuse of notation, we write $\Pi \equiv \Pi^{x,y}$ to simplify the expressions that follow. So, our proposal is basically the law of a Brownian bridge with the same starting and ending points; we call its law Π_0. Recall that the expression in (1.2.15) provides the density $d\Pi/d\Pi_0$. The algorithm works as follows:

1. Proposal $X' \sim \Pi_0$ is sampled from a simple Brownian bridge independently of current bridge X (see (1.2.3)).

2. If accepted, using the acceptance probability, then X' becomes the current value, otherwise the next position remains X.

We know that (1.1.5) is the acceptance probability for a MH on finite-dimensional spaces, with densities obtained with respect to a reference measure (typically the Lebesque measure). We need to make some rearrangement when working on the path space. For this, we take into account that this is an independent sampler with $Q(dx|X') = \Pi_0(dx)$, similarly $Q(dx'|X) = \Pi_0(dx')$ where Π_0 is the probability measure of the Brownian bridge process as already mentioned. Hence, the equation giving the acceptance probability (1.1.5) becomes:

$$\alpha(x, x') = \min\left(1, \frac{\Pi(dx')\Pi_0(dx)}{\Pi(dx)\Pi_0(dx')}\right).$$

Expressing this as a fraction of $\frac{\Pi(\cdot)}{\Pi_0(\cdot)}$ we get:

$$\alpha(x, x') = \min\left(1, \frac{\frac{\Pi(dx')}{\Pi_0(dx')}}{\frac{\Pi(dx)}{\Pi_0(dx)}}\right)$$

[26] Notice that a well-defined IS on the pathspace first appeared in [71]

Notice that this expression makes sense on the infinite dimensional pathspace, as we obtain the density $(d\Pi/d\Pi_0)$ from (1.2.15). That is, recalling the definition of $\mathcal{G}(x)$ in (1.2.15), now considered for the case of unit diffusion coefficient, we get the acceptance probability:

$$\alpha(x,x') = \min\left(1, \frac{\mathcal{G}(x')}{\mathcal{G}(x)}\right) . \tag{1.2.29}$$

As a simple numerical example of the methods described in this paper of the diffusion bridge we simulate an Ornstein-Uhlenbeck[27] diffusion process. This has the following SDE:

$$dX_t = r(\mu - X_t)dt + dW_t \ , \tag{1.2.30}$$

for some parameters $r > 0$, $\mu \in \mathbb{R}$. The mean of reversion μ affects where the diffusion process will 'gravitate around', while the speed of reversion r affects how closely the diffusion process follows the mean. This is a Gaussian process.

Now we are in a position to implement the IS algorithm on a personal computer. We have a method of sampling Brownian bridges $\Pi_0(\cdot)$ and a way of calculating the acceptance probability. We will be running the code for an IS with the following Ornstein-Uhlenbeck specification parameters: $r = 3$, $\mu = 4.6$, $X_0 = 3$, $X_\ell = 4$, $\ell = 1$. In Figure 1.4 we can observe a few diagnostic plots produced to judge the quality of the algorithm.

The first graph in figure 1.4 is a trace plot that corresponds to the middle point of all the diffusion processes being simulated. A good trace plot should show that it converges rapidly to the stationary distribution and that it has a good mixing. A trace plot which shows good mixing, traverses its posterior space rapidly, and it can jump from one remote region of the posterior to another in relatively few steps. Our graph shows that the algorithm has explored the region of the posterior very poorly: in its $10,000$ iterations only 0.8% of proposal moves were accepted. The second graph shows the autocorrelation function(ACF), we can judge the quality of an algorithm by the speed that the autocorrelation reaches zero. In this case it took a lag of about 350 to reach zero, which is not satisfactory. Overall we consider this algorithm to be a poor one. This simple example shows the need to develop better algorithms on the pathspace, which is the main theme of this thesis.

[27] More information about the Ornstein-Uhlenbeck process can be found in the original paper [82]

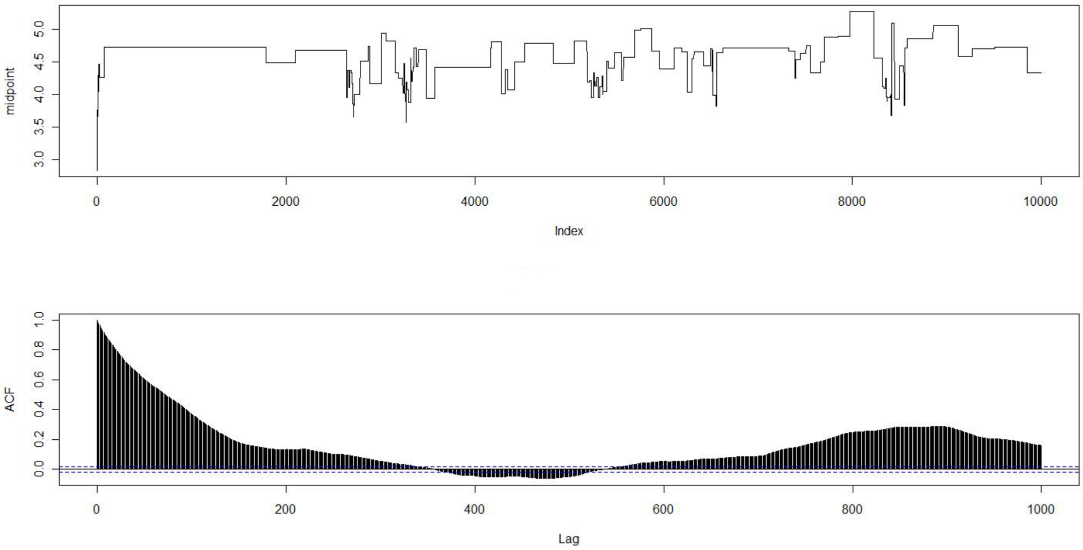

Figure 1.4: Diagnostic Plots for Independent Sampler targeting an O-U bridge. Top panel: Traceplot. Bottom Panel: Autocorrelation function.

1.3 A brief Note on Quadratic Variation

We provide some preliminary motivation for the pathspace algorithms, defined in the next Chapter, by looking at the so-called quadratic variation (see e.g. [65]) of a diffusion process. We assume that we are given the following SDE:

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t, \tag{1.3.1}$$

its quadratic variation is then determined as follows:

$$\langle X, X\rangle_t = \lim_{\Delta t_k \to 0} \sum_{t_k \le t}^{n} \left(X_{t_{k+1}} - X_{t_k}\right)^2 \equiv \int_0^t \sigma(X_s)^2 ds \tag{1.3.2}$$

where the discrete-time instances $0 = t_1 < t_2 < \cdots t_k < t_{k+1} \cdots$ with increment $\Delta t_k = t_{k+1} - t_k$ vanishing to 0. This limit can be shown to exist for general continuous diffusion processes under regulatory condition and in various convergence forms (see e.g. [52]). We will henceforth restrict our attention to the case where:

$$\sigma \equiv 1,$$

in this case, paths from the target SDE have quadratic variation $\langle X, X\rangle_t \equiv t$, with probability 1. As we have seen with the Independent Sampler, one can generate candidate paths for the target distribution (1.3.1) from a Brownian motion Π_0 (or a Brownian bridge if we had imposed some condition $X_\ell = y$).

Consider now a proposal mechanism of the $X' = aX + b\xi$ with $\xi \sim \Pi_0$ and some constants a, b, so that in this case the proposal X' is a linear combination of the current value X and a simple Brownian path ξ. This resembles the structure of a proposal used within a standard RWM algorithm. The choice of a, b will be critical when working on the infinite-dimensional pathspace, as naive choices can deliver proposals X' out of the domain of the target distribution, thus, having an acceptance probability of 0. We motivate this, by examining the choice of a, b that deliver paths with the correct quadratic variation. That is, we wish to have a proposal that has the same properties as the current value, hence, values a and b must be chosen so that X' has the same quadratic variation as X. Intuitively, it makes sense that we want to have proposals which have the same quadratic variation as our target distribution. Quadratic variation is a definitive property of SDE, if we were to have proposal that didn't meet this requirement the acceptance probability of the algorithm would converge to zero and the algorithm would break down.

Using the bilinearity of the quadratic variation we have that:

$$\begin{aligned}\langle X', X'\rangle_t &= \langle aX + b\xi, aX + b\xi\rangle_t \\ &= a^2\langle X, X\rangle_t + b^2\langle \xi, \xi\rangle_t + 2ab\langle X, \xi\rangle_t \\ &= (a^2 + b^2)t + 2ab\langle X, \xi\rangle_t.\end{aligned}$$

Notice that, due to the independency between X and ξ, it is a standard result (see e.g. [65]) that:

$$\langle X, \xi\rangle_t = 0.$$

Thus, clearly we need to have:

$$a^2 + b^2 = 1.$$

So, if we let $a = \rho \in (0, 1)$, then we are left with the following proposal:

$$X' = \rho X + \sqrt{1 - \rho^2}\,\xi \tag{1.3.3}$$

and X' will have the same quadratic variation as X.

Some comments here are in order: First, the standard RWM would use $a = b = 1$ and it would not work in this case since it would give the wrong quadratic variation. Second, proposal (1.3.3) has the important property that it preserves the reference Brownian motion (or Brownian bridge) measure. That is: if the target distribution was indeed a Brownian motion, then the acceptance probability should have been 1. Now that the target distribution is a general diffusion process, the acceptance probability will not be 1, but because of to the absolute continuity between the target and the Brownian motion, we expect that the acceptance probability is well-defined and non-zero (in contrast with the standard RWM).

1.4 Gaussian Measures on Hilbert Spaces

We summarize here some background material (see e.g. [25]) on Gaussian distributions on a separable Hilbert space $\mathcal{H}$ that will assist in the presentation of the later Chapters of this thesis. The Cameron-Martin space, $\mathcal{H}_0$, of the Gaussian law $\Pi_0 \equiv N(0,\mathcal{C})$ coincides with the image space of $\mathcal{C}^{1/2}$ and is formally defined below in this section. Essentially, $\mathcal{H}_0$ includes all elements of the Hilbert space which preserve the absolute continuity properties of Π_0 upon translation. This is made mathematically explicit via the following proposition:

Proposition 1.4.1. *If $T(x) = x + \mathcal{C}^{1/2}x_0$ for a constant $x_0 \in \mathcal{H}$ then Π_0 and $\Pi_0 \circ T^{-1}$ are absolutely continuous with respect to each other and with density:*

$$\frac{d\{\,\Pi_0 \circ T^{-1}\,\}}{d\Pi_0}(x) = \exp\left\{\langle x_0, \mathcal{C}^{-1/2}x\rangle - \tfrac{1}{2}|x_0|^2\right\}\,.$$

Proof. This is Theorem 2.21 of [25]. □

As we have already mentioned, for the diffusion pathspace we focus upon in this thesis, the target distribution $\Pi(dx)$ is defined on the Hilbert space of squared integrable paths $\mathcal{H} = L^2([0,\ell],\mathbb{R})$ (with appropriate boundary conditions) for some length $\ell > 0$. The centered Gaussian reference measure Π_0 corresponds to a Brownian motion (thus, boundary condition $x(0) = 0$) or a Brownian Bridge ($x(0) = x(\ell) = 0$). (Notice that the choice of 0-boundary conditions does not restrict the generality of the method, as the target with general boundary conditions can be transformed into one of 0-boundary conditions upon translating with the straight line that connects such general boundaries.)

The covariance operator is connected with the standard covariance function $c(u,v)$ of the Gaussian process via:

$$(\mathcal{C}f)(u) = \int_0^\ell c(u,v)f(v)dv\;, \quad f \in \mathcal{H}\;.$$

With this in mind, the covariance operators $\mathcal{C}^{bm}$, $\mathcal{C}^{bb}$ of the Brownian motion and

Brownian bridge respectively are as follows:

$$(\mathcal{C}^{bm}f)(u) = \int_0^\ell (u \wedge v)\, f(v)dv = u\int_0^\ell f(v)dv - \int_0^u \int_0^s f(v)dv\, ds \ ; \qquad (1.4.1)$$

$$\begin{aligned}(\mathcal{C}^{bb}f)(u) &= \int_0^\ell (u \wedge v - \tfrac{uv}{\ell})\, f(v)dv \\ &= \frac{u}{\ell}\int_0^\ell \int_0^s f(v)dv\, ds - \int_0^u \int_0^s f(v)dv\, ds \ . \qquad (1.4.2)\end{aligned}$$

Definition 1.4.1. *The Cameron-Martin spaces $\mathcal{H}_0^{bm}$ and $\mathcal{H}_0^{bb}$ of a Brownian motion and Brownian bridge respectively are analytically specified as follows*[28]*:*

$$\begin{aligned}\mathcal{H}_0^{bm} &= \big\{x : [0,\ell] \mapsto \mathbb{R} : \exists\ f \in L^2([0,\ell],\mathbb{R}) \textit{ such that } x(t) = \int_{[0,t]} f(s)ds\big\} \ ; \\ \mathcal{H}_0^{bb} &= \big\{x : [0,\ell] \mapsto \mathbb{R} : \exists\ f \in L^2([0,\ell],\mathbb{R}) \textit{ such that} \\ &\qquad\qquad x(t) = \int_{[0,t]} f(s)ds,\ x(\ell) = 0\big\} \ .\end{aligned}$$

The so-called Karhunen-Loève representation of the Gaussian law $N(0,\mathcal{C})$ will be used later on in the thesis. Analytically, considering the standard eigen-decomposition $\{\lambda_p, \phi_p\}_{p=1}^{\infty}$ of $\mathcal{C}$ so that $\mathcal{C}\,\phi_p = \lambda_p\,\phi_p$, we have that the (normalised) eigenfunctions $\{\phi_p\}_{p=1}^{\infty}$ constitute an orthonormal basis for the Hilbert space $\mathcal{H}$. In particular, for $x \sim N(0,\mathcal{C})$ we have the expansion:

$$x = \sum_{p=1}^{\infty} \langle x, \phi_p\rangle\, \phi_p = \sum_{p=1}^{\infty} x_p\, \phi_p = \sum_{p=1}^{\infty} \sqrt{\lambda_p}\, \xi_p\, \phi_p \ , \qquad (1.4.3)$$

where $\{\xi_p\}_{p=1}^{\infty}$ are iid variables from $N(0,1)$.

[28] See e.g. Lemma 2.3.14 of [11] for the case of Brownian motion; Brownian bridge involves the extra boundary condition $x(\ell) = 0$

Chapter 2

Advanced MCMC Methods

In this Chapter we present advanced versions of already established standard MCMC algorithms which were discussed in the previous chapter. The term 'advanced' is used in this thesis to characterize those algorithms that (in contrast with standard algorithms) are well-defined for target distributions Π on general (separable) Hilbert spaces $\mathcal{H}$ that are defined as a change of measure with respect to a centered Gaussian law $\Pi_0 = \tilde{\Pi} = N(0, \mathcal{C})$ i.e.

$$\frac{d\Pi}{d\tilde{\Pi}}(x) = \exp\{-\Phi(x)\}, \tag{2.0.1}$$

where $\Phi(x)$ is a real-valued function defined on $\mathcal{H}$ (in the previous Chapter we used the notation Π_0 to denote the reference Gaussian measure, but in this Chapter it will be convenient to switch the notation to $\tilde{\Pi}$). We sometimes refer to these algorithms which are well defined and stable as 'well-posed'. One consequence of the 'well-posedness' of these algorithms is that they have *mesh-free* properties that make their convergence properties stable upon increasing the dimension of the resolution in the target distribution. This is particularly important when we use MCMC to simulate SDE sample paths since, for practical purposes, the target path will be time-discretized, thus, having *mesh-free* MCMC algorithms means that as the discretization becomes finer the mixing time does not deteriorate.

2.1 Advanced MALA

Following the work from paper [8] we adapt the standard MALA algorithm described in Section 1.1.3 to obtain a new version of it that it is well-defined for infinite-dimensional target distributions Π as defined in (2.0.1).

Same as for the standard case, the development of advanced MALA builds upon the properties of the Langevin SDE discussed in Section 1.1.3 for finite-dimensional spaces. In the new setting where x refers to complete continuous paths, say, on

$\mathcal{H} = L^2([0,\ell],\mathbb{R}^d)$, $d \geq 1$ (we will use $d = 1$, unless otherwise stated,) we will be using the variational derivative[1] $\delta \log \Pi(x_t)$ instead of the gradient ∇ and, thus, we work with the following $\mathcal{H}$-valued SDE:

$$dx_t = \tfrac{1}{2}\,\mathcal{C}\,\{\delta \log \Pi(x_t)\}dt + \sqrt{\mathcal{C}}\,d\mathcal{W}_t, \tag{2.1.1}$$

where $\mathcal{W}_t$ refers to the standard cylindrical Wiener process used in the derivation of Stochastic Partial Differential Equations (SPDEs), (see e.g. [25]). The process $\mathcal{W}_t$ represents a complex object, we interpret it for practical purposes as follows, we have, for instance, that for any time-step $h > 0$:

$$\sqrt{\mathcal{C}}(\mathcal{W}_{t+h} - \mathcal{W}_h) \sim \sqrt{h}N(0,\mathcal{C}),$$

so increments of the noise term in (2.1.1) can be generated by using samples from the reference Gaussian measure $\Pi_0 \equiv N(0,\mathcal{C})$. It can be proven, that under regularity conditions on $\mathcal{C}$ and Ψ the continuous-time Markov process (2.1.1) has the target Π in (2.0.1) as its invariant distribution (see e.g. [47, 46]). We want to emphasize here, that each instantaneous position x_t is a complete path-element of $\mathcal{H}$, thus, we have that $x_t = \{x_t(u); u \in [0,\ell]\}$. Henceforth, we will use t and u to refer to the time and space directions respectively. Also, to provide an example for the computation of $\delta\Phi(x)$ needed in (2.1.1), we consider the fairly general case where $\Phi(x)$ has the following form:

$$\Phi(x) = \int_0^T \Psi(x(u))du. \tag{2.1.2}$$

for some sufficiently smooth mapping $\Psi : \mathbb{R} \mapsto \mathbb{R}$. Then, from standard calculus on $L^2([0,\ell],\mathbb{R})$ the variational derivative $\delta\Phi$ is a path on $[0,\ell]$ itself and is simply given as:

$$(\delta\Phi(x))(u) = \Psi'(x(u)), \quad u \in [0,\ell],$$

which is a generalisation of a gradient on pathspace (see e.g. [39]).

As for standard MALA, one cannot typically solve the SDE (2.1.1) analytically, so we will develop a time-discretization scheme that will deliver candidate paths within a Metropolis-Hasting framework. The choice of discretization scheme will be critically important for the developed algorithm. As we will soon explain, the

[1] The δ-notation refers to the Fréchet generalisation of differentiation on general Hilbert spaces; in particular, on the pathspace, under sufficient regularity, it corresponds to the notion of the variational derivative

choice for instance, of a simple Euler scheme would provide proposals that would be out of the domain of the target Π, so they would have zero acceptance probability. Notice that since we have $\tilde{\Pi} = N(0, \mathcal{C})$, the target Π can be formally defined via the density:

$$\Pi(x) \propto \exp\{-\Phi(x) - \tfrac{1}{2}\langle x, \mathcal{C}^{-1}x\rangle\},$$

with $\langle \cdot, \cdot \rangle$ being the inner product of the Hilbert space $\mathcal{H}$. Using this expression within the Langevin SDE (2.1.1), we can rewrite the latter as:

$$dx_t = \big(-\tfrac{1}{2}\,\mathcal{C}\delta\Phi(x) - x\big)\,dt + \sqrt{\mathcal{C}}\,d\mathcal{W}_t. \tag{2.1.3}$$

To time-discretize the Langevin SDE (2.1.3) we introduce a semi-implicit Euler scheme of the form:

$$x' - x = \big(\tfrac{1}{2}\,\mathcal{C}\delta\Phi(x) - \theta x - (1-\theta)x'\big)\,h + \sqrt{h}N(0, \mathcal{C}),$$

for parameter $\theta \in [0, 1]$. Notice that we use an implicit scheme only on the linear path of the drift; also, $\theta = 1$ corresponds to the standard explicit Euler scheme. In [8] it was shown empirically and theoretically that the only semi-implicit scheme that produces proposals with non-zero acceptance probability is the one that sets $\theta = \frac{1}{2}$, that is, the scheme that sets:

$$x' - x = \big(\tfrac{1}{2}\,\mathcal{C}\delta\Phi(x) - \tfrac{1}{2}x - \tfrac{1}{2}x'\big)\,h + \sqrt{h}N(0, \mathcal{C}).$$

After re-arranging, the proposed move x' is expressed as:

$$x' = \frac{1-\frac{h}{4}}{1+\frac{h}{4}}\,x + \frac{\sqrt{h}}{1+\frac{h}{4}}\,\Big(\xi - \sqrt{\tfrac{h}{4}}\,\mathcal{C}\delta\Phi(x)\Big), \tag{2.1.4}$$

for $\xi \sim N(0, \mathcal{C})$.

We can now provide some intuition about the reason why only when $\theta = 1/2$ we can achieve a scheme with ‘appropriate’ candidate proposals x'. Assume the setting that Π corresponds to the distribution of a target diffusion bridge of unit diffusion coefficient on some interval $[0, \ell]$, as for instance in the numerical example with the Ornstein-Uhlenbeck bridge in Section 1.2.3 (so, here Π_0 corresponds to the Gaussian law of a Brownian bridge). Recalling the discussion in Section 1.3, we can now think about the quadratic variation properties of x and x'.

Assuming $x \sim \tilde{\Pi}$ or $x \sim \Pi$, the quadratic variation of x at the terminal position ℓ is equal to ℓ. Now, if we were to ignore the non-linear term $\mathcal{C}\delta\Phi(x)$ for a moment, the proposal (2.1.4) would coincide with the ‘advanced’ version of the random walk proposal in (1.3.3) which was shown to preserve the quadratic variation properties of the paths. That is, the sum of the squares of the coefficients of x and ξ in (2.1.4) is equal to 1. This is a very critical remark for the well-posedness of the final algorithm. Considering $\mathcal{C}\delta\Phi(x)$ does not affect the almost-sure properties of x' (compared to not considering that term) since[2], $\mathcal{C}\delta\Phi(x)$ will typically belong in the image space of $\mathcal{C}^{1/2}$, thus the laws ξ and $\xi - \sqrt{h/4}\,\mathcal{C}\delta\Phi(x)$ are absolutely continuous with respect to each other.

$\mathcal{C}$ is essentially a 'smoothing' operator, so for any typical $x \sim \Pi$, we have that $\mathcal{C}\delta\Phi(x)$ will be smooth (thus, it's quadratic variation will be zero). This is obvious also from the exact specification of $\mathcal{C}$ for the case of the Brownian bridge in (1.4.2) in Section 1.4. Hence, the consideration of the quadratic variation already provides some intuition for the significance of selecting $\theta = 1/2$ and scheme (2.1.4). Next, we will be more formal, and define analytically the advanced MALA algorithm.

We define the advanced MALA algorithm by following the derivations in [8]. We introduce some notation to find the relevant acceptance probability on the Hilbert space $\mathcal{H}$. The development follows the theory in [81] which defines Metropolis-Hastings algorithms on general state spaces.

Let $Q(dx'|x)$ be the Markov transition kernel determined by the proposal (2.1.4). First, we define a bivariate law and its symmetrisation:

$$\begin{aligned}\mu(dx, dx') &:= \Pi(dx)Q(dx'|x)\\ \mu^T(dx, dx') &:= \Pi(dx')Q(dx|x'),\end{aligned}$$

so that if $(X, Y) \sim \mu$, then $(Y, X) \sim \mu^\top$. Following closely the generic specification of the accept/reject ratio from [81], if $\mu \simeq \mu^T$ (where '$\simeq$' means that the two relevant measures are absolutely continuous with each other) then, the acceptance probability is 'well-behaved' (i.e. it is not identically equal to zero, which is the case when μ and $\mu^\top$ are not absolutely continuous to each other) and equal to:

$$\alpha(x, x') = 1 \wedge \frac{d\mu^T}{d\mu}(x, x'). \tag{2.1.5}$$

Remark 2.1.1. *In the case that all probability measures involved (namely $\Pi(dx)$ and $Q(dx'|x)$) had a density with respect to a common reference measure (in the standard finite-dimensional settings, this would typically be the Lebesque measure) then the above expression (2.1.5) simplifies to the usual Metropolis-Hastings acceptance probability. The difference in our infinite-dimensional set-up is that, typ-*

[2] As we have seen in Section 1.4, and in particular Proposition 1.4.1

ically, there is not *a common reference measure for the probability laws $Q(dx'|x)$ over* all *current positions x. That is, we typically have that the laws $Q(dx'|x_1)$ and $Q(dx'|x_2)$ are singular for different x_1 and x_2. The non–existence of a common reference measure, thus, makes it necessary to instead consider simultaneously the joint bivariate laws of the current and proposed values x and x'.*

Finding the density involved in (2.1.5) is not a trivial task.We will need to work with a corresponding bivariate reference measure. For this reason, we define the following bivariate Gaussian law and its symmetrization:

$$\begin{aligned}\tilde{\mu}(dx, dx') &= \tilde{\Pi}(dx)\tilde{Q}(dx'|x)\\ \tilde{\mu}^T(dx, dx') &= \tilde{\Pi}(dx')\tilde{Q}(dx|x')\end{aligned}$$

where $\tilde{\Pi}$ is the reference Gaussian measure, and $\tilde{Q}(dx'|x)$ represents the distribution for the proposal that omits the non-linear term, that is:

$$x' = \frac{1-\frac{h}{4}}{1+\frac{h}{4}}\,x + \frac{\sqrt{h}}{1+\frac{h}{4}}\,\xi. \tag{2.1.6}$$

It is easy to check[3] that the bivariate Gaussian measure $\tilde{\mu}$ is symmetric, so that in fact:

$$\tilde{\mu}(dx, dx') \equiv \tilde{\mu}^T(dx, dx').$$

Now, using $\tilde{\mu}$ as the reference measure, we can re-write the density appearing in the acceptance probability in (2.1.5) as follows:

$$\frac{d\mu^T}{d\mu}(x, x') = \frac{d\mu^T/d\tilde{\mu}^T}{d\mu/d\tilde{\mu}}(x, x') = \frac{\frac{d\Pi}{d\tilde{\Pi}}(x')\frac{dQ}{d\tilde{Q}}(x|x')}{\frac{d\Pi}{d\tilde{\Pi}}(x)\frac{dQ}{d\tilde{Q}}(x'|x)}. \tag{2.1.7}$$

Notice now that $d\Pi/d\tilde{\Pi}$ is simply the original target density (i.e. we have the expression $(d\Pi/d\tilde{\Pi})(x) = \exp\{-\Phi(x)\}$). It remains to find the density $\frac{dQ}{d\tilde{Q}}(x'|x)$ for any $x \in \mathcal{H}$. Rewriting side-by-side the dynamics gives rise to these two transition probability measures:

$$\begin{aligned}Q(dx'|x):\quad x' &= \frac{1-\frac{h}{4}}{1+\frac{h}{4}}\,x + \frac{\sqrt{h}}{1+\frac{h}{4}}\,\big(\xi + \mathcal{C}^{1/2}g(x)\big),\\ \tilde{Q}(dx'|x):\quad x' &= \frac{1-\frac{h}{4}}{1+\frac{h}{4}}\,x + \frac{\sqrt{h}}{1+\frac{h}{4}}\,\xi,\end{aligned}$$

[3] See [8] for the analytical illustration

where we have defined:

$$g(x) := -\sqrt{\tfrac{h}{4}}\,\mathcal{C}^{1/2}\delta\Phi(x). \tag{2.1.8}$$

Now, for any fixed $x \in \mathcal{H}$, let $\rho_x(u)$ denote the density of the Gaussian law of $\xi + \mathcal{C}^{1/2}g(x)$ with respect to the law of ξ (recall that $\xi \sim \tilde{\Pi}$). Using Proposition 1.4.1 from Section 1.4 (the constant element x_0 used there corresponds now to $g(x)$), hence, we have that:

$$\begin{aligned}\rho_x(u) &= \exp\big\{\langle g(x), \mathcal{C}^{-1/2}u\rangle - \tfrac{1}{2}|g(x)|^2\big\}\\ &\equiv \exp\big\{-\langle\sqrt{\tfrac{h}{4}}\delta\Phi(x), u\rangle - \tfrac{h}{8}\langle\mathcal{C}\delta\Phi(x), \delta\Phi(x)\rangle\big\}\end{aligned} \tag{2.1.9}$$

to get the second expression we used the analytical expression for $g(x)$ in (2.1.8). Now, the actual density we are interested in involves probability measures that are simply an 1-1 transform of the laws with the density obtained in (2.1.9) above. And, thus, defining explicitly the transform mapping as:

$$r_x(\xi) := \frac{1-\frac{h}{4}}{1+\frac{h}{4}}\,x + \frac{\sqrt{h}}{1+\frac{h}{4}}\,\xi,$$

we immediately obtain that:

$$\frac{dQ}{d\tilde{Q}}(x'|x) = \rho_x(r_x^{-1}(x')). \tag{2.1.10}$$

We have now finished with the calculation of the required bivariate density in (2.1.7), thus, also with the acceptance probability of the advanced MALA in (2.1.5). We can now carry out some calculations using the analytical expressions above derived to obtain the following equality:

$$\begin{aligned}\log\big(\frac{d\mu}{d\tilde{\mu}}(x,x')\big) &= c - \Phi(x) - \tfrac{h}{4}\langle\mathcal{C}\delta\Phi(x), \delta\Phi(x)\rangle -\\ &- \tfrac{1}{2}\langle\delta\Phi(x), x'-x\rangle - \tfrac{h}{4}\langle\delta\Phi(x), x'+x\rangle\end{aligned} \tag{2.1.11}$$

for some constant $c \in \mathbb{R}$.

2.2 Advanced HMC

We now present here a derivation of advanced HMC. The advanced HMC algorithm was first introduced in [7]. A new contribution in this thesis is the derivation of a novel proof for the well-posedness of advanced HMC that avoids many of the technicalities of the proof in [7] and, more importantly, applies under weaker conditions on the target distribution. The findings in this Section have already being published in paper [5].

Recall that the target distribution is formally expressed as:

$$\Pi(x) \propto \{-\Phi(x) - \tfrac{1}{2}\langle x, Lx\rangle\}, \quad x \in \mathcal{H}, \tag{2.2.1}$$

where we have now explicitly defined the inverse covariance matrix of the reference Gaussian law:

$$L := \mathcal{C}^{-1}.$$

Recall the definition of standard HMC from Table 1.1, and the related quantities x^* (the proposal), h (the leapfrog step-size), and T (the time horizon). By going back to our initial arguments of Quadratic Variation in section 1.3 it is immediately obvious that the standard HMC scheme would not produce proposals suitable to the target distribution. Notice that applying the standard HMC algorithm in Table 1.1 on some N-dimensional projection of Π in (2.2.1), for $N \geq 1$, would give an algorithm where the proposal $x^\star$ would become an increasingly inappropriate candidate for a sample from the target with increasing N ([7]); thus, the acceptance probability would vanish with increasing dimension– N, assuming parameters h, T was kept fixed. Indeed, considering our standard scenario where Π corresponds to the law of a diffusion bridge with unit diffusion coefficient and the reference measure being the corresponding Brownian bridge, any single standard leapfrog step applied in this context would project Brownian bridge paths to paths of the wrong quadratic variation which would then necessarily have zero acceptance probability. In particular, the results in [6] suggest that one must decrease the step-size h to $\mathcal{O}(N^{-1/4})$ in order to control the acceptance probability for increasing N. The advanced HMC algorithm avoids this degeneration by exploiting the definition of the target as a change of measure from a Gaussian law and allows for fixed step-size $h = \mathcal{O}(1)$– even at the infinite-dimensional setting when $N = \infty$.

We can now recall the development of the Hamiltonian dynamics, as shown also for the standard HMC algorithm in Section 1.1.3. Notice that the target density involves here the extra quadratic term $\frac{1}{2}\langle x, Lx\rangle$ compared to the presentation for the standard HMC due to the presence of the reference Gaussian measure. Thus, in this context, the corresponding *total energy* function becomes as follows:

$$H(x, v; M) = \Phi(x) + \tfrac{1}{2}\langle x, Lx\rangle + \tfrac{1}{2}\langle v, Mv\rangle, \tag{2.2.2}$$

for some mass matrix M (recall that x should be interpreted as the ‘location’ variable and v as the 'velocity' variable). Recall that the Hamiltonian equations are determined as follows:

$$\frac{dx}{dt} = M^{-1}\frac{\partial H}{\partial v}, \quad M\frac{dv}{dt} = \frac{\partial H}{\partial x}, \tag{2.2.3}$$

or equivalently,

$$\frac{dx}{dt} = v, \quad M\frac{dv}{dt} = -Lx - \delta\Phi(x). \tag{2.2.4}$$

The choice of the mass matrix is important for the efficiency of standard HMC. In the infinite-dimensional context here, we have to be very careful with the choice of M to obtain a well-defined algorithm that has non-zero acceptance probability. Indeed, following [7] we select:

$$M = L.$$

Following the intuition mentioned in Section 1.1.3 for the mass matrix aiming at resembling the inverse covariance of the target distribution, it is clear that the choice of $M = L$ is ideal when the target is the reference Gaussian measure $N(0, \mathcal{C})$. Since the actual target is a change of measure from this reference law, $M = L$ seems like a sensible choice. We will see as we develop advanced MALA, that this is a choice that allows us to have a well-defined algorithm in the infinite-dimensional Hilbert space $\mathcal{H}$. Thus, we can now write the energy function as follows:

$$H(x, v) = \Phi(x) + \tfrac{1}{2}\langle x, Lx\rangle + \tfrac{1}{2}\langle v, Lv\rangle, \quad x \in \mathcal{H}, \tag{2.2.5}$$

and the Hamiltonian equations as:

$$\frac{dx}{dt} = v, \quad \frac{dv}{dt} = -x - \mathcal{C}\delta\Phi(x). \tag{2.2.6}$$

In order to derive the advanced algorithm, the Hamiltonian equations (2.2.6) are split into two equations[4]:

$$\frac{dx}{dt} = 0\ , \quad \frac{dv}{dt} = -\mathcal{C}\,\delta\Phi(x)\ ; \tag{2.2.7}$$

$$\frac{dx}{dt} = v\ , \quad \frac{dv}{dt} = -x\ . \tag{2.2.8}$$

Notice that both equations can be solved analytically. We construct a numerical integrator for (2.2.6) by synthesising steps on (2.2.7) and (2.2.8). Analytically, we

[4] This development of the method follows closely [7]; a similar splitting of the Hamiltonian equations used in [60], but in a different context

define the solution operators of (2.2.7) and (2.2.8) as follows:

$$\Xi_t(x,v) = (x,\, v - t\,\mathcal{C}\,\delta\Phi(x)) \; ; \tag{2.2.9}$$

$$\tilde{\Xi}_t(x,v) = \big(\cos(t)\,x + \sin(t)\,v,\, -\sin(t)\,x + \cos(t)\,v\big) \; . \tag{2.2.10}$$

The numerical integrator for (2.2.6) is defined as follows:

$$\Psi_h = \Xi_{h/2} \circ \tilde{\Xi}_{h^*} \circ \Xi_{h/2} \; , \tag{2.2.11}$$

for small $h, h^* > 0$. We can synthesize steps up to some time horizon T. We define

$$I = [\tfrac{T}{h}] \tag{2.2.12}$$

letting the operator Ψ_h^I correspond to the synthesis of I steps Ψ_h. Ψ_h^I provides the proposals for the MCMC steps. Now is a good time to state the assumption under which advanced HMC will be well-defined in infinite-dimensions:

Assumption 2.2.1. *$\mathcal{C}\,\delta\Phi(x)$ is an element of the Cameron-Martin space of the Gaussian measure Π_0 (so $\mathcal{C}\,\delta\Phi(x) \in \operatorname{Im}\mathcal{C}^{1/2}$) for all x in a set with probability 1 under Π_0.*

Based on Assumption 2.2.1, we make a remark that motivates the well-posedness of advanced HMC.

Remark 2.2.1. *Critically, operators $\Xi_t(x,v)$, $\tilde{\Xi}_t(x,v)$ have the property that they preserve the absolute continuity properties of an input random pair (x,v) distributed according to the Gaussian law:*

$$Q_0(x,v) \propto \exp\{-\tfrac{1}{2}\langle x, Lx\rangle - \tfrac{1}{2}\langle v, Lv\rangle\} \; , \tag{2.2.13}$$

(so, also for any other distribution absolutely continuous w.r.t. Q_0). This is obvious for $\tilde{\Xi}_t(x,v)$ as it defines a rotation, so this map is in fact invariant for Q_0. Then, as illustrated with Proposition 1.4.1, Assumption 2.2.1 guarantees precisely that also $\Xi_t(x,v)$ preserves absolute continuity of Q_0.

We will use h^* such that:

$$\cos(h^*) = \tfrac{1-h^2/4}{1+h^2/4} \; , \tag{2.2.14}$$

though any choice is, in principle, allowed. For this particular choice, it can be easily checked that the integrator $(x_0, v_0) \mapsto \Psi_h(x_0, v_0) =: (x_h, v_h)$ and it can be equivalently expressed as:

$$\begin{aligned} v_{h/2} &= v_0 - \tfrac{h}{2}\,\frac{x_0 + x_h}{2} - \tfrac{h}{2}\,\mathcal{C}\,\delta\Phi(x_0)\ , \\ x_h &= x_0 + h\, v_{h/2}\ , \\ v_h &= v_{h/2} - \tfrac{h}{2}\,\frac{x_0 + x_h}{2} - \tfrac{h}{2}\,\mathcal{C}\,\delta\Phi(x_h)\ , \end{aligned} \tag{2.2.15}$$

and that now can be interpreted as a semi-implicit-type integrator of (2.2.6). Under the interpretation (2.2.15), the justification for the choice of (2.2.14) is that it delivers an integrator Ψ_h that carries out steps of similar size h in the x and v directions, which is in accordance with standard HMC.

The complete algorithm is determined in Table 2.1. As with the standard HMC in Section 1.1.3, $\mathcal{P}_x$ denotes projection onto the x-argument.

Advanced HMC on Hilbert space $\mathcal{H}$:

(i) *Start with an initial value $x^{(0)} \sim N(0,\mathcal{C})$ and set $k = 0$.*

(ii) *Given $x^{(k)}$ sample $v^{(k)} \sim N(0,\mathcal{C})$ and propose*

$$x^{\star} = \mathcal{P}_x\, \Psi_h^I(x^{(k)}, v^{(k)})\ .$$

(iii) *Consider*

$$a = 1 \wedge \exp\{-\Delta H(x^{(k)}, v^{(k)})\} \tag{2.2.16}$$

for $\Delta H = H(\Psi_h^I(x,v)) - H(x,v)$.

(iv) *Set $x^{(k+1)} = x^{\star}$ with probability a; otherwise set $x^{(k+1)} = x^{(k)}$.*

(v) *Set $k \to k+1$ and go to (ii).*

Table 2.1: Advanced HMC on $\mathcal{H}$, with target $\Pi(x)$ in (2.2.1).

Remark 2.2.2. *The acceptance probability in Table 2.1 is here defined only formally, as $H(x,v) = \infty$ with probability 1. To see that, notice that using the Karhunen-Loève expansion introduced in (1.4.3) for $x \sim N(0,\mathcal{C})$ we have that $\langle x, Lx\rangle \equiv \sum_{p=1}^{\infty} \xi_p^2$, for ξ_p iid $N(0,1)$. We re-express the acceptance probability in the following section in a way that illustrates that the difference $\Delta H = H(\Psi_h^I(x,v)) - H(x,v)$ is a.s. well-defined; from a practical point of view, for the N-dimensional projection used in practice one could still use directly the expression*

$\Delta H = H(\Psi_h^I(x,v)) - H(x,v)$ *as each of the two* H*-terms will grow as* $\mathcal{O}(N)$*.*

Remark 2.2.3. *We will not prove the existence of a solution for the continuous-time Hamiltonian equations on Hilbert space (2.2.7)-(2.2.8) or that the solution would preserve* $\Pi(x,v)$ *as such proofs are beyond the scope of this thesis. In Section 2.2.1 below we will prove the validity of the algorithm in Table 2.1 which uses directly the numerical integrators of these equations in (2.2.9)-(2.2.10). This seems to suffice from a practical point of view: our proof below indicates that the algorithm will not collapse as* $N \to \infty$ *but will converge to a limit, with* N *being the dimension of the vector used when we discretize the complete infinite-dimensional diffusion paths when running the algorithms on a personal computer. Later, for a fixed finite dimension* N*, we can resort to the properties of finite-dimensional Hamiltonian equations to justify that, under standard regulatory conditions, they will indeed preserve the* N*-dimensional target distribution and, thus, we can attain average acceptance probabilities arbitrarily close to* 1 *by decreasing the step-size* h*.*

2.2.1 Validity of Advanced HMC

We will now prove analytically that the algorithm, as described in Table 2.1, is well defined on the Hilbert space $\mathcal{H} = L^2([0,\ell],\mathbb{R})$ and gives rise to Markov dynamics on the x-argument that preserve the target distribution Π in (2.2.1). The proof for the well-posedness of the algorithm in infinite dimensions will build upon the intuitive understanding described in Remark 2.2.1.

For the next proof, we define the operator $\tilde{\Psi}_h^I$ which is as Ψ_h^I but with the non-linear parts set to zero, that is $\Phi \equiv 0$. We consider the Gaussian product measure $Q_0 = N(0,\mathcal{C}) \otimes N(0,\mathcal{C})$ on $\mathcal{H} \times \mathcal{H}$ as in (2.2.13) and the bivariate distribution Q defined via the change of measure:

$$Q(dx,dv) = \exp\{-\Phi(x)\}\, Q_0(dx,dv).$$

We also consider the sequence of probability measures on $\mathcal{H} \times \mathcal{H}$:

$$Q^{(i)} = Q \circ \Psi_h^{-i}, \quad 1 \le i \le I,$$

the sequence

$$(x_i, v_i) = \Psi_h^i(x_0, v_0),$$

and set:

$$g(x) := -\tfrac{h}{2}\,\mathcal{C}^{1/2}\delta\Phi(x), \quad x \in \mathcal{H}.$$

Notice that under Assumption 2.2.1, $g(x)$ is a well-defined element of the Hilbert space $\mathcal{H}$ a.s. under Π_0. One can think of $Q^{(i)}(dx, dv)$ as the 'flow' of the joint distribution of the location and velocity components, started from stationarity, and evolving due to the application of the leapfrog deterministic maps. For instance, in a simple finite-dimensional setup we could simply use standard change-of-variables formulae to determine the sequence of probability distributions $Q^{(i)}(dx, dv)$. In our infinite-dimensional context, extra caution is needed as we cannot apply the change-of-variables formula anymore (for instance, it is not obvious how to extend the Jacobian to the infinite-dimensional setting). Also, some care is needed for the choice of reference measure with respect to which we will calculate relevant densities.

As already mentioned, Proposition 1.4.1 specifies the density of a translation of a centered Gaussian measure with respect to 'smooth' constant elements of the Cameron-Martin of the Gaussian measure (which coincides with the image space of $\mathcal{C}^{1/2}$). We can now prove the following result:

Proposition 2.2.1. *We have that:*

$$\begin{aligned}\frac{dQ^{(i)}}{dQ_0}(x_i, v_i) = \frac{dQ^{(i-1)}}{dQ_0}&(x_{i-1}, v_{i-1}) \times G(x_i, v_i) \\ &\times G(x_{i-1}, v_{i-1} + \mathcal{C}^{1/2} g(x_{i-1})),\end{aligned}$$

where we have defined:

$$\frac{d\{Q_0 \circ \Xi_{h/2}^{-1}\}}{dQ_0} = \exp\{\langle g(x), \mathcal{C}^{-1/2} v\rangle - \tfrac{1}{2}|g(x)|^2\} =: G(x, v).$$

Proof. We will use the chain rule and Proposition 1.4.1. Recall that for any two measurable spaces $(E, \mathcal{E})$, $(E', \mathcal{E}')$, probability measures M, M_0 on $(E, \mathcal{E})$ and 1-1 mapping $F : (E, \mathcal{E}) \mapsto (E', \mathcal{E}')$, we have the following identity rule for the Radon-Nikodym derivative:

$$\frac{d\{M_1 \circ F^{-1}\}}{d\{M_0 \circ F^{-1}\}}(x) = \frac{dM_1}{dM_0}(F^{-1}(x)) \ . \tag{2.2.17}$$

Following the definition of Ψ_h from (2.2.11), we have the equality of probability measures:

$$Q^{(i)} = Q^{(i-1)} \circ \Xi_{h/2}^{-1} \circ \tilde{\Xi}_{h^*}^{-1} \circ \Xi_{h/2}^{-1} \ . \tag{2.2.18}$$

Thus, we have that:

$$\begin{aligned}
\frac{dQ^{(i)}}{dQ_0}(x_i, v_i) &= \frac{d\left\{Q^{(i-1)} \circ \Xi_{h/2}^{-1} \circ \tilde{\Xi}_{h^*}^{-1} \circ \Xi_{h/2}^{-1}\right\}}{dQ_0}(x_i, v_i) \\
&= \frac{d\left\{Q^{(i-1)} \circ \Xi_{h/2}^{-1} \circ \tilde{\Xi}_{h^*}^{-1} \circ \Xi_{h/2}^{-1}\right\}}{d\left\{Q_0 \circ \Xi_{h/2}^{-1}\right\}}(x_i, v_i) \times \frac{d\left\{Q_0 \circ \Xi_{h/2}^{-1}\right\}}{dQ_0}(x_i, v_i) \\
&= \frac{d\left\{Q^{(i-1)} \circ \Xi_{h/2}^{-1} \circ \tilde{\Xi}_{h^*}^{-1}\right\}}{dQ_0}(\Xi_{h/2}^{-1}(x_i, v_i)) \times G(x_i, v_i), \qquad (2.2.19)
\end{aligned}$$

where we have used the chain rule in the second line, then the identity (2.2.17) and finally Proposition 1.4.1 (in this case with $x_0 \equiv g(x)$) in the third line. Using the fact that $Q_0 \circ \tilde{\Xi}_{h^*}^{-1} \equiv Q_0$ (as $\tilde{\Xi}_{h^*}$ is a rotation that clearly preserves the bivariate Gaussian law Q_0) and upon observing that we have the following identity:

$$(\tilde{\Xi}_{h^*}^{-1} \circ \Xi_{h/2}^{-1})(x_i, v_i) \equiv \Xi_{h/2}(x_{i-1}, v_{i-1}),$$

then, we obtain that:

$$\frac{d\left\{Q^{(i-1)} \circ \Xi_{h/2}^{-1} \circ \tilde{\Xi}_{h^*}^{-1}\right\}}{dQ_0}(\Xi_{h/2}^{-1}(x_i, v_i)) \equiv \frac{d\left\{Q^{(i-1)} \circ \Xi_{h/2}^{-1}\right\}}{dQ_0}(\Xi_{h/2}(x_{i-1}, v_{i-1})).$$

Finally, working as in (2.2.19) we have that:

$$\begin{aligned}
&\frac{d\left\{Q^{(i-1)} \circ \Xi_{h/2}^{-1}\right\}}{dQ_0}(\Xi_{h/2}(x_{i-1}, v_{i-1})) = \\
&\qquad = \frac{dQ^{(i-1)}}{dQ_0}(x_{i-1}, v_{i-1}) \times \frac{d\left\{Q_0 \circ \Xi_{h/2}^{-1}\right\}}{dQ_0}(\Xi_{h/2}(x_{i-1}, v_{i-1})) \\
&\qquad = \frac{dQ^{(i-1)}}{dQ_0}(x_{i-1}, v_{i-1}) \times G(\Xi_{h/2}(x_{i-1}, v_{i-1})).
\end{aligned}$$

The definition of $\Xi_{h/2}$ gives that:

$$G(\Xi_{h/2}(x_{i-1}, v_{i-1})) \equiv G(x_{i-1}, v_{i-1} + \tfrac{h}{2}\mathcal{C}^{1/2} g(x_{i-1})).$$

By following through the calculation from (2.2.19) we have now proven the requested result.

□

Thus, using Proposition 2.2.1 iteratively we have now obtained that:

$$\frac{dQ^{(I)}}{dQ_0}(x_I, v_I) = \frac{dQ}{dQ_0}(x_0, v_0) \times \prod_{i=1}^{I} G(x_i, v_i)\, G(x_{i-1}, v_{i-1} + \mathcal{C}^{1/2} g(x_{i-1})). \quad (2.2.20)$$

Now, following the definition of Ψ_h in (2.2.11), we set:

$$\begin{aligned} v_{i-1}^{-} &= \mathcal{P}_v\, \Xi_{h/2}(x_{i-1}, v_{i-1}) \equiv v_{i-1} + \mathcal{C}^{1/2} g(x_{i-1}), \\ v_i^{+} &= \mathcal{P}_v\, (\, \tilde{\Xi}_{h^*} \circ \Xi_{h/2}(x_{i-1}, v_{i-1})\,) \equiv v_i - \mathcal{C}^{1/2} g(x_i). \end{aligned}$$

($\mathcal{P}_v$ denotes projection onto the v-argument.) Using these definitions, for any $h, h^* > 0$ we have that:

$$\begin{aligned} &\log\{\, G(x_i, v_i)\, G(x_{i-1}, v_{i-1} + \tfrac{h}{2}\, \mathcal{C}^{1/2} g(x_{i-1}))\, \} = \\ &\quad = \langle \tfrac{h}{2}\, g(x_i), \mathcal{C}^{-1/2} v_i \rangle - \tfrac{1}{2} |\tfrac{h}{2} g(x_i)|^2 + \langle \tfrac{h}{2}\, g(x_{i-1}), \mathcal{C}^{-1/2} v_{i-1} \rangle + \tfrac{1}{2} |\tfrac{h}{2} g(x_{i-1})|^2 \\ &\quad = \tfrac{1}{2}\, \langle v_i, L v_i \rangle - \tfrac{1}{2}\, \langle v_i^{+}, L v_i^{+} \rangle - \tfrac{1}{2}\, \langle v_{i-1}, L v_{i-1} \rangle + \tfrac{1}{2}\, \langle v_{i-1}^{-}, L v_{i-1}^{-} \rangle \\ &\quad = \tfrac{1}{2}\, \langle x_i, L x_i \rangle + \tfrac{1}{2}\, \langle v_i, L v_i \rangle - \tfrac{1}{2}\, \langle x_{i-1}, L x_{i-1} \rangle - \tfrac{1}{2}\, \langle v_{i-1}, L v_{i-1} \rangle\ . \end{aligned}$$

The last equation is due to the mapping $(x_{i-1}, v_{i-1}^{-}) \mapsto (x_i, v_i^{+})$ corresponding to the modulus-preserving rotation $\tilde{\Xi}_{h^*}$. Thus, we can rewrite (2.2.20) as follows:

$$\frac{dQ^{(I)}}{dQ_0}(x_I, v_I) = \exp\{\Delta H(x_0, v_0) - \Phi(x_I)\}. \quad (2.2.21)$$

The above expression will now be used for proving the main result below.

Remark 2.2.4. *The operator Ψ_h (thus, also Ψ_h^I) has the following properties:*

i) Ψ_h is symmetric, that is $\Psi_h \circ S \circ \Psi_h = S$ where $S(x, v) = (x, -v)$.

ii) Ψ_h is (formally) volume-preserving, as it preserves volume when $\mathcal{H} \equiv \mathbb{R}^d$.

Theorem 2.2.1. *The Markov chain with transition dynamics specified in Table 2.1 has invariant distribution $\Pi(x)$ in (2.2.1).*

Proof. Assuming stationarity, so that $(x_0, v_0) \sim Q$, we can write for the next position, x', of the Markov chain (recall that $(x_I, v_I) = \Psi_h^I(x_0, v_0)$):

$$x' = \mathrm{I}\,[\, U \le a(\Psi_h^{-I}(x_I, v_I))\,]\, x_I + \mathrm{I}\,[\, U > a(x_0, v_0)\,]\, x_0,$$

for a uniform random variable $U \sim \mathrm{Un}\,[0, 1]$. Let $f : \mathcal{H} \mapsto \mathbb{R}$ be bounded and continuous. We need to prove that:

$$\mathrm{E}\,[\,f(x')\,] = \mathrm{E}\,[\,f(x_0)\,].$$

Integrating out U from above we get:

$$\mathrm{E}\,[\,f(x')\,] = \mathrm{E}[\,f(x_I)\,a(x_0,v_0)\,] - \mathrm{E}[\,f(x_0)\,a(x_0,v_0)\,] + \mathrm{E}\,[\,f(x_0)\,]. \tag{2.2.22}$$

The integrators in expectations/integrals are explicitly shown as a subscript of E, hence it is important to notice that:

$$\begin{aligned}\mathrm{E}[\,f(x_I)\,a(x_0,v_0)\,] &= \mathrm{E}_{Q^{(I)}}[\,f(x_I)\,a(\Psi_h^{-I}(x_I,v_I))\,]\\ &\overset{(2.2.21)}{=} \mathrm{E}_{Q_0}[\,f(x_I)\,a(\Psi_h^{-I}(x_I,v_I))\,e^{\Delta H(\Psi_h^{-I}(x_I,v_I))-\Phi(x_I)}\,]\\ &= \mathrm{E}_{Q_0}[\,f(x_I)\,(\,1\wedge e^{\Delta H(\Psi_h^{-I}(x_I,v_I))}\,)\,e^{-\Phi(x_I)}\,]\\ &= \mathrm{E}_{Q}[\,f(x_I)\cdot 1\wedge e^{\Delta H(\Psi_h^{-I}(x_I,v_I))}\,]\\ &= \mathrm{E}_{Q}[\,f(x_I)\cdot 1\wedge e^{\Delta H(\Psi_h^{-I}(x_I,-v_I))}\,].\end{aligned} \tag{2.2.23}$$

(For the last equation, notice that (x_I, v_I) and $(x_I, -v_I)$ have the same law Q.) Next, due to the symmetricity property $\Psi_h^I \circ S \circ \Psi_h^I = S$ of the leapfrog operator in Remark 2.2.4 we have that $\Psi_h^{-I} \circ S = S \circ \Psi_h^I$. Thus, we have:

$$\begin{aligned}\Delta H(\Psi_h^{-I}(x_I,-v_I))) &= \Delta H(S\circ\Psi_h^I(x_I,v_I)))\\ &= H(S(x_I,v_I)) - H(S\circ\Psi_h^I(x_I,v_I)) \equiv -\Delta H(x_I,v_I),\end{aligned}$$

which is the last equation where we used the fact that $H\circ S = H$ due to the energy H being quadratic in the velocity v. Thus, using this in (2.2.23), we have that:

$$\mathrm{E}[\,f(x_I)\,a(x_0,v_0)\,] = \mathrm{E}_Q[\,f(x_I)a(x_I,v_I)\,] \equiv \mathrm{E}[\,f(x_0)\,a(x_0,v_0)\,]. \tag{2.2.24}$$

So, from (2.2.22), the proof is now complete. □

Remark 2.2.5. *The demonstration of validity of standard HMC [34] does not require the recursive calculation of the forward density (2.2.21) as it exploits the preservation of volume (unit Jacobian) for the mapping $(x_0, v_0) \mapsto \psi_h^I(x_0, v_0)$ to directly prove the analogue to (2.2.24). So, finding (2.2.21) overcomes the difficulty of making sense of a Jacobian for the transform Ψ_h^I on the infinite-dimensional Hilbert space.*

2.3 Summary of Advanced Methods

A number of advanced MCMC algorithms corresponding to an upgrade of standard RWM, MALA and HMC and adapted to the infinite-dimensional pathspace are now available. So far in Section 2.1 we have defined advanced MALA. Then, in Section 2.2 we have defined advanced HMC. Following [8], a small modification of advanced MALA– whereby one uses only the 'blind' part of the proposal in 2.1.6– provides the advanced RWM. Here we briefly summarise all the advanced algorithms and the dynamics employed for their derivation.

The starting point for MALA is a Langevin SDE with drift $\frac{1}{2}\mathcal{C}\,\delta\log\Pi(x)$ and diffusion coefficient $\mathcal{C}^{1/2}$, that is, after a calculation on the drift:

$$\frac{dx}{dt} = -\tfrac{1}{2}\,x - \tfrac{1}{2}\,\mathcal{C}\,\delta\Phi(x) + \mathcal{C}^{1/2}\,\frac{d\mathcal{W}}{dt}. \tag{2.3.1}$$

In an Euclidean setting $\{\mathcal{W}_t\}$ denotes a standard Brownian motion, whereas in the pathspace it denotes a cylindrical Brownian motion. In both cases, the process can be easily understood via the distribution of it's increments, as $\mathcal{C}^{1/2}\frac{(\mathcal{W}_{t+s}-\mathcal{W}_t)}{\sqrt{s}} \sim N(0,\mathcal{C})$. On pathspace, the SDE (2.3.1) is shown in [8] to have invariant distribution Π under Lipschitz continuity and absolute boundedness assumptions on $\delta\Phi$. In the interesting, from a practical point of view, case of nonlinearity, this SDE cannot be solved analytically. So, a proposal can be derived via the following Euler-type scheme on (2.3.1) for an finite increment $\Delta t > 0$:

$$x' - x = -\Delta t\,(\theta\,\tfrac{x'}{2} + (1-\theta)\tfrac{x}{2}) - \tfrac{\Delta t}{2}\,\mathcal{C}\,\delta\,\Phi(x) + \sqrt{\Delta t}\,N(0,\mathcal{C}). \tag{2.3.2}$$

Standard MALA is derived from an explicit Euler scheme with $\theta = 0$ and advanced pathspace MALA from a semi-implicit scheme with $\theta = 1/2$. Contrasting (2.3.2) with the leapfrog steps (2.2.15), one can observe that standard (resp. advanced) MALA is a particular case of standard (resp. advanced) HMC when choosing $h = \sqrt{\Delta t}$ and a single leapfrog step $I = 1$. Finally, the advanced RWM algorithm on pathspace is derived in [8] via proposal (2.3.2) for $\theta = 1/2$ and also by omitting the nonlinear term $\mathcal{C}\,\delta\Phi(x)$. That is, the proposal for advanced RWM is:

$$x' = \rho\,x + \sqrt{1-\rho^2}\,N(0,\mathcal{C}),$$

with parameter

$$\rho = \frac{1-\frac{\Delta t}{4}}{1+\frac{\Delta t}{4}}.$$

The Metropolis-Hastings acceptance probability for this proposal (see [8]) is reminiscent of the one for standard RWM, namely $1 \wedge (\Pi(x')/\Pi(x))$, which also explains the interpretation of this algorithm as 'advanced RWM'. Table 2.2 summarises the three pathspace samplers looked at in this Chapter together with their standard versions for finite-dimensional spaces.

Algorithm	Pathspace Proposal	Standard Proposal
HMC	$x' = \mathcal{P}_x \Psi_h^I(x,v)$	$x' = \mathcal{P}_x \psi_h^I(x,v)$
MALA	$x' = \rho\, x + \sqrt{1-\rho^2}\, v - \frac{\Delta t}{2}\, \mathcal{C}\, \delta\Phi(x)$	$x' = (1 - \frac{\Delta t}{2})x - \frac{\Delta t}{2}\, \mathcal{C}\, \delta\Phi(x) + \sqrt{\Delta t}\, v$
RWM	$x' = \rho\, x + \sqrt{1-\rho^2}\, v$	$x' = x + \sqrt{\Delta t}\, v$

Table 2.2: Advanced MCMC algorithms on pathspace together with their standard versions. In all cases $v \sim N(0,\mathcal{C})$. HMC for $I = 1$ and $h = \sqrt{\Delta t}$ coincides with MALA.

Chapter 3

Investigation of Algorithmic Efficiency of Advanced MCMC Algorithms

In this chapter we investigate the performance of the hitherto mentioned advanced MCMC algorithms through the application of a variety of statistical models driven by diffusion processes. Also, we obtain analytical mathematical results for a particular target diffusion bridge by examining the relationship between the length of the bridge and the complexity of the algorithm. More specifically, we increase the length ℓ of the target bridge and investigate the choice of step-size as a function of ℓ so that the acceptance probability is controlled. This also provides evidence for the mixing time of the algorithms. An important conclusion here is that, for the case of long bridges, advanced MALA and RWM have similar behaviour, whereas advanced HMC performs much better. Thus, the take-home message is that well thought out use of information about the derivative of the target density can have an important impact on the performance of MCMC methodology. As far as analytical results is concerned, we first start with RWM and MALA and exploit many of the calculations carried out in [83] where the quantity of interest was not the length of the bridge, but rather the 'strength' of non-linearity for the target bridge (determined by a parameter of the drift function). We then produce some analytical results for the case of advanced HMC. These findings have been published in our paper [5].

3.0.1 Analytical study of RWM and MALA

The derivations that follow in this Section will exploit a number of analytical results obtained in [83]. The model of interest here is an Ornstein-Uhlenbeck diffusion bridge with reversion parameter κ and mean $\mu = 0$, namely:

$$\begin{aligned} dX_t &= -\kappa X_t dt + dB_t, \\ X_0 &= X_\ell = 0. \end{aligned} \tag{3.0.1}$$

From Girsanov's theorem we get that the target distribution is defined on the Hilbert space $\mathcal{H} = L^2([0,\ell],\mathbb{R})$ and is expressed in the general form of (2.0.1), so that:

$$\begin{aligned} &\frac{d\Pi}{d\tilde{\Pi}}(x) = \exp\{-\Phi(x)\} \ ; \\ \tilde{\Pi} = N(0, \mathcal{C}^{bb})\ ,\ \ &\Phi(x) = \tfrac{\kappa^2}{2}\int_0^\ell x^2(u)du + c \ , \end{aligned} \tag{3.0.2}$$

for some constant $c \in \mathbb{R}$, with $N(0, \mathcal{C}^{bb})$ the distribution of a standard Brownian bridge with $x(0) = x(\ell) = 0$ (recall here that $\mathcal{C}^{bb}$ denotes the covariance operator of the Brownian bridge, analytically specified in (1.4.2)). We look at the complexity of pathspace samplers as a function of the length ℓ of the bridge. The work in [83] has looked at identifying the complexity of advanced RWM and MALA for increasing $\kappa > 0$. We will perform a similar complexity analysis for the case of the length of the bridge $\ell > 0$, and include, also for the first time, an analysis for advanced HMC. All the advanced algorithms are defined as in Table 2.2.

Note 3.0.1. *For practical purposes, we use a specific case of an Ornstein-Uhlenbeck process for our analytical results. It still remains to show these scaling results for a general case. Regardless, our results for a OU bridge still provide some insight on how other models might scale. In section 3.1 we use numerical methods to compare various different models.*

Our main result summarises the mixing times as follows:

$$\begin{aligned} \text{RWM}: &\ \ \mathcal{O}(\ell^2) \ ; \\ \text{MALA}: &\ \ \mathcal{O}(\ell^2) \ ; \\ \text{HMC}: &\ \ \mathcal{O}(\ell) \ . \end{aligned} \tag{3.0.3}$$

The notion of mixing time is used here in an informal, practical manner and should not be confused with analytical definitions of various different versions of mixing times appearing in the Markov chain literature. In particular, the results below provide appropriate scalings of the step-sizes for the relevant MCMC samplers as a function of ℓ that deliver non-vanishing acceptance probabilities as ℓ grows.

Then, informal arguments are used to connect mixing times with the inverse of such step-sizes.

Note 3.0.2. *We only consider the case where the Markov Chain is in equilibrium, so we do not directly examine burn-in times. It is not immediately obvious that these results can be extrapolated to the case where the chain is not in stationarity.*

Notice that the acceptance probability for both advanced RWM and MALA can be written as:

$$a(x, v) = 1 \wedge e^{R(x,v)}$$

for some appropriate choice of exponent $R = R(x, v)$.

Theorem 3.0.1. *Consider the advanced RWM and MALA algorithms as specified in Table 2.2 targeting the OU-bridge in (3.0.2). Let $a = a(x, v)$ denote the acceptance probability for both algorithms. For any constant $c > 0$ we then have:*

i) If $\Delta t = c/\ell^2$ then $\limsup_\ell E[\alpha] > 0$.

ii) If $\Delta t = c/\ell^\epsilon$ for $\epsilon \in (0, 2)$ then $E[\alpha] \to 0$ as $\ell \to \infty$.

Proof. To prove the first result we will use several of the analytical results in [83]. First, for proving (i): notice that it is sufficient to show that $\sup_\ell |R(x, v)|_{L_1} < \infty$ since for any $\lambda > |R|_{L_1}$ we have the inequality:

$$E[1 \wedge e^R] \geq \exp(-\lambda) \left(1 - \frac{|R|_{L_1}}{\lambda}\right). \tag{3.0.4}$$

Result (i) illustrates that using the scaling $\Delta t = c/\ell^2$ provides an acceptance probability that does not deteriorate to zero when increasing ℓ. The second result in (ii) illustrates that any step-size larger than the one in (i) will provide an unstable algorithm with diminishing acceptance probability for an increasing ℓ. To prove (ii), we need to identify the term in R that has the largest L_1-norm. We denote this term as J. Intuitively, this term will lie in the interval $(-\infty, 0)$ and will be approaching $-\infty$ faster than the term $|R - J|$. Following [83], we carry out our proof by using the following inequality, for any $\gamma > 0$:

$$\begin{aligned} E[1 \wedge e^R] &\leq P[R \geq -\gamma] + e^{-\gamma} \\ &= P[\{R \geq -\gamma\} \bigcap |R - J| \leq \gamma] + P[\{R \geq -\gamma\} \bigcap |R - J| > \gamma] \\ &\leq P[J \geq -2\gamma] + P[|R - J| > \gamma] + e^{-\gamma} \\ &\leq P[J \geq -2\gamma] + \frac{|R - J|_{L_1}}{\gamma} + e^{-\gamma}, \end{aligned} \tag{3.0.5}$$

where we used Chebyshev's Inequality. To prove that $E[\alpha] \to 0$ we will use the above inequality and choose an appropriate γ that grows to infinity faster than $|R - J|_{L_1}$ but slower than the growth of $|J|_{L_1}$. To do this we must examine all the terms in R using the Karhunen-Loève expansion (see 1.4) for the Gaussian measure corresponding to the OU-bridge.

We can borrow directly the following results from [83], where we have assumed that we are in stationarity so that x is distributed according to the target OU-bridge and u is distributed according to the relevant reference measure corresponding to a Brownian bridge:

$$\begin{aligned}
&E|x|^2 = \frac{\ell}{2\kappa\ell} - \frac{1}{2\kappa^2}; \quad E|\mathcal{C}x|^2 = \frac{\ell^4}{90\kappa^2} - \frac{1}{6\kappa^4} + \frac{l}{2\kappa^5 \tanh(\kappa\ell)};\\
&E[\langle x, \mathcal{C}x\rangle] = \frac{3+\kappa^2\ell^2}{6\kappa^4} - \frac{\ell}{2\kappa^3\tanh(\kappa\ell)};\\
&E[\langle x, \mathcal{C}^3x\rangle] = \frac{945+315\kappa^2\ell^2-21\kappa^4\ell^4+2\kappa^6\ell^6}{1890\kappa^8} - \frac{\ell}{2\kappa^7\tanh(\kappa\ell)};\\
&E|v|^2 = \frac{\ell^2}{6}; \quad E[\langle v, \mathcal{C}v\rangle] = \frac{\ell^4}{90};\\
&E[\langle x, v\rangle^2] = \frac{3+\kappa^2}{6\kappa^4} - \frac{\ell}{2\kappa^3\tanh(\kappa\ell)};\\
&E[\langle \mathcal{C}x, v\rangle^2] = \frac{945+315\kappa^2\ell^2-21\kappa^4\ell^4+2\kappa^6\ell^6}{1890\kappa^8};\\
&E[\langle \mathcal{C}^2x, v\rangle^2] =\\
&\frac{467775+155925\kappa^2\ell^2-10395\kappa^4\ell^4+990\kappa^6\ell^6-99\kappa^8\ell^8+10\kappa^{10}\ell^{10}}{467775\kappa^{12}}. \qquad (3.0.6)
\end{aligned}$$

These results will be used for both advanced RWM and MALA algorithms.

Proof for Advanced RWM:
Notice now that for the case of advanced RWM we have that:

$$\begin{aligned}
R(x,v) = \frac{\kappa^2}{2(1+\frac{\Delta t}{4})^2}\Delta t\langle x, x\rangle - \frac{\kappa^2}{2(1+\frac{\Delta}{4})^2}\Delta t\langle v, v\rangle\\
- \frac{\kappa^2(1-\frac{\Delta t}{4})}{(1+\frac{\Delta t}{4})^2}\sqrt{\Delta t}\langle x, v\rangle.
\end{aligned}$$

Using the results from (3.0.6), we can see that setting $\Delta t = c/\ell^2$ would make the L_1-norm of each of these summands $\mathcal{O}(1)$. Thus, we obtain that $\sup_\ell |R(x,v)|_{L_1} < \infty$. For the negative result we identify the term, termed J, in the analytical expression for $R(x, v)$ with the largest L_1-norm. Specifically:

$$J = -\frac{\kappa^2}{2(1+\frac{\Delta t}{4})^2}\Delta t\langle v, v\rangle. \tag{3.0.7}$$

Thus, we have that:

$$|J|_{L_1} = \mathcal{O}(\Delta t\,\ell^2); \quad |R-J|_{L_1} = \mathcal{O}(\sqrt{\Delta t}\,\ell).$$

We now use (3.0.5) under the following choice of $\gamma = (\Delta t\,\ell^2)^{2/3} \to \infty$. This selection implies that $|R-J|_{L_1}/\gamma \to 0$ as $\ell \to \infty$. We now turn our attention to the term $P[J \geq -2\gamma]$, and show that it converges to zero (the intuition being that term J deviates to $-\infty$ faster than -2γ). To prove this, we need to look at the analytical definition of J. Using the rescaling properties of a Brownian bridge, we can re-write $v_{t\ell} = \sqrt{\ell}\,\tilde{v}_t$ where we have that $\tilde{v}$ is a standard Brownian bridge on $[0, 1]$. Thus, we can re-write:

$$\langle v, v\rangle = \int_0^\ell v_t^2 dt = \ell\int_0^1 v_{t\ell}^2 dt \equiv \ell^2\int_0^1 \tilde{v}_t^2 dt = \ell^2|\tilde{v}|^2.$$

Hence, we have that:

$$P[J \geq -2\gamma] = P\Big[|\tilde{v}|^2 \leq \frac{4(1+\Delta t/4)^2}{\kappa^2}\frac{\gamma}{\Delta t\,\ell^2}\Big] \tag{3.0.8}$$

which goes to 0 when $\ell \to \infty$ as $\frac{\gamma}{\Delta t\,\ell^2} \to 0$.

Proof for Advanced MALA:

The proof for the case of advanced MALA follows the same pattern. The first step is to obtain the analytical expression for $R(x,v)$. Following the calculations in [83] we have that:

$$\begin{aligned} R(x,v) = &\frac{\kappa^2}{8(1+\frac{\Delta t}{4})^2}\Delta t^2\langle x,x\rangle - \frac{\kappa^6}{32(1+\frac{\Delta t}{4})^2}\Delta t^3\langle \mathcal{C}x,\mathcal{C}x\rangle \\ &- \frac{\kappa^4(1-\frac{\Delta t}{4})}{8(1+\frac{\Delta t}{4})^2}\Delta t^2\langle x,\mathcal{C}x\rangle + \frac{\kappa^2}{8(1+\frac{\Delta t}{4})^2}\Delta t^2\langle x,\mathcal{C}x\rangle \\ &- \frac{\kappa^6}{32(1+\frac{\Delta t}{4})^2}\Delta t^3\langle \mathcal{C}^2x,\mathcal{C}x\rangle + \frac{\kappa^4(1-\frac{\Delta t}{4})}{8(1+\frac{\Delta t}{4})^2}\Delta t^2\langle \mathcal{C}x,\mathcal{C}x\rangle \\ &- \frac{\kappa^2}{8(1+\frac{\Delta t}{4})^2}\Delta t^2\langle v,v\rangle - \frac{\kappa^2}{8(1+\frac{\Delta t}{4})^2}\Delta t^2\langle v,\mathcal{C}v\rangle \\ &- \frac{\kappa^2(1-\frac{\Delta t}{4})}{4(1+\frac{\Delta t}{4})^2}\Delta t^{3/2}\langle x,v\rangle + \frac{\kappa^4}{8(1+\frac{\Delta t}{4})^2}\Delta t^{5/2}\langle \mathcal{C}x,v\rangle \\ &- \frac{\kappa^2(1-\frac{\Delta t}{4})}{4(1+\frac{\Delta t}{4})^2}\Delta t^{3/2}\langle \mathcal{C}x,v\rangle + \frac{\kappa^4}{8(1+\frac{\Delta t}{4})^2}\Delta t^{5/2}\langle \mathcal{C}^2x,v\rangle. \end{aligned} \tag{3.0.9}$$

In the case when we use step-size $\Delta t = c/\ell^2$, using the analytical calculations in (3.0.6) we see that all terms above have bounded L_1-norm, thus, obtaining immediately that $\sup_\ell |R(x,v)|_{L_1} < \infty$. It now remains to show that when $\Delta t = c/\ell^\epsilon$ with $\epsilon \in (0,2)$ then $E[a] \to 0$. First we use (3.0.5) and the analytical calculations in (3.0.6). To do this we need to identify the term J with the largest L_1-norm, which is the following:

$$J = -\frac{\kappa^6}{32(1+\frac{\Delta t}{4})^2}\Delta t^3\langle \mathcal{C}^2x,\mathcal{C}x\rangle.$$

In particular, using the calculations in (3.0.6) we find that:

$$|J|_{L_1} = \mathcal{O}(\Delta t^3\ell^6);\ |R-J|_{L_1} = \mathcal{O}(\Delta t^{5/2}\ell^5).$$

Again, the idea is to choose γ having an L_1-norm larger than $R-J$ but smaller than J. Indeed, in this case we apply (3.0.5) with the choice $\gamma = (\Delta t\ell^2)^{11/4}$. As with the case with RWM, and with this choice of γ the second and third terms in (3.0.5) will clearly vanish as $\ell \to \infty$. Some care is needed for the first term, for which we have that:

$$P[J \geq -2\gamma] \leq P\left[\frac{1}{\pi^2+\kappa^2}\frac{1}{\pi^6}\xi_1^2 \leq \frac{32(1+\Delta t/4)^2}{\kappa^6}\frac{1}{(\Delta t\ell^2)^{1/4}}\right] \to P[\xi_1^2 \leq 0] = 0.$$

This completes the proof. □

3.0.2 Analytical study for HMC

We now derive the corresponding results for HMC. First, we introduce some new notation. We rewrite the HMC leapfrog scheme Ψ_h as a transition matrix:

$$\Psi_h = \begin{pmatrix} \rho - (1-\rho)\kappa^2 \mathcal{C} & \sqrt{1-\rho^2} \\ -\frac{I-(\rho-(1-\rho)\kappa^2\mathcal{C})^2}{\sqrt{1-\rho^2}} & \rho-(1-\rho)\kappa^2\mathcal{C} \end{pmatrix}. \tag{3.0.10}$$

It is useful here to add a remark on the Karhunen-Loève expansion.

Remark 3.0.1. *Karhunen-Loève expansion for Brownian and OU Bridge: The Karhunen-Loève expansion (see Section 1.4) of the Gaussian distributions corresponding to the target OU-bridge and the reference Brownian bridge is used in this Section. In particular, we will use the orthonormal basis $\{\phi_p\}_{p=1}^{\infty}$ of $\mathcal{H}$ corresponding to the eigenfunctions of $\mathcal{C}^{bb}$ and make the standard correspondence $x \mapsto \{x_p\}_{p=1}^{\infty}$ between an element $x \in \mathcal{H}$ and it's squared summable co-ordinates $x_p = \langle x, \phi_p \rangle$ w.r.t. the basis $\{\phi_p\}$. In particular, recall from Section 1.4 that the eigen-structure $\{\lambda_p, \phi_p\}_{p=1}^{\infty}$ of $\mathcal{C}^{bb}$ is specified as follows:*

$$\lambda_p = \frac{\ell^2}{\pi^2 p^2} \ ; \quad \phi_p(u) = \sqrt{\frac{2}{\ell}}\,\sin(\frac{\pi p u}{\ell}). \tag{3.0.11}$$

Then, the Karhunen-Loève expansion of the two Gaussian distributions w.r.t. the above basis of sinusoidals is as below (see e.g. [83]):

$$\textit{BB:}\ \ x = \sum_{p=1}^{\infty} \frac{\ell}{\pi p}\,\xi_p\,\phi_p \ ; \quad \textit{OU Bridge:}\ \ x = \sum_{p=1}^{\infty} \frac{1}{\sqrt{\frac{\pi^2p^2}{\ell^2}+\kappa^2}}\,\xi_p\,\phi_p, \tag{3.0.12}$$

where $\{\xi_p\}_{p=1}^{\infty}$ are iid variables from $N(0,1)$.

Similarly, as for the RWM and MALA case, we can rewrite (3.0.10) in terms of the co-ordinates $\{x_p\}_{p=1}^{\infty}$ and $\{v_p\}_{p=1}^{\infty}$ of the complete paths x and v from their Karhunen-Loève expansion $x = \sum_{p=1}^{\infty} x_p\phi_p$ and $v = \sum_{p=1}^{\infty} v_p\phi_p$, then we can write the transition for each coordinate as:

$$\Psi_{h,p} = \begin{pmatrix} \rho - (1-\rho)\kappa^2 \lambda_p & \sqrt{1-\rho^2} \\ -\frac{1-(\rho-(1-\rho)\kappa^2\lambda_p)^2}{\sqrt{1-\rho^2}} & \rho-(1-\rho)\kappa^2\lambda_p \end{pmatrix} \tag{3.0.13}$$

where $\{\lambda_p = \ell^2/(\pi^2 p^2)\}_{p=1}^{\infty}$ are the eigen-values of $\mathcal{C}^{bb}$. Powers of the above matrix are determined by its eigen-structure. Therefore, we only consider the case when there are complex eigenvalues, i.e. when:

$$|\rho - (1-\rho)\kappa^2\lambda_p| < 1. \tag{3.0.14}$$

Since the Jacobian of the above matrix is unit, having eigenvalues greater than 1 will cause the powers to explode rendering the algorithm unstable. The above condition is equivalent to requiring that $(4 - \frac{c^2}{\ell^2} - 2\frac{2c^2\kappa^2}{p^2\pi^2})/(4 + \frac{c^2}{\ell^2})$ lie in $(-1, 1)$, which can be easily seen to be guaranteed, for any $l \geq l_0 > 0$ and for all $p \geq 1$, under the condition:

$$c\kappa < 2\pi.$$

Due to condition (3.0.14), it is reasonable to assume that a θ_p exists such that:

$$\begin{gathered}
\cos(\theta_p) = \rho - (1-\rho)\kappa^2\lambda_p; \quad \sin(\theta_p) = \sqrt{1 - \cos^2(\theta_p)}; \\
a_p = \frac{\sqrt{1-\rho^2}}{\sin(\theta_p)}.
\end{gathered} \tag{3.0.15}$$

Thus, (3.0.13) can be rewritten as:

$$\Psi_{h,p} = \begin{pmatrix} \cos(\theta_p) & a_p \sin(\theta_p) \\ -\frac{1}{a_p}\sin(\theta_p) & \cos(\theta_p) \end{pmatrix}. \tag{3.0.16}$$

Notice that if a_p was a constant that didn't depend on θ_p then (3.0.16) would correspond precisely to a clockwise elliptical rotation around the axis, where θ_p is the angle of rotation and a_p corresponds to the shape of the ellipsoid (see figure 3.1). In reality, this is not the exactly the case, but it serves as a good illustration of the effect of the leapfrog integration. Critically, representation (3.0.16) provides a mechanism for getting an analytical expression for the synthesis of several leapfrog steps. In particular, we have that:

$$\begin{aligned}
\Psi_{h,p}^{I} &= \begin{pmatrix} \cos(\theta_p) & a_p \sin(\theta_p) \\ -\frac{1}{a_p}\sin(\theta_p) & \cos(\theta_p) \end{pmatrix}^{I} \\
&= \begin{pmatrix} \cos(I\theta_p) & a_p \sin(I\theta_p) \\ -\frac{1}{a_p}\sin(I\theta_p) & \cos(I\theta_p) \end{pmatrix}.
\end{aligned} \tag{3.0.17}$$

We can now prove the following result:

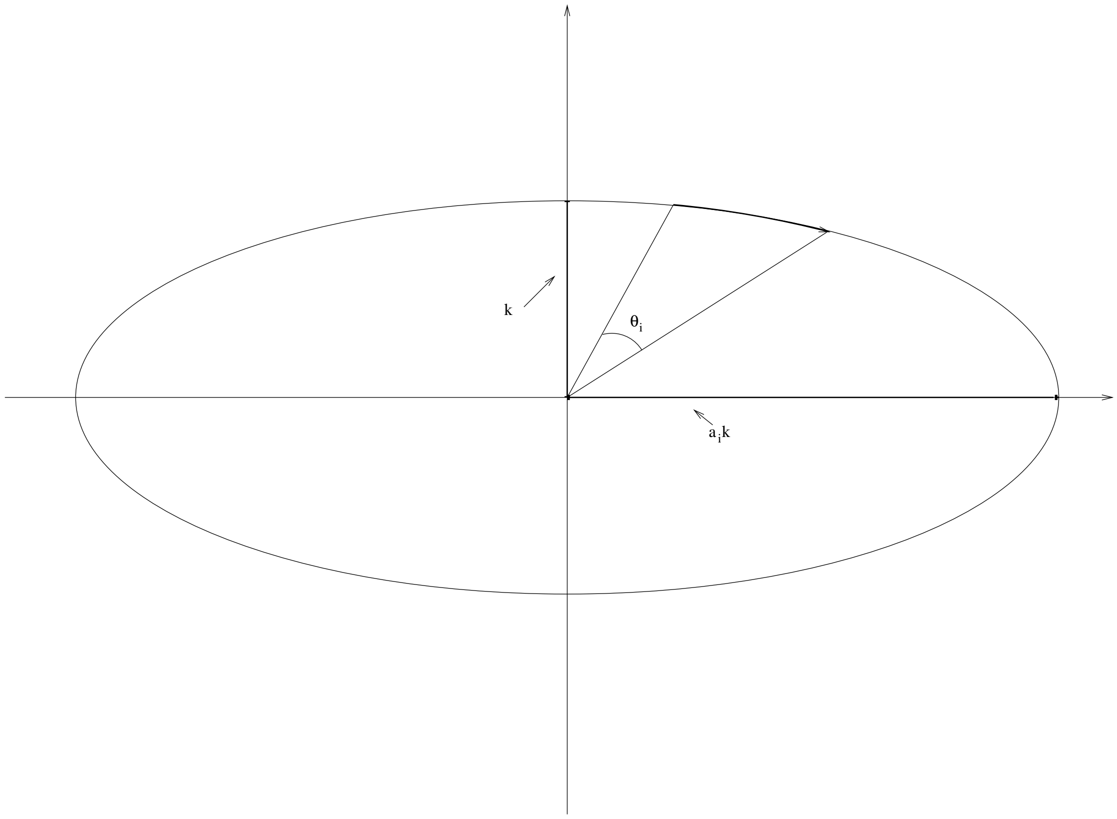

Figure 3.1: The Elliptical Rotation of a Point via Matrix (3.0.16)

Proposition 3.0.1. *Consider the advanced HMC algorithm (as described in Table 2.2) with target distribution the OU-bridge* Π *from* (3.0.2)*. If* $\alpha = \alpha(x, v)$ *is the acceptance probability of current position* x *and* $v \sim N(0, \mathcal{C})$*, then in stationarity* $(x \sim \Pi)$ *we have the following:*

If $h = c/\ell$ *with* $c\kappa < 2\pi$ *then* $\limsup_{\ell} E[\alpha] > 0$.

Proof. We exploit the representation of Ψ_h^I in (3.0.17). Recall that we denote $(x_i, v_i) = \Psi_h^i(x_0, v_0)$, for number of leapfrog steps $0 \le i \le I$. Given the particular context that Π is the OU-bridge in (3.0.1), we can rewrite the change of energy as follows:

$$\begin{aligned}\Delta H = H(x_I, v_I) - H(x_0, v_0) = \\ \tfrac{\kappa^2}{2}\langle x_I, x_I\rangle + \tfrac{1}{2}\langle x_I, \mathcal{C}^{-1}x_I\rangle + \tfrac{1}{2}\langle v_I, \mathcal{C}^{-1}v_I\rangle \\ - \tfrac{\kappa^2}{2}\langle x_0, x_0\rangle - \tfrac{1}{2}\langle x_0, \mathcal{C}^{-1}x_0\rangle - \tfrac{1}{2}\langle v_0, \mathcal{C}^{-1}x_0\rangle. \qquad (3.0.18)\end{aligned}$$

Clearly, using the derivation in (3.0.17), we get the following analytical expressions:

$$\begin{aligned} x_I &= \cos(I\theta)x_0 + a\sin(I\theta)v_0 \\ v_I &= -\tfrac{1}{a}\sin(I\theta)x_0 + \cos(I\theta)v_0, \end{aligned} \tag{3.0.19}$$

Notice that we have used in (3.0.19) the notation for operators and coordinates interchangeably. That is, $\sin(I\theta)$ is an operator such that we have $\sin(I\theta)x = \{\sin(I\theta_p)x_p\}_{p=1}^{\infty}$ where $\{x_p\}$ are the co-ordinates of $x \in \mathcal{H}$ w.r.t. to the orthonormal basis corresponding to the eigen-functions of $\mathcal{C}^{bb}$. The same interpretation can be used to explain the operation of $\sin(2I\theta)$ on elements of $\mathcal{H}$. Similarly $ax \equiv \{a_p x_p\}_{p=1}^{\infty} \equiv \sum_p a_p x_p \phi_p$. The sequences $\{\theta_p\}$ and $\{a_p\}$ have been defined in (3.0.15). and we can substitute (3.0.19) into (3.0.18) to get that:

$$\begin{aligned} \Delta H = {} & \langle (\tfrac{\kappa^2}{2}\cos^2(I\theta) + \tfrac{1}{2}\cos^2(I\theta)\mathcal{C}^{-1} + \tfrac{1}{2\alpha^2}\sin^2(I\theta)\mathcal{C}^{-1} - \tfrac{\kappa^2}{2} - \tfrac{1}{2}\mathcal{C}^{-1})x_0, x_0\rangle \\ & + \langle (\tfrac{\kappa^2}{2}\alpha^2\sin^2(I\theta) + \tfrac{1}{2}\alpha^2\sin^2(I\theta)\mathcal{C}^{-1} + \tfrac{1}{2}\cos^2(I\theta)\mathcal{C}^{-1} - \tfrac{1}{2}\mathcal{C}^{-1})v_0, v_0\rangle \\ & + \langle (\kappa^2\alpha\sin(I\theta)\cos(I\theta) + \alpha\cos(I\theta)\sin(I\theta)\mathcal{C}^{-1} - \tfrac{1}{\alpha}\sin(I\theta)\cos(I\theta)\mathcal{C}^{-1})x_0, v_0\rangle. \end{aligned} \tag{3.0.20}$$

After some calculations we get:

$$\Delta H(x_0, v_0) = H(x_I, v_I) - H(x_0, v_0) \equiv \langle \mathcal{A}x_0, x_0\rangle + \langle \mathcal{B}v_0, v_0\rangle + \langle \mathcal{G}x_0, v_0\rangle \tag{3.0.21}$$

for the operators (for convenience we set $\mathcal{C} \equiv \mathcal{C}^{bb}$):

$$\begin{aligned} &\mathcal{A} = -\tfrac{1}{2}\sin^2(I\theta)\mathcal{P};\ \mathcal{B} = \tfrac{1}{2}\sin^2(I\theta)a^2\mathcal{P};\ \mathcal{G} = \tfrac{1}{2}\sin(2I\theta)a\mathcal{P}; \\ &\qquad\qquad \mathcal{P} = \kappa^2 I + (1 - \tfrac{1}{a^2})(\mathcal{C})^{-1}. \end{aligned} \tag{3.0.22}$$

Now, if we denote by $\mathcal{C}^{OU}$ the covariance matrix of the OU target bridge, then the corresponding Karhunen-Loève expansion implies the eigenstructure $\{\lambda_{p,OU}, \phi_p\}_{p=1}^{\infty}$ for $\mathcal{C}^{OU}$ with eigen-values:

$$\lambda_{p,OU} = \frac{1}{\frac{\pi^2 p^2}{\ell^2} + \kappa^2}. \tag{3.0.23}$$

Plugging these eigen-values into (3.0.15) we get that:

$$a_p = \lambda_p^{-1/2}\lambda_{p,OU}^{1/2}c_p; \qquad c_p^2 = (\frac{1}{c^2\kappa^2} - \frac{1}{4p^2\pi^2})^{-1}. \tag{3.0.24}$$

Notice that the term c_p^2 is guaranteed to be positive by the condition $c\kappa < 2\pi$. In particular we have that:

$$c_p^2 \leq M, \tag{3.0.25}$$

for some constant $M > 0$. Using the above calculations into (3.0.22), and recalling the convention $\mathcal{P} \equiv (P_p)_{p=1}^{\infty}$, we get that:

$$\begin{aligned} P_p &= \frac{p^2\pi^2}{\ell^2} + \kappa^2 - \frac{(\frac{\ell^2}{p^2\pi^2})(\frac{p^2\pi^2}{\ell^2} + \kappa^2)}{c_p^2}\frac{p^2\pi^2}{\ell^2} \\ &= (1 - \tfrac{1}{c_p^2})\frac{p^2\pi^2}{\ell^2} + \kappa^2(1 - \tfrac{1}{c_p^2}) \\ &\equiv (\frac{p^2\pi^2}{\ell^2} + \kappa^2)(1 - \tfrac{1}{c_p^2}). \end{aligned} \tag{3.0.26}$$

It follows from (3.0.24) that:

$$0 \leq 1 - \frac{1}{c_p^2} \equiv \frac{h^2}{2 + \frac{h^2}{2}}\frac{\kappa^2\ell^2}{p^2\pi^2}\frac{1}{1+\rho}, \tag{3.0.27}$$

therefore,

$$P_p \equiv (\frac{p^2\pi^2}{\ell^2} + \kappa^2)\frac{\kappa^2 h^2 \ell^2}{(2 + \frac{h^2}{2})(1+\rho)}\frac{1}{p^2\pi^2}. \tag{3.0.28}$$

The latter, can be equivalently re-expressed as the operator:

$$\mathcal{P} \equiv \mathcal{C}_{OU}^{-1}\,\mathcal{C}\,\frac{\kappa^2\,h^2}{(2+\frac{h^2}{2})(1+\rho)}$$

so that:

$$0 \leq P_p \leq M\,\lambda_{p,OU}^{-1}\,\lambda_p\,\frac{1}{\ell^2}\ , \tag{3.0.29}$$

for some constant $M > 0$. Taking squares in (3.0.21), we have that:

$$\begin{aligned} E\,[\,(\Delta H)^2\,] = E\,[\,\langle \mathcal{A}x_0, x_0\rangle^2\,] + E\,[\,\langle \mathcal{B}v_0, v_0\rangle^2\,] \\ + E\,[\,\langle \mathcal{G}x_0, v_0\rangle^2\,] + 2\,E\,[\,\langle \mathcal{A}x_0, x_0\rangle\,]\,E\,[\,\langle \mathcal{B}v_0, v_0\rangle\,] \end{aligned} \tag{3.0.30}$$

since the rest of the expectations will be equal to zero. Henceforth, $\{A_i\}_{i=1}^{\infty}$, $\{B_i\}_{i=1}^{\infty}$, $\{G_i\}_{i=1}^{\infty}$ denote the eigenvalues of the operators $\mathcal{A}$, $\mathcal{B}$, $\mathcal{G}$ respectively. Recalling that $\langle \mathcal{A}x_0, x_0\rangle = \sum_{i=1}^{\infty} A_i x_{0,i}^2$, we have:

$$E[\langle \mathcal{A}x_0, x_0\rangle^2] = Var[\langle \mathcal{A}x_0, x_0\rangle] + E^2[\langle \mathcal{A}x_0, x_0\rangle] \tag{3.0.31}$$

and, in more detail we have that:

$$\begin{aligned} Var[\langle \mathcal{A} x_0, x_0\rangle] &= Var\left(\sum_{i=1}^{\infty} A_i x_{0,i}^2\right) = \sum_{i=1}^{\infty} A_i^2 Var\left(x_{0,i}^2\right) \\ &= \sum_{i=1}^{\infty} A_i^2 \lambda_{OU,i}^2 Var\left(z_i^2\right) = 2\sum_{i=1}^{\infty} A_i^2 \lambda_{OU,i}^2 \qquad (3.0.32) \end{aligned}$$

where, z_i are iid from $N(0,1)$. This gives the result:

$$E[\langle \mathcal{A} x_0, x_0\rangle^2] = 2\sum_{i=1}^{\infty} A_i^2 \lambda_{OU,i}^2 + \sum_{i=1}^{\infty} \left(A_i \lambda_{OU,i}\right).s \qquad (3.0.33)$$

Using similar calculations for the rest of the terms in (3.0.30), we obtain that:

$$\begin{aligned} E\,[\,(\Delta H)^2\,] = 2\,\sum_{i=1}^{\infty} A_i^2\,\lambda_{i,OU}^2 + 2\,\sum_{i=1}^{\infty} B_i^2\,\lambda_i^2 + \Big(\,\sum_{i=1}^{\infty}\big(\,A_i\,\lambda_{i,OU} + B_i\,\lambda_i\,\big)\Big)^2 \\ + \sum_{i=1}^{\infty} G_i^2\,\lambda_{i,OU}\,\lambda_i. \qquad (3.0.34) \end{aligned}$$

It remains to show that all involved terms above are upper bounded to complete the proof. We have that:

$$\sum_{i=1}^{\infty} A_i^2\,\lambda_{i,OU}^2 \le M\,\sum_{i=1}^{\infty}\,\lambda_{i,OU}^{-2}\,\lambda_i^2\,\frac{1}{\ell^4}\,\lambda_{i,OU}^2 \;= M\,\sum_{i=1}^{\infty}\frac{1}{i^4\,\pi^4} < \infty\ .$$

Using similar calculations we get that:

$$\sum_{i=1}^{\infty} B_i^2\,\lambda_i^2 \le M\,\sum_{i=1}^{\infty} \lambda_i^{-2}\,\lambda_{i,OU}^2\,c_i^4\,\lambda_{i,OU}^{-2}\,\lambda_i^2\,\frac{1}{\ell^4}\,\lambda_i^2 \le M\,\sum_{i=1}^{\infty}\frac{1}{i^4\,\pi^4} < \infty\ ,$$

and:

$$\begin{aligned} \sum_{i=1}^{\infty} G_i^2\,\lambda_{i,OU}\,\lambda_i &\le M\,\sum_{i=1}^{\infty}\,\lambda_i^{-1}\,\lambda_{i,OU}\,c_i^2\,\lambda_{i,OU}^{-2}\,\lambda_i^2\,\frac{1}{\ell^4}\,\lambda_{i,OU}\,\lambda_i \\ &\le M\,\sum_{i=1}^{\infty}\frac{1}{i^4\,\pi^4} < \infty\ . \end{aligned}$$

Finally, we turn to the third term on the right-hand side of (3.0.34) and we have the last term. For this term it is simpler to take the absolute value, instead of the square, and use the analytical expression of A_i and B_i. We have that:

$$\begin{aligned}
\big|\sum_{i=1}^{\infty}\big(A_i\,\lambda_{i,OU}+B_i\,\lambda_i\big)\big| &= \tfrac{1}{2}\,\big|\sum_{i=1}^{\infty}\sin^2(I\theta_i)\,P_i\,(-\lambda_{i,OU}+a_i^2\,\lambda_i)\big| \\
&= \tfrac{1}{2}\,\big|\sum_{i=1}^{\infty}\sin^2(I\theta_i)\,P_i\,\lambda_{i,OU}\,(c_i^2-1)\big| \\
&\leq M\sum_{i=1}^{\infty}\lambda_i\,\frac{1}{\ell^2} = M\sum_{i=1}^{\infty}\frac{1}{i^2\pi^2} < \infty\ .
\end{aligned}$$

Hence, we have shown that $\sup_{\ell} E\,[\,\Delta H^2\,] < \infty$, which as shown in (3.0.4), is sufficient for completing the proof. □

Remark 3.0.2. *We can now make some informal arguments to connect the above step-sizes, that control the average acceptance probability for the advanced RWM, MALA and HMC algorithms, with their mixing times –and as stated in (3.0.3) it involves their inverses. We are now going to consider the effect of the proposal of each algorithm for increasing ℓ on a fixed time-window of a path, say on $[0,\ell_0]$ for some $\ell_0>0$. For HMC, the synthesis of $I=\lfloor\frac{T}{h}\rfloor$ leapfrog steps will give a proposal moving the whole sub-path on $[0,\ell_0]$ an $\mathcal{O}(1)$-distance within it's state space. To show that, we ignore for a moment the effect of the nonlinear map $\Xi_{h/2}$ at the the leapfrog update in (2.2.11) and focus on the synthesis of I linear maps $\tilde{\Xi}_{h^*}$. This gives:*

$$\tilde{\Xi}_{h^*}^{I} = \begin{pmatrix} \cos(Ih^*) & \sin(Ih^*) \\ -\sin(Ih^*) & \cos(Ih^*) \end{pmatrix} \longrightarrow \begin{pmatrix} \cos(T) & \sin(T) \\ -\sin(T) & \cos(T) \end{pmatrix}\ , \quad \textit{as } \ell\to\infty\ .$$

The effect of the nonlinear operator $\Xi_{h/2}$ does not have such a simple interpretation, but it should not offset the main effect of proposals making $\mathcal{O}(1)$-steps from a current position, for an arbitrarily large ℓ. Thus, as a function of ℓ, the mixing time for advanced HMC only corresponds to the order of the number of leapfrog steps, $\mathcal{O}(\ell)$. For advanced RWM, shown in Table 2.2, for $\Delta t = c/\ell^2$ we can express the proposal as:

$$x^* = (1+\mathcal{O}(\ell^{-2}))\,x + \tfrac{\sqrt{c}}{\ell}\,(1+\mathcal{O}(\ell^{-2}))\,\xi\ . \tag{3.0.35}$$

Here, due to the random walk nature of the proposal, the algorithm will have to synthesize $\mathcal{O}(\ell^2)$-steps to move $\mathcal{O}(1)$-distance from a current position for a fixed point of the sub-path in $(0,\ell_0]$, thus the $\mathcal{O}(\ell^2)$-mixing time. Finally, for MALA, we have to refer to the interpretation of the algorithm as a discretization of an SDE on the pathspace, as expressed in (2.3.1). Without being too rigorous here, advanced

MALA essentially carries out steps of size $\Delta t = \mathcal{O}(\ell^{-2})$ along the continuous-time dynamics, thus, it will require $1/\Delta t = \mathcal{O}(\ell^2)$ steps to propagate a point of the sub-path on $[0, \ell_0]$ an $\mathcal{O}(1)$-distance from it's current position.

Of course, a rigorous analysis of mixing times would involve characterising the eigenvalues of the Markov chains, but this is beyond the scope of this thesis.

3.1 Numerical Illustration

In this Section, we employ the advanced algorithms of Table 2.1 to perform various simulation experiments involving diffusion bridges, stochastic volatility and latent diffusion survival models. In all these experiments, we treat the involved model parameters as known and focus on the update of the latent diffusion path. The aim is to assess and compare the performance of the algorithms on various aspects, including efficiency of the MCMC output and central processor unit (CPU) time. To measure CPU time in two different computing environments, the simulations for diffusion bridges and stochastic volatility models were carried out in MATLAB, whereas for the latent diffusion survival models the `C` programming language was used. The measure used to compare algorithms is the Effective Sample Size whose derivation is detailed in [40]. ESS can be interpreted as a measure of the equivalent size of independent samples corresponding to the dependent sample obtained from the MCMC simulation. It is calculated as follows:

$$ESS = \frac{N}{1 + 2\sum_k \gamma(k)}, \tag{3.1.1}$$

where, N is the number of posterior samples and $\sum_k \gamma(k)$ is the sum of the first k sample autocorrelation, where k is a suitably chosen truncation point [1]. Intuitively we notice that samples that are completely independent will have an ESS equal to the posterior sample size, and samples that are completely dependent will have an ESS equal to 1. A similar approach was also taken in [42] where the minimum ESS, taken over a number of univariate MCMC trajectories, was used. In our context, the MCMC performance is assessed by monitoring the posterior draws of the diffusion, recorded at a fine partition of its path, and reporting the minimum ESS over these points. The number k was set to a high enough value so that the minimum ESS, for a large enough number of iterations (set to 100,000), stabilises for all algorithms. The value of ESS was multiplied by a factor of 100 to reflect the

[1] The use of ESS is frequent in the MCMC literature, see e.g. [40]

percentage of the total MCMC iterations that can be considered as independent draws from the posterior.

The MCMC algorithms employed consist of an Independent Sampler proposing from the reference Brownian path $\tilde{\Pi}$ and the advanced algorithms in Table 2.1. The algorithms were tuned to achieve certain acceptance probability levels that, according to our experience and previous literature, are associated with better performance. Specifically, we aimed to attain an acceptance probability[2] around (15% - 30%) for RWM, (50%-70%) for MALA and (65%-85%) for HMC. To explore the performance of HMC we first fixed the number of leapfrog steps (e.g. to 5 or 10) and then recorded the minimum ESS for various levels of acceptance probability. We then considered cases with additional leapfrog steps. For each of these algorithms, we monitor the values of the minimum ESS, CPU times and their ratio in absolute and relative terms. The results herewith presented contain the best version of these algorithms.

3.1.1 Diffusions Observed at a Discrete Skeleton

Consider the diffusion discussed in Section 3.0.1, i.e. an OU process with SDE:

$$dX_t = -\kappa X_t dt + dB_t \ , \quad 0 \leq t \leq \ell \ ,$$

with $X_0 = 0$ and an observation at time $\ell = 1$. We set $X_1 = 0$ and consider 3 different values for κ, i.e. $12, 20, 30$ in our investigation of the MCMC performance. The MCMC components comprise of the equidistant points from a discrete skeleton of the diffusion. The discretization step was set to $\delta = 0.02$. Table 3.1 provides the results, i.e. the values of the minimum ESS, CPU times and their absolute and relative ratio. The HMC algorithm consisted of 5 leapfrog steps with the parameter h set to values $(0.43, 0.26, 0.17)$ for values of κ equal to $(12, 20, 30)$ respectively. For advanced MALA, that can be thought as HMC with a single leapfrog step, the corresponding values of $h = \sqrt{\Delta t}$ were very similar $(0.45, 0.26, 0.18)$ indicating much smaller total steps. Overall, advanced HMC consistently overperforms, in terms of ESS, the other algorithms. In particular, for $\kappa = 30$, HMC is faster than the Independent Sampler by a factor of over 30. Its performance remains at high levels as we increase κ and does not deteriorate as δ becomes smaller, as indicated by the results obtained for $\delta = 0.01$ and $\delta = 0.005$. In line with the results of [83] and Section 3.0.2, we notice a substantial improvement over advanced

[2] The optimal acceptance probability was selected based on the following research, for RWM and MALA see [41] and for HMC see [6]

MALA suggesting a more efficient use of the gradient within HMC. MALA offers some improvement over RWM and Independent Sampler, but at a heavy additional computational cost. The Independent Sampler performs reasonably well for $\kappa = 12$ (acceptance rate of 16%) but its performance drops substantially as κ increases and the acceptance rate becomes smaller; 8% for $\kappa = 20$ and 1.2% for $\kappa = 30$.

$\kappa = 12$	min(ESS)	time	$\frac{\text{min(ESS)}}{\text{time}}$	relative $\frac{\text{min(ESS)}}{\text{time}}$
IS	3.9173	4.8811	0.8025	2.1733
RWM	3.9584	5.8925	0.6718	1.8192
MALA	4.0112	10.8626	0.3693	1
HMC ($\delta = 0.02$)	35.7274	20.8695	1.7119	4.6361
HMC ($\delta = 0.01$)	35.8903	32.5594	1.1023	2.9848
HMC ($\delta = 0.005$)	35.5875	51.6085	0.6895	1.8670
$\kappa = 20$	min(ESS)	time	$\frac{\text{min(ESS)}}{\text{time}}$	relative $\frac{\text{min(ESS)}}{\text{time}}$
IS	0.5013	4.4977	0.1115	1
RWM	1.0086	5.4445	0.1853	1.6621
MALA	1.6202	10.0588	0.1611	1.4452
HMC	26.6214	20.8841	1.2747	11.4369
$\kappa = 30$	min(ESS)	time	$\frac{\text{min(ESS)}}{\text{time}}$	relative $\frac{\text{min(ESS)}}{\text{time}}$
IS	0.1012	4.7043	0.0215	1
RWM	0.4343	5.7229	0.0759	3.5277
MALA	0.5372	10.0438	0.0535	2.4863
HMC	13.3350	20.4831	0.6510	30.2631

Table 3.1: Relative efficiency via the minimum ESS (%) and CPU times (seconds), for the advanced pathspace algorithms - Case of OU bridges. IS denotes the Independent Sampler.

3.1.2 Stochastic Volatility Models

The following stochastic volatility model was used to simulate data:

$$\begin{cases} dS_t = \exp(V_t/2)dB_t \ , \quad 0 \le t \le \ell \ ; \\ dV_t = \kappa(\mu - V_t)dt + \sigma dW_t \ . \end{cases}$$

The parameters were set according to previous analyses based on similar models for the S&P500 dataset [18]. Specifically, we set $\kappa = 0.03$, $\mu = 0.07$, $\sigma^2 = 0.03$

Sampler	min(ESS)	time	$100\times\ \frac{\min(\text{ESS})}{\text{time}}$	relative $\frac{\min(\text{ESS})}{\text{time}}$
RWM	0.1400	161.8298	0.0865	1.3561
MALA	0.2181	341.8737	0.0638	1.0000
HMC (5 steps)	2.5695	689.6767	0.3726	5.8400
HMC (10 steps)	8.1655	1188.1201	0.6873	10.7729
HMC (20 steps)	8.3216	2200.1311	0.3782	5.9288

Table 3.2: Relative efficiency, via the minimum ESS (%) and CPU times (seconds) for the diffusion pathspace algorithms - Case of stochastic volatility paths.

and $V_0 = 0$. We considered about a year measured in days ($\ell = 250$) and recorded observations at a daily frequency (250 data points). The transformation of V_t to a unit volatility diffusion was utilised to write the target density and construct the HMC algorithms. The model for a pair of consecutive observations, (y_{i-1}, y_i) can be written as:

$$\begin{cases} y_i | y_{i-1} \ \sim \ N\big(y_{i-1},\ \int_{t_{i-1}}^{t_i} \exp(\sigma x_s) ds\big) \ ; \\ dX_t = \kappa \left(\frac{\mu}{\sigma} - X_t\right) dt + dW_t \ , \quad t_0 \le t \le t_1 \ . \end{cases}$$

The results are shown in Table 3.2. The Independent sampler performs very poorly in this case, with an acceptance rate below 10^{-4}, and is omitted from the table. MALA provides a small improvement over RWM which is nevertheless not enough to cover the associated increase in the corresponding computations. Nevertheless, this is not the case for HMC that reaches its optimal performance roughly at 10 leapfrog steps. Advanced HMC offers a considerable improvement, being nearly 8 times faster than RWM and 11 times faster than MALA. Parameter h, that corresponded to the desired acceptance probability levels, was 0.085 for the MALA algorithm and 0.075 for all the versions of the HMC algorithm.

3.1.3 Latent Diffusion Survival models

This Section provides a numerical illustration of simulated data from a latent diffusion survival model appearing in [70]. Survival models target the probability of an individual i surviving up to time u or else $P(Y > u)$, where Y denotes the event time. The aim is to model the hazard function $h(u)$ that reflects the probability that an event will occur in the infinitesimal period $[u, u + du)$ by retrieving information from available data in the form of event times. Latent diffusion survival models [1, 70] provide parametric formulations for $h(u)$, which is assumed to be a positive function $h(\cdot)$ of a diffusion process $x = x(s)$. The motivation in such models is to consider an underlying process that results in the occurrence of each event [1]. The distribution function for a single observation y_i is given as follows:

$$F\big(y_i|x\big) = 1 - \exp\big(- \int_0^{y_i} h(x(s))ds\big) \ , \quad 0 < y_i \leq \ell \ ,$$

with density:

$$f\big(y_i|x\big) = h(x(y_i)) \exp\Big(- \int_0^{y_i} h(x(s))ds\Big) \ , \quad 0 < y_i \leq \ell \ .$$

The likelihood for the observed event times $y = (y_1, \ldots, y_n)$, with $\max_i y_i \leq \ell$, can be written as:

$$f\big(y|x\big) = \Big[\prod_{i=1}^{n} h(x(y_i))\Big] \exp\Big(- \sum_{i=1}^{n} \int_0^{y_i} h(x(s))ds\Big) \ . \tag{3.1.2}$$

For ease of exposition we assume that $x(s)$ corresponds to a diffusion with unit coefficient and drift function $\nu(x)$. Hence, the log-density $\log((d\Pi/d\Pi_0)(x|y))$ for the latent diffusion[3] x becomes (up to an additive normalising constant):

$$\sum_{i=1}^{n} \Big\{ \log h\left(x(y_i)\right) - \int_0^{y_i} h(x(s))ds\Big\} \\ + \int_0^{\ell} \nu(x(s))dx(s) - \frac{1}{2}\int_0^{\ell} \nu^2(x(s))ds$$

where Π_0 denotes the distribution of a standard Brownian motion. Specifically, we assume the scenario that the underlying diffusion process is specified as follows:

$$dX_t = -(1.4\sin(X_t)dt + 1)dt + dB_t \ , \quad X_0 = 2 \ .$$

so that the likelihood for event times $Y = \{Y_1, \ldots, Y_n\}$ is given by:

$$p\big(y|\eta^{-1}(x)\big) = \Big[\prod_{i=1}^{n} x_{y_i}^2\Big] \exp\Big(- \sum_{i=1}^{n} \int_0^{y_i} x_s^2 ds\Big).$$

Table 3.3 provides the measures of performance for the algorithms[4] in Table

[3] For more information about such models, including cases of censored data, the reader is referred to [70]

[4] The calculations in this Section were obtained using the `C` programming language, for the

Sampler	min(ESS)	time	$100\times \frac{\text{min(ESS)}}{\text{time}}$	relative $\frac{\text{min(ESS)}}{\text{time}}$
RWM ($\delta = 0.01$)	0.1039	55.2342	0.1881	1
MALA ($\delta = 0.2$)	0.6466	87.5021	0.7389	3.9284
HMC ($\delta = 0.15$)	25.2985	248.0301	10.1997	54.2229

Table 3.3: Relative efficiency, via the minimum ESS (%) and CPU times (seconds) for the advanced pathspace algorithms - Case of latent diffusion survival model.

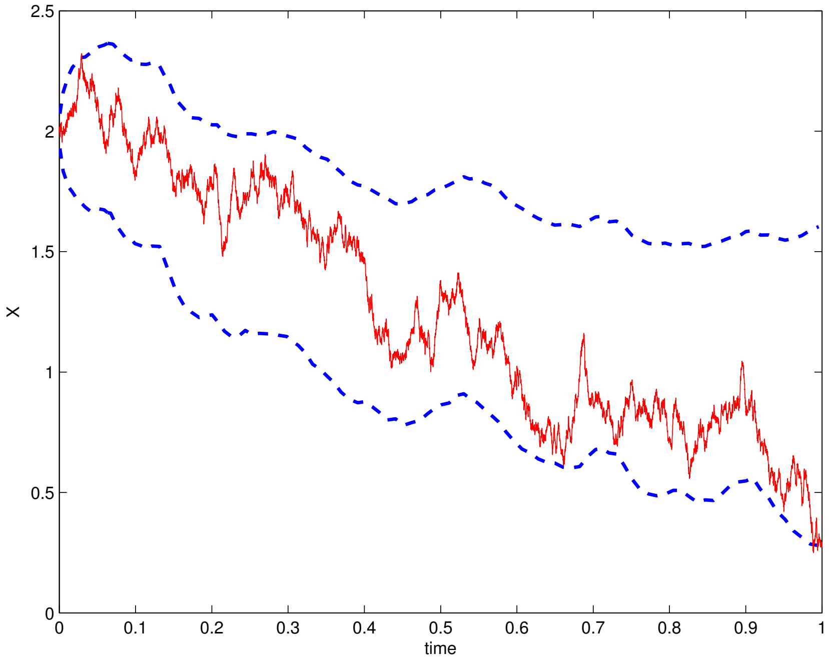

Figure 3.2: 95% Pointwise credible intervals (blue dashed lines) overlayed on the true path of X (red solid line).

2.1. Similar to the stochastic volatility simulation experiment, the independent sampler is associated with an extremely low acceptance rate, thus, rendering it unfeasible in practice. RWM also performs poorly. A very small step is required to achieve the desired acceptance rate, thus, resulting in very small moves around the diffusion pathspace. MALA with $h = 0.2$ performs better in this case, but a massively better performance is achieved by the advanced HMC. Specifically, HMC with 10 leapfrog steps and $h = 0.15$ is about 54 times faster than RWM. Figure 3.2 depicts the trajectory of X_t, determining the hazard function that was used to generate the data. The figure displays 95% pointwise credible intervals obtained from the HMC algorithm appearing in Table 3.3.

previous two applications we used MATLAB

3.2 Calculation of $\mathcal{C}\delta\Phi(x)$

Hitherto, for the derivation of advanced MALA and HMC, we have taken for granted that $\mathcal{C}\delta\Phi(x)$ is an element of the Cameron-Martin space[5] of the reference Gaussian measure $\tilde{\Pi} = N(0,\mathcal{C})$. In this Section we verify this assumption for a large class of target distributions, therefore, demonstrating that these advanced methods are well-defined for typical SDE-driven models.

Motivated by the expression of the negative log-density arising in the application in Section 3.1 we will carry out calculations assuming the following general form:

$$\Phi(x) = \alpha(x(t_1), x(t_2), \ldots, x(t_M)) + \beta(I_1, I_2, \ldots, I_L) + \gamma(S_1, S_2, \ldots, S_J) \quad (3.2.1)$$

where we have set:

$$I_l = \int_0^\ell z_l(s, x(s))ds \;\; , 1 \le l \le L \;; \quad S_j = \int_0^\ell r_j(s, x(s))dx(s) \;\; , 1 \le j \le J \; ,$$

for positive integers M, L, J, times $t_1 < t_2 < \cdots < t_M$ in $[0,\ell]$ that could be determined by some data Y and functions $\alpha, \beta, \gamma, z_l, r_j$ determined via the particular model. All applications in Section 3.1 correspond to particular instances of this generic structure. Here, the target posterior distribution $\Pi(dx)$ is defined on the Hilbert space of squared integrable paths $\mathcal{H} = L^2([0,\ell],\mathbb{R})$ (with appropriate boundary conditions). The centered Gaussian reference measure Π_0 corresponds to a Brownian motion (thus, boundary condition $x(0) = 0$) or a Brownian Bridge ($x(0) = x(\ell) = 0$). Recall here the specification of the covariance operators $\mathcal{C}^{bm}$, $\mathcal{C}^{bb}$ and Cameron-Martin spaces $\mathcal{H}_0^{bm}$, $\mathcal{H}_0^{bb}$ of a Brownian motion and Brownian bridge respectively in Section 1.4. We make the following definitions, for the relevant range of subscripts:

$$\alpha_m = \frac{\partial \alpha}{\partial x_m}(x_{t_1}, x_{t_2}, \ldots, x_{t_M}) \;; \quad \beta_l = \frac{\partial \beta}{\partial I_l}(I_1, I_2, \ldots, I_L) \;;$$
$$\gamma_j = \frac{\partial \gamma}{\partial S_j}(S_1, S_2, \ldots, S_J) \;; \quad z_l' = \frac{\partial z_l}{\partial x} \;; \quad r_j' = \frac{\partial r_j}{\partial x} \; .$$

Remark 3.2.1. *With a somewhat abuse of notation, path-elements $\{\mathcal{C}^{bm}\delta\Phi(x)\}$, $\{\mathcal{C}^{bb}\delta\Phi(x)\}$ found in Proposition 3.2.1 below are obtained (at least for the terms in $\Phi(x)$ involving stochastic integrals) by recognising that the N-dimensional al-*

[5] See Assumption 2.2.1

gorithm used in practice, after using finite-difference, corresponds to applying the finite-difference scheme on the Hilbert-space-valued algorithm employing precisely the shown paths $\{\mathcal{C}^{bm}\delta\Phi(x)\}$ and $\{\mathcal{C}^{bb}\delta\Phi(x)\}$ within its specification. (So, here $\delta\Phi(x)$ corresponds to a variational derivative only formally.)[6]

Proposition 3.2.1. *For the functional $\Phi(x)$ given in (3.2.1), for any $x \in \mathcal{H}$:*

$$\begin{aligned}\big(\mathcal{C}^{bm}\delta\Phi(x)\big)(u) &= \sum_{m=1}^{M}\alpha_m \cdot \big(\, u\,\mathbb{I}[\,u < t_m\,] + t_i\,\mathbb{I}[\,u \geq t_m\,]\,\big) \\ &+ \sum_{l=1}^{L}\beta_l \cdot \Big(u\int_0^{\ell} z_l'(v, x(v))dv \; - \int_0^u\int_0^s z_l'(v, x(v))dv\, ds\Big) \\ &+ \sum_{j=1}^{J}\gamma_j \cdot \Big(u\,\big(\,r_j(\ell, x(\ell)) + \int_0^{\ell} dq_j(v)\,\big) - \int_0^u\int_0^s dq_j(v)\,ds\Big)\ , \quad u \in [0,\ell]\ ,\end{aligned}$$

for the integrator

$$dq_j(v) = r_j'(v, x(v))dx(v) - dr_j(v, x(v))\ .$$

Also:

$$\begin{aligned}\big(\mathcal{C}^{bb}\delta\Phi(x)\big)(u) &= \sum_{m=1}^{M}\alpha_m \cdot \big(\, u\,\mathbb{I}[\,u < t_i\,] + t_i\,\mathbb{I}[\,u \geq t_i\,] - u\,t_i/\ell\,\big) \\ &+ \sum_{l=1}^{L}\beta_l \cdot \Big(\frac{u}{\ell}\int_0^{\ell}\int_0^s z_l'(v, x(v))dv\,ds - \int_0^u\int_0^s z_l'(v, x(v))dv\,ds\Big) \\ &+ \sum_{j=1}^{J}\gamma_j \cdot \Big(\frac{u}{\ell}\int_0^{\ell}\int_0^s dq_j(v)\,ds - \int_0^u\int_0^s dq_j(v)\,ds\Big)\ , \qquad u \in [0,\ell]\ .\end{aligned}$$

Proof. We use the analytical expressions for $\mathcal{C}^{bm}$, $\mathcal{C}^{bb}$ given in (1.4.1) and (1.4.2) respectively. For the first term in the expression for Φ, namely $\alpha = \alpha(x(t_1), x(t_2), \ldots, x(t_M))$, we can formally write:

$$(\delta\alpha)(s) = \sum_{m=1}^{M}\alpha_m \cdot \delta_{t_i}(ds),$$

where δ_{t_i} is the Dirac measure centered at t_i. Applying $\mathcal{C}^{bm}$ and $\mathcal{C}^{bb}$ will give immediately the terms in the first lines of the expression for $\mathcal{C}^{bm}\,\delta\Phi(x)$ and $\mathcal{C}^{bm}\,\delta\Phi(x)$ in the statement of the proposition. For the second term $\beta = \beta(I_1, I_2, \ldots, I_L)$, we

[6] This remark applies also to a similar result shown in Proposition 4.1.1 in the next Chapter

have the variational derivative:

$$(\delta\beta)(s) = \sum_{l=1}^{L} \beta_l \cdot z_l'(s, x(s)).$$

Again, applying $\mathcal{C}^{bm}$ and $\mathcal{C}^{bb}$ will give the terms in the second lines of the expression for $\mathcal{C}^{bm}\,\delta\Phi(x)$ and $\mathcal{C}^{bm}\,\delta\Phi(x)$ in the statement of the proposition.

We proceed to the term $\gamma = \gamma(S_1, S_2, \ldots, S_J)$ with the stochastic integrals. The algorithm applied in practice will involve a finite-difference approximation of the stochastic integrals $\{S_j\}$. Down below we sometimes sacrifice accuracy of notation to avoid taking too much space for what otherwise involve straightforward derivative calculations. Consider the discretized time instances $0 = s_0 < s_1 < \cdots s_{N-1} < s_N = \ell$, denoting three consecutive discrete time instances among the above by $s_- < s < s_+$, the finite-difference approximation, say S_j^N, of S_j can be written as follows:

$$S_j^N = \sum_{s \in \{s_1, \ldots, s_N\}} r_j(s_-, x(s_-))(x(s) - x(s_-)).$$

We can now calculate the partial derivative of S_j w.r.t. to the one of the N variables, $x(s)$, making up the discretized path. Notice that $x(s)$ will appear in two terms in the summation, unless it is the last point $x(s_N)$ of the x-vector when it will only appear once. This explains the following calculation of the partial derivatives:

$$\frac{\partial S_j}{\partial x(s)} = \Delta q(s)\ ,\quad s \in \{s_1, \ldots, s_{N-1}\}\ ;\quad \frac{\partial S_j}{\partial x(s_N)} = r_j(s_{N-1}, x(s_{N-1})), \tag{3.2.2}$$

where we have defined:

$$\Delta q(s) = r'(s, x(s))(x(s_+) - x(s)) - (r_j(s, x(s) - r_j(s_-, x(s_-))).$$

Overall, we have that:

$$\frac{\partial \gamma}{\partial x(s)} = \sum_{j=1}^{J} \gamma_j \cdot \frac{\partial S_j^N}{\partial x(s)}. \tag{3.2.3}$$

Then, the $N \times N$ discretized covariance operator $C^{bm} = (\min\{s_i, s_k\})_{i,k}$, corresponding to the covariance matrix of a standard Brownian motion at the time instances $s_1, s_2, \ldots, s_N$ (this is the discretized version of $\mathcal{C}^{bm}$), can easily be shown to apply to a finite-dimensional vector $f \in \mathbb{R}^N$ as follows:

$$(C^{bm} f)_u = s_u \cdot \big(\sum_{i=1}^{N} f_i \big) - \sum_{k=1}^{u-1} \big(\sum_{i=1}^{k} f_i \big)\, \Delta s_{k+1}, \quad u = 1, 2, \ldots, N, \tag{3.2.4}$$

where $\Delta s_{k+1} = s_{k+1} - s_k$. Combining (3.2.2), (3.2.3) and, (3.2.4) will give (with ∇ denoting the gradient) the following:

$$(C^{bm} \nabla\gamma)_u =$$
$$\sum_{j=1}^{J} \gamma_j \cdot \Big(s_u \big(r_j(s_{N-1}, x(s_{N-1})) + \sum_{i=1}^{N-1} \Delta q(s_i) \big) - \sum_{k=1}^{u-1} \big(\sum_{i=1}^{k} \Delta q(s_i) \big) \Delta s_{k+1} \Big)$$

which now can be recognised as the finite-difference discretization of the term appearing in the third line of the expression for $\mathcal{C}^{bm}\delta\Phi(x)$ in the statement of the proposition. A similar approach for the Brownian bridge case, considering the discrete time instances $0 = s_0 < s_1 < \cdots s_{N-1} < s_N < s_{N+1} = \ell$, and the corresponding N-dimensional matrix C^{bb} is represented as below:

$$(C^{bb} f)_u = \tfrac{s_u}{\ell} \cdot \sum_{k=1}^{N} \big(\sum_{i=1}^{k} f_i \big)\, \Delta s_{k+1} - \sum_{k=1}^{u-1} \big(\sum_{i=1}^{k} f_i \big)\, \Delta s_{k+1}, \ u = 1, \ldots, N, \tag{3.2.5}$$

where now:

$$S_j^N = \sum_{s \in \{s_1, \ldots, s_{N+1}\}} r_j(s_-, x(s_-))(x(s) - x(s_-)); \quad \frac{\partial S_j}{\partial x(s_i)} = \Delta q(s_i),$$

where $\Delta q(s)$ is as defined earlier and $1 \leq i \leq N$. This gives the following calculation:

$$(C^{bb}\, \nabla\gamma)_u = \sum_{j=1}^{J} \gamma_j \cdot \Big(\frac{s_u}{\ell} \big(\sum_{k=1}^{N} \big(\sum_{i=1}^{k} \Delta q(s_i) \big) \Delta s_{k+1} - \sum_{k=1}^{u-1} \big(\sum_{i=1}^{k} \Delta q(s_i) \big) \Delta s_{k+1} \Big),$$

immediately recognised as the finite-difference discretization of the term appearing in the third line of the expression for $\mathcal{C}^{bb}\delta\Phi(x)$ in the statement of the proposition.

□

Thus, in both cases the first terms appearing in the specification of $\{\mathcal{C}^{bm}\delta\Phi(x)\}$ and $\{\mathcal{C}^{bb}\delta\Phi(x)\}$ in the proposition, are continuous and piece-wise linear in u (there is a turn at the time instances of the observations) so still lie within the Cameron-Martin spaces $\mathcal{H}_0^{bm}$, $\mathcal{H}_0^{bb}$ respectively (even if the variational derivative $\delta\alpha$ itself will not necessarily lie within the Hilbert space, as shown in the proof). The second terms are clearly a.s. elements of the corresponding spaces $\mathcal{H}_0^{bm}$, $\mathcal{H}_0^{bb}$ under weak continuity conditions on z_l'. Finally, for the third terms, again weak regulatory conditions on r_j and r_j' guarantee that the corresponding paths in u are elements of the appropriate Cameron-Martin spaces.

3.3 Conclusions: Future Work

In this chapter we have studied the relative efficiency and well-posedness of HMC, as well as its performance in a variety of applications. We have also compared, analytically and with numerical examples, the computational costs of different advanced MCMC methods. We have shown that both– advanced Random Walk Metropolis (RWM) and Metropolis-adjusted Langevin Algorithm (MALA)– have similar complexity when applied to 'long' diffusion paths, whereas HMC is more efficient than both of them. These desirable properties make HMC an ideal algorithm to be used for parameter inference.

In this Section we provide a simple illustration of an application of parameter inference with HMC in the form of a Metropolis-within-Gibbs sampler. In contrast with Section 3.1, here we try to infer unknown model parameters. In a later chapter we will be using a similar framework to perform parameter inference on more complex diffusion models driven by fractional Brownian motion. As said before, this Section is only meant to serve as a simple illustration on how to treat the infinite-dimensional diffusion paths as a latent variable within a data augmentation framework[7], this is: when the main interest lies in identifying other model parameters.

Consider the following example where the stochastic processes of interest (X_t, q_t) are defined via the bivariate SDE:

$$dX_t = \sigma_\theta(q_t)dB_t \ , \tag{3.3.1}$$

$$dq_t = \mu_\theta(q_t)dt + dW_t \ , \quad t \in [0, \ell] \ , \tag{3.3.2}$$

where B_t and W_t represent independent standard Wiener processes, for appropriate mappings $\sigma_\theta(q_t), \mu_\theta(q_t)$ involving some parameter θ. Now assume we have observed X_t at a collection of discrete time instances, so that we have the data:

$$Y = \{X_{t_0}, X_{t_1}, X_{t_2}, ..., X_{t_n}\} \ ,$$

[7] See e.g. [80]

for some $t_0 < t_1 < t_2 < \cdots < t_n = \ell$ and $n \geq 1$. This regime corresponds to a case of a stochastic volatility model. In this example we cannot observe directly the process q_t, instead, we only observe the data Y. Inference of the parameters θ, given the information in Y, is not straightforward due to the unavailability of the conditional density $p(Y|\theta)$, thus, we can approach this problem via data augmentation and a carefully designed Metropolis-within-Gibbs sampler.

To summarise, we want to sample from the posterior distribution the parameters we wish to draw inference from (θ in this case):

$$p(\theta \mid Y) \propto p(Y|\theta)p(\theta) \ . \tag{3.3.3}$$

Using a Gibbs sampler, we repeat the following steps for $i = 0, 1, \ldots$:

$$\begin{aligned} &\text{Step 1: } \theta^{(i+1)} \sim p(\theta|q^{(i)}, Y) \ , \\ &\text{Step 2: } q^{(i+1)} \sim p(q|\theta^{(i+1)}, Y) \ , \end{aligned} \tag{3.3.4}$$

so that, after a sufficiently large burn-in period, then the samples of θ should come from the target posterior distribution (see [41]). That is: to perform the Gibbs sampler we start off from the analytically available joint distribution:

$$p(Y, q, \theta) = p(Y|q, \theta)\, p(q|\theta, q_{t_0})\, p(q_{t_0}|\theta)\, p(\theta) \tag{3.3.5}$$

so that we simply have:

$$p(\theta|q, Y) \propto p(Y, q, \theta) \ ; \quad p(q|\theta, Y) \propto p(Y, q, \theta) \tag{3.3.6}$$

which is invoked within the Gibbs approach to sample θ and q. In particular, we have that:

$$\begin{aligned} p(Y|q, \theta) &= \prod_{i=1}^{n} p(X_{t_i} \mid X_{t_{i-1}}, q, \theta) \\ &= c \times \prod_{i=1}^{n} I_i^{-\frac{1}{2}} \cdot e^{-\frac{(X_{t_i} - X_{t_{i-1}})^2}{2I_i}} \end{aligned} \tag{3.3.7}$$

for a constant $c > 0$ not depending on q, θ, where I_i is obtained via Ito's isometry (see [65]) as:

$$I_i = \int_{t_{i-1}}^{t_i} \sigma_\theta^2(q_s) ds \ . \tag{3.3.8}$$

Then, the density $p(q \mid \theta, q_{t_0})$ can be obtained as a Radon-Nikodym derivative with respect to a Gaussian measure. That is, using Girsanov's theorem we can formally write:

$$p(q \mid \theta, q_{t_0}) = c \times \exp\{-\Phi(q)\} \times e^{-\frac{1}{2}\langle (q-m), L(q-m)\rangle} \tag{3.3.9}$$

for L the inverse covariance of the Brownian motion on $[0,\ell]$ and m the constant path equal to q_{t_0} throughout $[0,\ell]$, where we have set:

$$\Phi(q) = M_\theta(q_0) - M_\theta(q_{t_n}) + \int_{t_0}^{t_n} g_\theta(q_s)ds \tag{3.3.10}$$

for:

$$M_\theta(v) = \int^v \mu_\theta(u)du \ ; \quad g_\theta(v) = \tfrac{1}{2}(\mu_\theta^2 + \mu_\theta')(v) \ . \tag{3.3.11}$$

For $p(q_{t_0} \mid \theta)$, one common choice is that it is specified as the equilibrium distribution of q, though, other choices could be considered.

Now we turn our attention to the first step of (3.3.4), that is, sampling $p(\theta \mid q, Y)$. A standard choice here involves considering a RWM step:

- Set $\theta' = \theta^{(i)} + \xi$ where, for a user specified h, $\xi \sim N(0,h)$;
- Accept/reject so that $\theta^{(i+1)} = \theta'$ w.p. $\alpha(\theta', \theta^{(i)}) = 1 \wedge \frac{p(\theta', Y, q)}{p(\theta^{(i)}, Y, q)}$.

In the second step of the Gibbs sampler we are interested in sampling the high-dimensional volatility path from $p(q|\theta, Y)$, and at this point we can use one of our advanced MCMC algorithms. One choice would be to update the whole q-path over $[0,\ell]$ simultaneously, using, say, an HMC sampler. If we take a simplistic approach we can update q as a single continuous path starting from q_{t_0} using an HMC and we could implement (3.3.4). For long time intervals, it may be beneficial to split up the q-path into a series of overlapping blocks and update each one separately– a strategy known as blocking (see [69]). With the exception of the last block, every single block has a conditioned starting and ending point, so we will treat those as diffusion bridges. The reason for using an overlapping block is to guarantee that all points of q get updates after a full sweep. We can summarise the sequence of steps for the block update as follows: for a block-size user parameter $l = (t_n - t_0)/k$ for some $k \geq 1$:

1. Initiate q_{t_0}.
2. Update q-bridge with fixed points q_{t_0} to q_{t_0+l}.
3. Update next bridge from $q_{t_0+l/2}$ to $q_{t_0+3l/2}$

4. Repeat for all overlapping bridges by incrementing initial time by $l/2$.

5. Update path with starting point $q_{t_n - l}$ until time t_n.

To be precise about the full conditional distribution for each relevant block, let us consider an arbitrary bridge with starting point at time S and ending point at T, i.e. $q_{S:T}$. Then, clearly, the full conditional distribution is as follows:

$$\begin{aligned} p(q_{S:T}|q_{else},\theta,Y) &\propto p(q_{S:T},q_{else},\theta,Y) \\ &\propto p(q_{S:T}|q_S,q_T,\theta)p(Y|q,\theta) \ . \end{aligned} \tag{3.3.12}$$

where we now have:

$$p(q_{S:T}|q_S,q_T,\theta) \propto \exp\{-\Phi(q_{S:T})\} \times e^{-\frac{1}{2}\left\langle (q_{S:T}-m_{S:T}),\, L_{S:T}(q_{S:T}-m_{S:T})\right\rangle} \tag{3.3.13}$$

where:

$$\Phi(q_{S:T}) = \int_S^T g_\theta(q_u)du - M_\theta(q_{t_n})\,\mathbb{I}\,[\,T = t_n\,] \tag{3.3.14}$$

with $m_{S:T}$ the straight line between q_S, q_T and $L_{S:T}$ the inverse covariance of a Brownian bridge from starting point 0 to ending point 0 of lenght $T-S$ (in the single case $T = t_n$, $m_{S:T}$ is the line which is always equal to q_S over $[S,T]$, and $L_{S:T}$ the inverse covariance of a standard Brownian motion on a time interval of length $T-S$).

As seen in previous chapters, our advanced MCMC methods are relevant for target distributions defined as a change of measure from a reference Gaussian law, as is the case in (3.3.12). We can simplify the expression for the target distribution as follows:

$$\begin{aligned} p(q_{S:T}\ |q_{else},\theta,Y) &\propto p(q_{S:T}|q_S,q_T,\theta)\,p(Y|q,\theta) \\ &= c_\theta \times \exp\{-\mathcal{G}(q_{S:T})\} \times e^{-\frac{1}{2}\left\langle (q_{S:T}-m_{S:T}),\, L_{S:T}(q_{S:T}-m_{S:T})\right\rangle} \end{aligned} \tag{3.3.15}$$

where we have set:

$$\mathcal{G}(q_{S:T}) = \Phi(q_{S:T}) + \sum_{i:[t_{i-1},t_i]\cap[S,T]\neq\emptyset} \left(\tfrac{(X_{t_i}-X_{t_{i-1}})^2}{2I_i} + \tfrac{1}{2}\log I_i \right) \tag{3.3.16}$$

We now have all the necessary parts to run the advanced MCMC algorithms.

As mentioned before, we have defined this algorithm as an illustration and we will not run it in this section. We return to the topic of Parameter Inference in Chapter 5, where we use a similar sampling scheme for a more complex model. In the sequel, blocking as described in this section, is no longer possible and HMC's superior scaling properties will become more relevant.

Chapter 4

Advanced Algorithms for Processes with General Diffusion Coefficient

In this chapter, we consider the general case where the diffusion process of interest is defined via the d-dimensional SDE (for some $d \geq 1$):

$$dX_t = b(X_t;\theta)dt + \sigma(X_t;\theta)dW_t, \quad t \in [0, \ell] \tag{4.0.1}$$

where $\sigma = \sigma(\cdot;\theta) : \mathbb{R}^d \mapsto \mathbb{R}^{d\times d}$ is a non-constant matrix diffusion coefficient. So far, we have presented advanced MCMC algorithms for models driven by SDEs, in the context of a constant diffusion coefficient σ, or for cases when the SDE of interest can be transformed into one of constant diffusion coefficient. We will review such transformations here, and look in particular for approaches that can be relevant beyond a scalar context.

In the context of a non-scalar diffusion X_t, defined by the equation in (4.0.1), it is not guaranteed that X can be transformed into an SDE of unit diffusion coefficient. Indeed, the Lamperti transform in such a multivariate context would look at the existence of a mapping $X_t \mapsto \tilde{X}_t = \eta(X_t)$ (with $\eta = (\eta_1, \eta_2, \ldots, \eta_d)^\top : \mathbb{R}^d \mapsto \mathbb{R}^d$) such that, for all x in the state space of X_t:

$$D\eta(x) \cdot \sigma(x) = I_d \tag{4.0.2}$$

where we have defined the $d \times d$ matrix of partial derivatives $D\eta = (\partial\eta_i/\partial x_j)_{i,j=1}^d$. This follows directly from the multivariate version of Itô's formula, see e.g. [2]. Ait-Sahalia's work in [2] also shows the existence of a mapping η where the property in (4.0.2) is equivalent to the diffusion coefficient matrix satisfying $\partial\sigma_{ij}^{-1}/\partial x_k = \partial\sigma_{ik}/\partial x_j$ for all i, j, k with $j < k$. This certainly restricts considerably the applicability of the Lamperti transform in the cases of non-scalar diffusions.

It would be relatively simple to come up with models for where it does not exist an appropriate mapping η that solves the differential equation (4.0.2). The Heston model, originally described in [49], is a mathematical model commonly used to describe the joint evolution of the price and volatility of an underlying asset. Heston's model corresponds to a stochastic volatility model that is described by the following bivariate SDE:

$$\begin{aligned} dS_t &= \mu S_t dt + \sqrt{\nu_t} S_t \, dW_t^S , \\ d\nu_t &= \kappa(\theta - \nu_t) \, dt + \xi \sqrt{\nu_t} \, dW_t^\nu \, , \end{aligned} \tag{4.0.3}$$

with S_t, ν_t denoting the price and volatility processes respectively, W_t^S, W_t^ν the relative Wiener processes driving the SDE (assumed independent here, though a leverage effect could also be considered) and μ, κ, θ, ξ being appropriate model parameters. Then, one can easily work with Ait-Sahalia's condition to show that for this model there is no solution to equation (4.0.2).

There are, however, other methods suggested in the literature with a wider scope for transforming a diffusion model of multiplicative noise into a distribution which can be expressed as a change of measure from a Gaussian law. Indeed, our advanced MCMC algorithms are relevant for posterior distributions on pathspace which are absolutely continuous w.r.t. Brownian motion related distributions, and it is of interest to verify the well-definition of such algorithms on the pathspace when using such alternative transforms.

4.1 Beyond the Lamperti Transform

The method introduced in [18] maps the process of interest X_t onto the driving Wiener noise $\tilde{X}_t = W_t$ of the SDE, similarly to the Wilkinson-Golightly transformation we described earlier in section 1.2.2. Assuming some relevant data Y with conditional likelihood $p(y|x)$, and since the prior on $\tilde{X}$ is simply the Wiener measure $\Pi_0 = N(0, \mathcal{C}^{bm})$, we can write the posterior distribution on $\tilde{X}$ as:

$$\frac{d\Pi}{d\Pi_0}(\tilde{x}) \propto p(y|x) =: \exp\{-\Phi(\tilde{x})\} \, .$$

Thus, via an application of Bayes theorem and the transform considered, we have obtained a target distribution which is within the class of distributions that can be tackled by our advanced MCMC samplers. It remains to calculate $\mathcal{C}^{bm}\delta\Phi(\tilde{x})$ in this context and show that with probability 1 in $\tilde{x}$ this path lies within the Cameron-Martin space of the reference Wiener measure[1]. Differentiation of $\Phi(\tilde{x})$ will involve finding derivatives of x w.r.t. the driving noise $\tilde{x}$. So it is not a surprise that the dynamics of the so-called Malliavin derivative $D_s X_t$ (see e.g. [37]) will appear in the calculations as they are precisely meant to describe changes in the process under small changes in the driving noise. More particularly, $D_s X_t$ expresses the rate of change of the process X at time t when the driving noise changes at time $s < t$ and, following [37]), we have that $D_s X_t$ is analytically defined as follows (ignoring model parameters):

$$\begin{aligned} \tfrac{dY_t}{Y_t} &= b'(X_t)dt + \sigma'(X_t)\,dW_t\ , \quad Y_0 = 1\ ; \\ D_s X_t &= \tfrac{Y_t}{Y_s}\,\sigma(X_s)\,\mathbb{I}\,[\,s \le t\,]\ . \end{aligned}$$

We will assume here a scalar setting and the following general structure for[2] $\Phi(\tilde{x}) = -\log(p(y|x))$ for appropriate mappings α, β, $z_1, z_2, \ldots, z_L$:

$$\begin{aligned} \Phi(\tilde{x}) = \Phi_x(x(\tilde{x})) &= \alpha(x(t_1), x(t_2), \ldots, x(t_M)) \\ &\quad + \beta\Big(\int_0^\ell z_1(s, x(s))ds,\ \int_0^\ell z_2(s, x(s))ds,\ \int_0^\ell z_L(s, x(s))ds\Big)\ . \end{aligned} \tag{4.1.1}$$

The terms α_m, β_l appearing below correspond to partial derivatives of the functionals α, β (w.r.t. the m-th and l-th argument respectively) as in the case of Proposition 3.2.1.

Proposition 4.1.1. *For the functional $\Phi(\tilde{x})$ given in (4.1.1), for any $\tilde{x} \in \mathcal{H}$ we have the following expression:*

$$\begin{aligned} \big(\mathcal{C}^{bm}\delta\Phi(\tilde{x})\big)(u) &= \sum_{m=1}^{M} \alpha_m \cdot \Big(\,(u \wedge t_m)\,(F_{m,t_m} + \sigma(x_{t_m})) - \int_0^{u \wedge t_m} F_{m,r}dr\Big) \\ &\quad + \sum_{l=1}^{L} \beta_l \cdot \Big(\,u\,(G_{l,\ell} + J_{l,\ell}) + \int_0^u (G_{l,r} + J_{l,r})dr\Big)\ , \quad u \in [0, \ell]\ , \end{aligned}$$

for the processes, for $m = 1, 2, \ldots, M$ and $l = 1, 2, \ldots, L$:

[1] Recall from Section 1.4 that this corresponds to showing that $\mathcal{C}^{bm}\,\delta\Phi(\tilde{x})$ is a weakly differentiable mapping on $[0, \ell]$

[2] Compared with the structure assumed earlier in (3.2.1) we do not include here stochastic integral terms to avoid excessively cumbersome expressions

$$F_{m,r} = \int_0^r e^{\int_s^{t_m} \left(\mu'(x_u)du+\sigma'(x_u)d\tilde{x}_u\right)}\, dQ_s \ ;$$
$$G_{l,r} = \int_0^r \int_s^{\ell} z_l'(t,x_t)\, e^{\int_s^{t} \left(\mu'(x_u)du+\sigma'(x_u)d\tilde{x}_u\right)} dt\, dQ_s \ ;$$
$$J_{l,r} = \int_0^r z_l'(s,x_s)\, \sigma(x_s) ds \ ,$$

with integrator:

$$dQ_s = \sigma(x_s)(b'(x_s)du + \sigma'(x_s)d\tilde{x}_s) - d\sigma(x_s) \ .$$

Focusing on the properties of the calculated path $\mathcal{C}^{bm}\delta\Phi(\tilde{x})$ over its domain of definition $u \in [0,\ell]$, it is easy to check the following: a.s. in $\tilde{x}$, the first terms in the expression obtained above are continuous, piece-wise linear with points of non-differentiability at the data instances $t_1, t_2, \ldots, t_M$. Then, under the weak assumption that the processes $r \mapsto G_{l,r}$, $r \mapsto J_{l,r}$ are a.s. continuous, we have that the second terms in the calculation in Proposition 4.1.1 are a.s. differentiable. Thus, under weak conditions Assumption 2.2.1 requiring that $\mathcal{C}^{bm}\delta\Phi(\tilde{x})$ be in the Cameron-Martin space of the reference Gaussian measure is satisfied and advanced MALA and HMC are well-defined on the pathspace in the present context.

Proof. Consider a collection of discrete time instances $0 < s_1 < s_2 < \cdots < s_N$ with $s_0 = 0$ and $s_N = \ell$ that include the data instances, so that:

$$\{t_1, t_2, \ldots, t_M\} \subset \{s_1, s_2, \ldots, s_N\} \ .$$

Let $\Delta s_i = s_i - s_{i-1}$. We will consider the following finite-difference approximation $\Phi(\tilde{x}) = \Phi(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N)$ of the negative log-density:

$$\begin{aligned}\Phi(\tilde{x}) = \Phi_1(\tilde{x}) + \Phi_2(\tilde{x}) = \alpha(x_{i_1}, x_{i_2}, \ldots, x_{i_M}) \qquad \qquad (4.1.2)\\ + \beta\Big(\sum_{i=1}^{N} z_1(s_{i-1},x_{i-1})\Delta s_i, \sum_{i=1}^{N} z_2(s_{i-1},x_{i-1})\Delta s_i, \ldots, \sum_{i=1}^{N} z_L(s_{i-1},x_{i-1})\Delta s_i\Big)\end{aligned}$$

for indices $i_1, i_2, \ldots i_M$ such that $s_{i_m} = t_m$, for $m = 1, 2, \ldots M$, and vector x constructed via the finite-difference approximation:

$$x_i = x_{i-1} + b(x_{i-1})\Delta s_i + \sigma(x_{i-1})\Delta\tilde{x}_i \ ,$$

for $i = 1, 2, \ldots, N$ with x_0 equal to a specified fixed initial condition. We will be using the obtained expression in (3.2.4) for the $N \times N$ covariance matrix $C = C^{bm} = (\min\{s_i, s_j\})_{i,j}$ of a standard brownian motion at the time instances $s_1, s_2, \ldots, s_N$. The function $\Phi : \mathbb{R}^N \mapsto \mathbb{R}$ in (4.1.2) and the matrix C fully specify a finite-difference approximation of the original target defined on the Hilbert space.

Now, we have the following recursion for the derivatives

$$Y_{i,j} = \frac{\partial x_i}{\partial \tilde{x}_j} \ ,$$

for any $j \geq 1$:

$$\begin{aligned}
Y_{i,j} &= Y_{i-1,j} + b'(x_{i-1})\, Y_{i-1,j} \Delta s_i + \sigma'(x_{i-1})\, Y_{i-1,j}\, \Delta\tilde{x}_i \ ; \quad i > j+1 \\
Y_{j+1,j} &= Y_{j,j} + b'(x_j)\, Y_{j,j} \Delta s_{j+1} + \sigma'(x_{i-1})\, Y_{j,j}\, \Delta\tilde{x}_{j+1} - \sigma(x_j) \ ; \\
Y_{j,j} &= \sigma(x_{j-1}) \ ; \\
Y_{i,j} &= 0 \ , \quad i < j \ .
\end{aligned}$$

So, we can obtain that, for $i > j+1$:

$$\log(Y_{i,j}) = \log(Y_{i-1,j}) + \log\left(1 + b'(x_{i-1})\Delta s_i + \sigma'(x_{i-1})\Delta\tilde{x}_i\right) \ ,$$

and using this recursion we get that:

$$\begin{aligned}
Y_{i,j} &= \Delta Q_j \times e^{\sum_{k=j+2}^{i} \log\left(1+b'(x_{k-1})\Delta s_k + \sigma'(x_{k-1})\Delta\tilde{x}_k\right)} \ , \quad i \geq j+1 \ , && (4.1.3) \\
Y_{j,j} &= \sigma(x_{j-1}) && (4.1.4) \\
Y_{i,j} &= 0 \ , \quad i < j \ , && (4.1.5)
\end{aligned}$$

where we have set:

$$\Delta Q_j \equiv Y_{j+1,j} = \sigma(x_{j-1})(b'(x_j)\Delta s_{j+1} + \sigma'(x_j)\Delta\tilde{x}_{j+1}) - \Delta\sigma(x_j) \ ,$$

and $\Delta\sigma(x_j) = \sigma(x_j) - \sigma(x_{j-1})$. We will also define for $1 \leq m \leq M$ and $1 \leq l \leq L$:

$$\begin{aligned}
F_{m,r} &= \sum_{j=1}^{r} Y_{i_m,j} \ , \quad r < i_m \ ; \\
G_{l,r} &= \sum_{j=1}^{r} \Big(\sum_{i \geq j+1} z_l'(s_i, x_i)\Delta s_{i+1} \Big)\, Y_{i,j} \ ; \\
J_{l,r} &= \sum_{j=1}^{r} z_l'(s_j, x_j) Y_{j,j}\, \Delta s_{j+1} \ .
\end{aligned}$$

The above sequences will appear in the calculation of partial derivatives of $\Phi(\tilde{x})$. It is important to notice here that these sequences indeed constitute a finite-difference approximation of their continuous-time counterparts appearing at the statement of the proposition: to see that one only needs to look at the analytical definition of $Y_{i,j}$ in equations (4.1.3)-(4.1.4), and realise that the sum $\sum_k \log\big(1 + b'(x_{k-1})\Delta s_k + \sigma'(x_{k-1})\Delta\tilde{x}_k\big)$ is essentially a finite-difference approximation of $\int(b'(x_u)du + \sigma'(x_u)d\tilde{x}_u)$ as for $\epsilon \approx 0$ we have that $\log(1+\epsilon) \approx \epsilon$.

We can now proceed with the calculation of the partial derivatives of Φ. Clearly:

$$\frac{\partial\Phi}{\partial\tilde{x}_j} = \frac{\partial\Phi_1}{\partial\tilde{x}_j} + \frac{\partial\Phi_2}{\partial\tilde{x}_j} = \sum_{i\geq j}\big(\frac{\partial\Phi_1}{\partial x_i}\cdot Y_{i,j} + \frac{\partial\Phi_2}{\partial x_i}\cdot Y_{i,j}\big)\ .$$

We can easily get:

$$\sum_{i\geq j}\frac{\partial\Phi_1}{\partial x_i}\cdot Y_{i,j} = \sum_{m=1}^{M}\alpha_m Y_{i_m,j}\ .$$

Using (3.2.4), a long but otherwise straightforward calculation will give that, for vector index $1 \leq u \leq N$:

$$(C\,\nabla\Phi_1(\tilde{x}))_u = \sum_{m=1}^{M}\alpha_m\left(s_{u\wedge i_m}\left(F_{m,i_m-1} + Y_{i_m,i_m}\right) - \sum_{k=1}^{u\wedge i_m-1}F_{m,k}\,\Delta s_{k+1}\right)\ . \quad (4.1.6)$$

Proceeding to the second term, $\Phi_2(\tilde{x})$, we have that:

$$\sum_{i\geq j}\frac{\partial\Phi_1}{\partial x_i}\cdot Y_{i,j} = \sum_{l=1}^{L}\beta_l\sum_{i\geq j}z_l'(s_i,x_i)\Delta s_{i+1}\,Y_{i,j}\ .$$

We now multiply with the covariance matrix C to obtain, after some calculations, for $1 \leq u \leq N$:

$$(C\,\nabla\Phi_2(\tilde{x}))_u = \sum_{l=1}^{L}\beta_l\left(\,s_u\,(\,G_{l,N} + J_{l,N}\,) - \sum_{k=1}^{u-1}(\,G_{l,k} + J_{l,k}\,)\,\Delta s_{k+1}\right)\ . \quad (4.1.7)$$

Upon inspection, (4.1.6)-(4.1.7) provide the proof of the statement of the proposition.

□

4.2 Advanced RWM in a Golightly-Wilkinson framework

We want to emphasize here that the approach described in Section 4.1 is applicable when, given the diffusion path x, there exists a non-trivial data likelihood $p(y|x)$ w.r.t. the Lebesque measure. So, for instance, this context does not cover the case of *directly* observed processes and an alternative approach will have to be followed. In particular, the method which is relevant in the case of directly observed processes is the Golightly-Wilkinson transformation. Due to the great importance of this data regime, we will illustrate this approach here in the context of using RWM, to simulate from an SDE with a non-constant diffusion coefficient.

In Section 1.2.2, we introduced the Golightly-Wilkinson transformation and discussed its advantage over the Lamperti transformation. As discussed above, and since we are now in the context of directly observed processes, that we can assume the following modeling scenario:

$$\begin{aligned} \Pi^{x,y}: \quad dX_t &= b(X_t;\theta)dt + \sigma(X_t;\theta)dW_t \ , \\ X_0 &= x \ , \quad X_\ell = y \ , \end{aligned} \tag{4.2.1}$$

where,

$$\begin{gathered} x, y \in \mathbb{R}^d, \quad b: \ \mathbb{R}^d \mapsto \mathbb{R}^d, \\ \sigma: \ \mathbb{R}^d \mapsto \mathbb{R}^{d\times d}, \end{gathered}$$

with the the direct observation $X_\ell = y$ giving rise to the target distribution $\Pi^{x,y}$ corresponding to a diffusion bridge. Recapping from Section 1.2.2, we develop an alternative diffusion bridge process which is easy to sample from by using the following SDE:

$$Q^{x,y}: \quad dX_t = \frac{y - X_t}{\ell - t}dt + \sigma(X_t;\theta)dW_t \ , \tag{4.2.2}$$

The particular choice of drift function in the above expression ensures that the diffusion processes is 'pushed' towards the terminal position y, when $t \to \ell$. In this context, we are required now to obtain the density $\frac{d\Pi^{x,y}}{dQ^{x,y}}$, which is given via the bridge version of Girsanov's theorem in (1.2.21).

As we have seen before, one way to think of the SDE in (4.2.2) is as a mapping which projects the Brownian motion path W_t onto an actual bridge X_t, i.e.

expression (4.2.2) gives rise to a map:

$$X = \Psi(W)\ .$$

The inverse of this mapping will also be relevant here. With X now referring to the target bridge from $\Pi^{x,y}$, we have the transformation:

$$\tilde{X} = \Psi^{-1}(X),$$

where the target $\tilde{X}$ is a process with a distribution that will be absolutely continuous w.r.t. the Wiener law of the Brownian motion path W. This mapping corresponds precisely to the Golightly-Wilkinson transformation.

Once we have obtained a target distribution, which is a change of measure from a Gaussian law, the complete machinery of our advanced MCMC methods becomes immediately relevant, in the form of RWM, MALA or HMC methods. Focusing on advanced RWM we can use the above transform to propose values in the standard advanced RWM approach:

$$\tilde{x}' = \rho\,\tilde{x} + \sqrt{1-\rho^2}\,\xi \tag{4.2.3}$$

for a step-size $\rho \in (0,1)$, where ξ is a standard Brownian motion path, $\tilde{x}$ is the current path in the MCMC sampler and $\tilde{x}'$ is the proposed path. As discussed in the presentation of advanced RWM earlier in the thesis, this particular structure of the proposal mechanism ensures that the law $\tilde{x}$ is absolutely continuous with that of the target distribution, so it will have a non-trivial acceptance probability on the infinite-dimensional pathspace.

Due to the above absolute continuity properties, we can use the same arguments for the acceptance probability as in Section 1.2.1 and Section 1.2.3 (specifically equation (1.2.21)). Analytically, from standard results on 1-1 transformations of probability measures we can find the relevant probability density involved in the acceptance probability. Let $\tilde{\Pi}^{x,y}$ denote the law of $\Psi^{-1}(X)$ for $X \sim \Pi^{x,y}$, and Π_0 the law of a standard Brownian motion on $[0,\ell]$. Then we have:

$$\frac{d\tilde{\Pi}^{x,y}}{d\Pi_0}(\tilde{X}) = \frac{d\Pi^{x,y}}{dQ^{x,y}}(\Psi(\tilde{X})) =: \mathcal{D}(\Psi(\tilde{X}')) \tag{4.2.4}$$

with the latter density being given via the diffusion bridge version of Girsanov's theorem in (1.2.20). Recall from the discussion over advanced RWM in Section 1.2.1, that the identification of the target distribution as a change of measure from a Gaussian law provides immediately the relevant Metropolis-Hastings acceptance probability. Indeed, we have that:

$$\alpha(\tilde{X}, \tilde{X}') = \min\left(1, \frac{\mathcal{D}(\Psi(\tilde{x}'))}{\mathcal{D}(\Psi(\tilde{x}))}\right). \tag{4.2.5}$$

This completes the algorithm, which we summarise in Table 4.1.

RWM, general σ	
Target diffusion	$dX_t = b(X_t;\theta)dt + \sigma(X_t;\theta)dW_t,\;$ for $X_0 = x,\ X_\ell = y$
Proposal	$\tilde{x}' = \rho\tilde{x} + \sqrt{1-\rho^2}\,\xi$
Acceptance probability	$\min\left(1, \frac{\mathcal{D}(\Psi(\tilde{x}'))}{\mathcal{D}(\Psi(\tilde{x}))}\right)$

Table 4.1: Advanced RWM, for target distribution that of a diffusion bridge with non-constant diffusion coefficient

4.2.1 Advanced RWM for a diffusion bridge with non-constant diffusion coefficient, explained with a numerical example

The algorithm we described in the previous section was defined for a d-dimensional SDE and uses a Wilkinson-Golightly mapping. Indeed, one of the advantages of this type of mapping versus the Lamperti transformation is that Wilkinson-Golightly only requires the existence of the inverse σ^{-1}. Since this condition is weaker, it can be used in a wider range of problems than the Lamperti transform.

In this section we provide a one dimensional example of the RWM algorithm as described in table 4.1. The example we provide is a Cox-Ingersoll-Ross (CIR) bridge, a diffusion first proposed in the context of modelling interest rates in [23]. In particular, we propose the following model:

$$dX_t = r(\mu - X_t)dt + \sigma\sqrt{X_t}\,dW_t\ ,$$
$$X_0 = x, \quad X_\ell = y\ .$$

Note 4.2.1. *We have selected a relatively simple example for our explanation since the Lamperti transform could have been used instead. Our objective is to illustrate how the Wilkinson-Golightly transform could be used in a simple context. In the next Chapter we will be using mappings in a more complex context where the Lamperti transform can not be used.*

We will be applying the advanced RWM algorithm on this model under the following specification of the CIR parameters: $r = 3$, $\mu = 4.6$, $X_0 = 3$, $X_l = 4$, $\ell = 1$. Figure 4.1 shows some diagnostic plots from the output of the advanced MCMC algorithm. Notice that the trace plot for the mid-point of the complete path looks reasonably good. Also, the burn-in time also seems to be relatively short, reaching equilibrium in about 100 iterations. The relevant ACF plot reaches 0 in about 80 lags, which is quite satisfactory. In summary, it seems that the algorithm is behaving as expected. We tried a finer resolution and still obtained a very similar graphic representation.

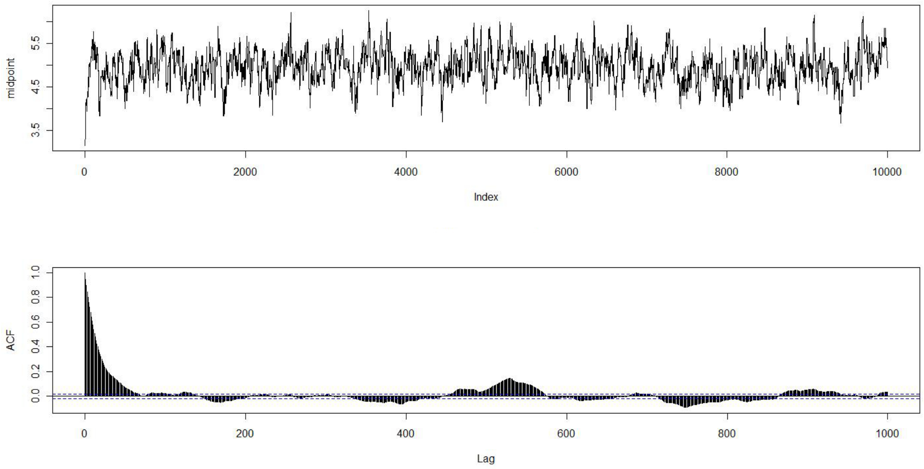

Figure 4.1: Diagnostics Plots for advanced RWM targeting a Cox-Ingersoll-Ross bridge. Top panel: Traceplot. Bottom Panel: Autocorrelation function.

4.3 Summary

In previous chapters, we have shown how SDEs with non-constant diffusion coefficient can be transformed to SDEs with unit diffusion coefficients. We have expanded on this in this chapter by showing that some clever use of mapping can be applied to a wider range of applications than the Lamperti when using advanced MCMC algorithms to sample from complex target distributions such as d-dimensional SDEs with general diffusion coefficients. Most importantly, we have shown that $\mathcal{C}^{bm}\delta\Phi(\tilde{x})$ is part of the Cameron-Martin space and that HMC and MALA are well behaved algorithms in this context.

As an example, we analytically adapted this mapping to work for an advanced RWM algorithm using a d-dimensional SDE as a target. Finally, we used this scheme numerically to simulate a Cox-Ingersoll model. We found that our scheme has good mixing and low autocorrelation despite not using a derivative driven method.

Chapter 5

Parameter Inference for SDEs Driven by Fractional Noise

5.1 Introduction to fBm

Originally introduced in [57], fractional Brownian Motion (fBm) is a continuous time Gaussian process. We will denote it here as $B^H = \{B_t^H; t \in [0, \ell]\}$, for some relevant length of time $\ell > 0$, with $H \in (0, 1)$ being the Hurst parameter specifying fBm. The main innovation introduced in fBm, as compared to regular Brownian motion, is the presence of the Hurst parameter H which describes the level of long range dependence for fBm values. In particular, the covariance function for B^H is specified as follows:

$$E[B_t^H B_s^H] = \tfrac{1}{2}\big(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\big) \ , \tag{5.1.1}$$

where we have that $B_0^H = 0$, and the mean is zero for all $t \geq 0$.

To quantify better the effect of the Hurst parameter H on the memory properties of fBm, we define the increments over time periods of length $\delta = \ell/N$, for some integer $N \geq 1$, as:

$$G(j) = B_{j\delta}^H - B_{(j-1)\delta}^H \ , \quad 1 \leq j \leq N \ , \tag{5.1.2}$$

so that we have the N-dimensional vector of Gaussian increments[1]:

$$G_N = \{G(j) : 1 \leq j \leq N\} \ . \tag{5.1.3}$$

Using (5.1.1), it is easy to check that fGn has the following autocovariances (for

[1] Such increments are sometimes given the name of fractional Gaussian noise (fGn) in the literature (for instance, in [66])

any integer $j, j_0 \geq 1$):

$$\gamma_\delta(j) := E[G(j + j_0)G(j_0)] = \frac{1}{2}\,\delta^{2H}(j+1)^{2H} + \frac{1}{2}\delta^{2H}(j-1)^{2H} - \delta^{2H}j^{2H}. \quad (5.1.4)$$

It is now easy to check that different values of H will have the following effect on the correlation of the increments of B^H:

- if $0 < H < \frac{1}{2}$, then the increments are negatively correlated;
- if $H = \frac{1}{2}$ then the increments are independent, and B^H is simply the Wiener process;
- if $\frac{1}{2} < H < 1$ then the increments are positively correlated (i.e. $\gamma_\delta(j) > 0$).

As mentioned above, when $H = 1/2$, fBm is simply a Brownian Motion (or Wiener process), thus fBm can be thought of as a generalization of the Brownian motion allowing for memory in its increments.

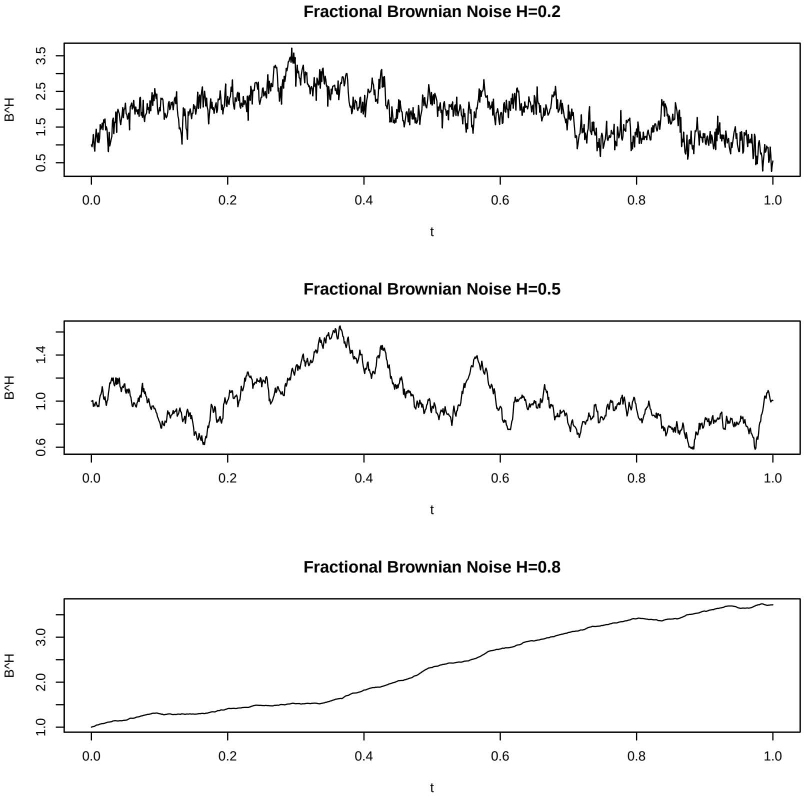

Figure 5.1: An illustration of the effect the Hurst coefficient has on fBm.

We will now detail some of interesting properties of fBm[2]:

[2] The proofs of the statements that follow can be found in [66]

Self-Similarity
Same as for the standard Brownian Motion, fBm is self similar. That is, for any $c > 0$:

$$\{B_t^H\}_{t\geq 0} = \{c^H B_{t/c}^H\}_{t\geq 0}.$$

Long-Range Dependence
A process, say X_t, is said to have long range dependence when the auto-covariance function $E[X_t X_{t+k}]$ (assuming X_t is of zero mean) decreases slow enough as $k \geq 1$ increases so that we have that:

$$\sum_{k=0}^{\infty} E[X_t X_{t+k}] = \infty \ .$$

When $H > 1/2$ it can be shown that:

$$\sum_{k=0}^{\infty} E[B_1^H(B_{k+1}^H - B_k^H)] = \infty \ , \qquad (5.1.5)$$

so that, indeed, the increments of fBm exhibit long range dependence for $H > 1/2$. The same infinite series becomes finite when $H < 1/2$.

P-order Variation
The p-variation of an fBm B_H on $[0, \ell]$ is specified as follows[3]:

$$\text{p-th variation} := \lim_{n\to\infty} \sum_{j=0}^{2^n-1} \left| B^H_{\frac{j+1}{2^n}\ell} - B^H_{\frac{j}{2^n}\ell} \right|^p = \begin{cases} 0, & pH > 1 \ , \\ \infty, & pH < 1 \ , \\ \ell, & pH = 1 \ . \end{cases} \qquad (5.1.6)$$

Notice that in the case where $p = 2$ and $H = 1/2$, by using (5.1.6) we can retrieve the quadratic-variation of the standard Brownian motion, that is, we have the well-known result (for B denoting standard Brownian motion):

$$\lim_{n\to\infty} \sum_{j=0}^{2^n-1} \left| B_{\frac{j+1}{2^n}\ell} - B_{\frac{j}{2^n}\ell} \right|^2 = \ell \ .$$

The memory properties of fBm have been found to be desirable for many models in a variety of scientific fields, with financial mathematics being an important area of application. There have been attempts to use fBm for modelling the underlying

[3] A formal proof and discussion of these results can be found in [28, 27]

price of an asset, for instance by generalising the geometric Brownian motion as follows:

$$dS_t = \mu S_t dt + \sigma S_t dB_t^H \ ,$$

for the relevant parameters μ, σ. Such an approach raises some concerns as the introduction of memory violates the no-arbitrage rule (see for instance [17, 76]) due to the noise process not being a martingale anymore. Some more popular models which respect the no-arbitrage rule have been constructed by specifying, instead, the underlying volatility process via an fBm, so that for a price process we have:

$$dS_t = \mu S_t dt + \sigma(Y_t) S_t dB_t,$$

for some function σ, with Y_t being now a stochastic process itself driven by fBm (see e.g. [20, 21]). Besides financial mathematics, fBm has been used in a wide range of applications. Another important area of application is Biophysics where fBm-driven processes are used to model sub-diffusions within proteins (see [54]), and in telecommunications it has been used to model ethernet traffic [55, 84].

5.1.1 Davies-Harte method for simulating fBm

We use the exact Davies-Harte method to simulate fBm sample paths, originally introduced in [26]. First though, we will briefly explain the standard Cholesky decomposition approach to further motivate the computational advantages of the Davies-Harte method that we will present later on. Given an N-dimensional Gaussian vector with zero mean and covariance matrix Σ, the Cholesky method involves decomposing Σ as:

$$\Sigma = \Gamma \Gamma^\top, \tag{5.1.7}$$

where Γ is the lower triangular matrix in the decomposition. Since Σ is the covariance matrix, then, loosely speaking, Γ can be thought of as the 'standard-deviation' matrix. It is now trivial, that given a sample $u \sim N(0, I)$ in N-dimensions, we get:

$$v = \Gamma u,$$

which provides a sample from $N(0, \Sigma)$.

It is easy to see how this result is useful in the context of simulating fBm. Constructing the relevant covariance matrix based on the fBm covariance function in (5.1.1) so that:

$$\Sigma = \left(E[B_{t_i}^H B_{t_j}^H] \right)_{i,j=1}^N \ , \tag{5.1.8}$$

for some ordered time instances $0 < t_1 < t_2 < \cdots < t_N$, we can then derive the Cholesky decomposition of Σ and set $v = \Gamma u$ as above to construct an fBm sample path at these N time points. One concern with this approach is that the computational cost of performing the Cholesky decomposition is typically $\mathcal{O}(N^3)$.

The Davies-Harte method (detailed e.g. in [85]) follows a different approach compared to the Cholesky decomposition method, and by exploiting a fast Fourier transform (FFT) it achieves the sampling task with $\mathcal{O}(N \log N)$ calculations. Notice though that the method is relevant only for regular grids, so that the relevant time instances t_i need to be equidistant.

We describe the Davies-Harte method here since the details in the development of the method are relevant for the MCMC algorithms we will construct later on in this Chapter. Consider an fBm path B_H defined on a regular grid of N points with step size $\delta = \ell/N$. Then, we consider the increments for B_H on this grid, giving rise to the vector G_N of increments defined in (5.1.2), with autocovariance function $\gamma_\delta(j)$ defined in (5.1.4).

The Davies-Harte method simulates G_N by using a $2N$-sized vector of i.i.d $N(0,1)$ samples. It exploits the fact that the increments' vector G_N is stationary, thus, its covariance matrix is Toeplitz. Indeed, the covariance matrix of G_N for $\delta = 1$ (that is $\ell = N$; we will use the notation $\gamma(j)$ as a shorthand for $\gamma_\delta(j)$ when $\delta = 1$) is as follows:

$$\Sigma = \begin{pmatrix} \gamma(0) & \gamma(1) & \dots & \gamma(N-2) & \gamma(N-1) \\ \gamma(1) & \gamma(0) & \dots & \gamma(N-1) & \gamma(N-2) \\ \vdots & \vdots & & \vdots & \vdots \\ \gamma(N-1) & \gamma(N-2) & \dots & \gamma(1) & \gamma(0) \end{pmatrix}.$$

We also define the matrix:

$$\Sigma^f = \begin{pmatrix} 0 & \gamma(N-1) & \dots & \gamma(2) & \gamma(1) \\ \gamma(N-1) & 0 & \dots & \gamma(3) & \gamma(2) \\ \vdots & \vdots & & \vdots & \vdots \\ \gamma(1) & \gamma(2) & \dots & \gamma(N-1) & 0 \end{pmatrix}.$$

These covariance matrices are then embedded within the following *circular* matrix:

$$C = \begin{pmatrix} \Sigma & \Sigma^f \\ \Sigma^f & \Sigma \end{pmatrix}.$$

The matrix C is circular in the sense that the last element in a row in C becomes the first element in the next row. This structure allows for a simple eigen-expansion for C. In particular, we can write the decomposition:

$$C = P\Lambda_H P^*$$

where P is a $(2N) \times (2N)$-matrix with the following elements:

$$P_{jk} = \tfrac{1}{\sqrt{2N}} \exp(-2\pi i \tfrac{jk}{2N}), \quad 0 \leq j, k \leq 2N - 1 \ ,$$

for the complex i such that $i^2 = -1$. Then P^* is the complex transpose of P. We also define the diagonal matrix:

$$\Lambda_H = \mathtt{diag}\{\lambda_0, \lambda_1, \ldots, \lambda_{2N-1}\} \ , \tag{5.1.9}$$

with the following eigenvalues:

$$\lambda_k = \sum_{j=0}^{2N-1} c_{0,j} \exp(-2\pi i \tfrac{jk}{2N}) \ , \tag{5.1.10}$$

where $c_{0,j}$, for $0 \leq j \leq 2N - 1$, are the components of the first row of the circular matrix C. Notice that using FFT, the components of λ_H can be calculated with $\mathcal{O}(N \log N)$ calculations. We can easily obtain:

$$\sqrt{C} = P\sqrt{\Lambda_H}P^* \ . \tag{5.1.11}$$

Note 5.1.1. *Here it is necessary that $\lambda_k \geq 0$. It is shown in the literature that this is, indeed, the case for a general sequence of covariances $\gamma(j)$ if either one of the following conditions is met:*

1. *the auto-covariance sequence $\gamma(j)$ is non-negative, decreasing and a convex function of $j \geq 1$ (see [33, 43]); or:*

2. *we are in a stationary context and we have that $\gamma(j) < 0$ for $k > 0$ (see [24]).*

In Section 4.2 of [24] it is shown that fBm satisfies the second condition when $0 < H < 1/2$. For the case where $1/2 \leq H < 1$ the first condition is shown to hold in [43, 19].

We can now summarise the Davies-Harte sampling algorithm as follows: First, we simulate $Z_0 \sim N(0, I_{2N})$ and then we calculate $\sqrt{C} = P\sqrt{\Lambda_H}P^*$. Calculating:

$$\sqrt{C}Z_0 = P\sqrt{\Lambda_H}P^* Z_0 \tag{5.1.12}$$

and retrieving the first N values provide precisely the required fBm sample on the regular grid with $\delta = 1$. Proposition 3 of [85] proposes a small variation of the above approach that replaces the $\mathcal{O}(N \log N)$ computation $P^* Z_0$ with an alternative which costs $\mathcal{O}(N)$. The method involves simulating directly the distribution $W = P^* Z_0$ as follows:

1. Sample independently $W_0, W_N \sim N(0, 1)$

2. Sample independently $V, V' \sim N(0, I_{N-1})$

3. Let $W_j = \frac{1}{\sqrt{2}}(V_j + iV_j')$ and $W_{2N-j} = \frac{1}{\sqrt{2}}(V_j - iV_j')$, for $1 \leq j \leq N - 1$.

We can observe that this is equivalent to calculating the following:

$$P\sqrt{\Lambda_H}MZ \ ,$$

for a vector $Z \sim N(0, I_{2N})$ where M is a matrix:

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}$$

for the following sub-matrices:

$M_{11} = \mathtt{diag}\{1, 1/\sqrt{2}, 1/\sqrt{2}, \ldots, 1/\sqrt{2}\}$;
$M_{12} = \{m_{ij}\}$, with $m_{i,i-1} = 1/\sqrt{2}$ for $1 \leq i \leq N-1$ and $m_{i,j} = 0$ otherwise;
$M_{21} = \{m_{ij}\}$, with $m_{i,N-i} = 1/\sqrt{2}$ for $1 \leq i \leq N-1$ and $m_{i,j} = 0$ otherwise;
$M_{22} = \mathtt{diag}_{inv}\{1, -i/\sqrt{2}, -i/\sqrt{2}, \ldots, -i/\sqrt{2}\}$.

Finally, due to the self-similarity property of fBm, we can generate fGn corresponding to an fBm sampler path on a regular grid of arbitrary step-size $\delta > 0$ by setting:

$$G_N = \mathcal{P}_{1:N}\left\{\, \delta^H\, P\sqrt{\Lambda_H}MZ \,\right\}, \tag{5.1.13}$$

where $\mathcal{P}_{1:N}$ denotes a projection onto the first N co-ordinates. This concludes our description of the Davies-Harte algorithm.

5.1.2 Alternative Methods for Simulating fBm

In this Chapter we use the exact Davies-Harte method to simulate fBm– which is just one of many different methods for generating fBm sample paths. In fact, there are many available algorithms, both exact and approximate, summarised for instance in [32]. Here, we provide a brief discussion on one alternative method that is popular in recent literature and we explain the reasons why we have chosen the Davies-Harte method.

An effective method for generating fBm sample paths is the so-called *conditionalized random midpoint displacement* or $RMD_{l,r}$ (for some algorithmic parameters $l, r \geq 1$), originally developed in [63]. Unlike Davies-Harte, $RMD_{l,r}$ is an approximate method for simulating fBm and it involves some difficult to quantify bias. The main properties of the method are as follows:

- **Computational Costs:** $RMD_{l,r}$ has costs of $\mathcal{O}(N)$, which makes it more effective than Davies-Harte which has costs of $\mathcal{O}(N \log N)$.

- **On-the-Fly Generation:** $RMD_{l,r}$ can operate 'on-the-fly', i.e. it can be used to generate fBm traces without a-priori knowledge of the length of the path [63].

- **Bias:** one can try to tune algorithmic parameters l and r to obtain an algorithm with small bias[4] and make it comparable to the Davies-Harte method. In the sequel, since we have already been using the Euler method to discretize diffusion processes driven by fBm, it could be argued that the small amount of bias intrinsic in $RMD_{l,r}$ should have a minimal effect on the properties of the overall algorithm compared to using an exact algorithm.

Despite the above described characteristics of $RMD_{l,r}$ we still chose to use the Davies-Harte method for the MCMC algorithms, we will expand on this later in the Chapter. The main reason on selecting the Davies-Harte algorithm boils down to a simple convenient expression in the form of a linear mapping between a vector of i.i.d. normal values and the required fBm path. This allows us to easily transform between the two, which we later use to break the dependence between variables and calculate derivatives for gradient-based MCMC methods. In contrast, $RMD_{l,r}$ is constructed by means of bisections and interpolations [63], which do not allow for a simple clean expression.

[4] In empirical studies e.g. [32] $RMD_{3,3}$ has been found to have relatively small error

5.1.3 Numerical Approximation of fBm-Driven Diffusion

The Davies-Harte method provides a fast way for simulating sample paths of B^H, thus, allowing for the generation of paths for general non-linear diffusions of the form:

$$dX_t = b(X_t)dt + \sigma(X_t)dB_t^H \tag{5.1.14}$$

(for some relevant drift and diffusion coefficient functions b and σ respectively) using Euler's method, that is, by setting:

$$X_{i\delta} \approx X_{(i-1)\delta} + b(X_{(i-1)\delta})\delta + \sigma(X_{(i-1)\delta})(B_{i\delta}^H - B_{(i-1)\delta}^H) \ .$$

Such a simple discretization scheme has been shown to have a diminishing bias as $\delta \to 0$ in the case where $H \geq \frac{1}{2}$, but caution is needed when $H < 1/2$, as the Euler approximation of stochastic integrals with respect to fBm can explode in that case as $\delta \to 0$. This has to do with the fact that when $H < 1/2$ the 'roughness' of fBm is more intense than when $H = 1/2$ (and certainly more than when $H > 1/2$). Indeed, recall from (5.1.6) that when $H < 1/2$ the quadratic variation of fBm explodes[5]. Here, we look at a particular stochastic integral and illustrate the issues that can arise when $H < 1/2$. We apply the Euler scheme to discretize the stochastic integral $\int B_t^H dB_t^H$ as follows:

$$\int_0^\ell B_t^H dB_t^H \approx \sum_{i=1}^N B_{(i-1)\delta}^H (B_{i\delta}^H - B_{(i-1)\delta}^H) \ , \tag{5.1.15}$$

for $\delta = \ell/N$, and some large $N \geq 1$. Then, through some simple algebraic manipulations we can get that:

$$\begin{aligned}
&\sum_{i=1}^N B_{(i-1)\delta}^H (B_{i\delta}^H - B_{(i-1)\delta}^H) = \\
&\qquad = \sum_{i=1}^N \left\{ \tfrac{(B_{i\delta}^H)^2}{2} - \tfrac{(B_{(i-1)\delta}^H)^2}{2} \right\} - \tfrac{1}{2} \sum_{i=1}^N (B_{i\delta}^H - B_{(i-1)\delta}^H)^2 \ .
\end{aligned} \tag{5.1.16}$$

The first quantity on the RHS of (5.1.16) only concerns the first and last values of B^H as the rest gets canceled out, whereas the second quantity is the quadratic variation of B^H (as $N \to \infty$). We know from (5.1.6) that the quadratic variation of fBm goes to infinity when $H < 1/2$, thus, the Euler scheme in (5.1.16) will also diverge to infinity. Notice that the definition itself of a stochastic integral when

[5] For some general investigation on this matter we refer the reader to [56]

$H < 1/2$ can be done via different approaches, see e.g. [9].

Thus, to discretize SDEs driven by fBm when $H < 1/2$ in the context of a non-constant diffusion coefficient we need a mechanism that will overcome the above issue. The approach suggested in [56] involves using the Doss-Sussmann representation (introduced in [79], see also [64]), to define the solution of an SDE of the type in (5.1.14). Under the Doss-Sussmann interpretation, standard calculus rules apply [6], thus, we can remove the diffusion coefficient with a simple transformation, and then apply the Euler scheme in a process with unit diffusion coefficient.

Here we only include a quick overview of the approach sketched above[7]. Consider the SDE (5.1.14) written now in an integral form:

$$X_t = X_0 + \int_0^t b(X_s)ds + \int_0^t \sigma(X_s)dB_s^H \ . \tag{5.1.17}$$

The Doss-Sussmann approach makes sense of a solution for (5.1.17) in a pathwise manner. That is, for any fixed fBm path $B_s(\omega)$, we obtain solutions for (5.1.17) for arbitrary smooth paths (i.e. continuously differentiable paths) in a small neighborhood of $B_s(\omega)$, and then define the solution of (5.1.17) as the limit of these solutions as the neighborhood gets tighter. The work in [79] shows all the theoretical details for making this approach rigorous. A consequence of making sense of a solution in this way is that standard calculus rules will apply when considering transformations of X_t. Another relevant detail, following [79], is that the solution of the SDE in (5.1.17) is determined as:

$$X_t = \phi(B_t^H, Z_t) \ , \tag{5.1.18}$$

where the function $\phi(x,y) : \mathbb{R}^2 \mapsto \mathbb{R}$ satisfies $\frac{\partial}{\partial x}\phi(x,y) = \sigma(\phi(x,y))$, $\phi(0,y) = y$ for all $y \in \mathbb{R}$, and the process Z_t solves the random ordinary differential equation:

$$Z_t = X_0 + \int_0^t a(B_s^H, Z_s)ds \ , \tag{5.1.19}$$

where we have set:

$$a(x,y) = b(\phi(x,y)) \exp\big\{ -\int_0^x \sigma'(\phi(u,y))du\big\} \ . \tag{5.1.20}$$

Then, it can be shown that under the conditions that σ is continuously differen-

[6] For $H = 1/2$ the Doss-Sussmann approach coincides with the Stratonovich interpretation of the solution of an SDE, for which it is known that standard calculus rules also apply

[7] For a more detailed explanation the reader is referred to [79, 56]

tiable and both b and σ' are locally Lipschitsz, the specification of the solution Y_t in (5.1.18) is unique.

The Doss-Sussmann interpretation is relevant for any $H \in (0,1)$ (again see [56]), and as mentioned earlier coincides with the Stratonovich interpretation of standard SDEs with $H = 1/2$, see [79]. Critically, it allows for standard calculus rules, thus, considering a sufficiently smooth mapping h and the process $h(X_t)$, we get the following change of variables rule:

$$h(X_t) = h(X_0) + \int_0^t h'(X_s)b(X_s)ds + \int_0^t h'(Y_s)\sigma(Y_s)dB_s^H \ . \tag{5.1.21}$$

This is critical, since setting $h(y) = \int^y (1/\sigma)(u)du$ will allow for transforming the original SDE to one that has a constant diffusion coefficient.

Hence, we can now describe the numerical scheme for the approximation of the SDE in (5.1.14) on the regular grid $\{i\delta\}$, for $i = 1, 2, \ldots N$, and $\delta = \ell/N$. We also allow for the possibility that the drift function and diffusion coefficients depend on some parameter θ (thus, we assume that $b(x) = b(x,\theta)$ and $\sigma(x) = \sigma(x,\theta)$), as this will be needed later on when developing our Bayesian inference method. Thus, we have:

(i) Consider the process $F_t = \int_{x_0}^{X_t} \sigma^{-1}(u,\theta)du =: F(X_t, \theta, x_0)$. It can be easily shown, using standard calculus, that F_t solves the SDE:

$$dF_t = b_F(F_t, \theta, x_0)dt + dB_t^H \ , \quad F_0 = 0 \ ,$$

where $b_F(\cdot, \theta, x_0) = (b/\sigma)\big(F^{-1}(\cdot, \theta, x_0), \theta\big)$.

(ii) Apply now the standard Euler scheme, for $G(i) = B_{i\delta}^H - B_{(i-1)\delta}^H$:

$$F_i - F_{i-1} = b_F(F_{i-1}, \theta, x_0)\,\delta + G(i) \ , \quad F_0 = 0 \ . \tag{5.1.22}$$

(iii) Return $X_i = F^{-1}(F_i, \theta, x_0)$, for $1 \leq i \leq N$, with $X_0 = x_0$.

We want to briefly mention that the Doss-Sussmann interpretation can be used also for multi-dimensional SDEs, but only for scalar fBm-noise. Also, there are many other interpretations for solving SDEs when $H < 1/2$, with corresponding numerical schemes which must be higher-order (compared to Euler) for the approximations to converge, see for instance [62, 31, 59], but the technicalities involved are beyond the scope of this thesis.

5.2 Statistical Inference with fBm

We now develop an advanced MCMC algorithm for performing parameter inference for diffusion models driven by fBm. As in Section 3.3, we adopt a data augmentation framework[8] wherebdeyay the driving fBm is treated as a latent variable. Given the parameters and the latent fBm, we can calculate the likelihood of the data. The algorithm we describe here will be of the advanced HMC type and will have a cost of $\mathcal{O}(N \log N)$ per step induced by the complexity of the Davies-Harte method for simulating the driving fBm.

More specifically, we aim to perform some Bayesian inference about the parameters of the following SDE:

$$dX_t = b(X_t, \theta)dt + \sigma(X_t, \theta)dB_t^H, \tag{5.2.1}$$

given that we observe the process X_t with error:

$$Y_{t_i} = X_{t_i} + N(0, \xi^2) \ , \tag{5.2.2}$$

for some ordered discrete time instances $0 < t_1 < t_2 < \cdots < t_n = \ell$ (by convention $t_0 = 0$). Thus, the unknown parameter vector here is (θ, H, ξ).

Note 5.2.1. *We use the model in (5.2.1) and (5.2.2) for illustration purposes. But we should mention that the method described here is relevant for more general data regimes than the one in (5.2.2). In particular, any context where we can have an explicit expression of the likelihood of data Y given the underlying process X, can in principle, be treated with our method.*

Notice, that for the model structure in (5.2.1) and (5.2.2), we can sketch the dependencies among the involved variables via the hierarchical graph in Figure 5.2. In a data augmentation setup, an MCMC algorithm will try to sample from the posterior distribution of the parameter vector θ and the latent diffusion path given the data. A first issue that we need to tackle here is that B^H and H are highly correlated. In fact, given the complete continuous path of B^H we can uniquely identify H, so that the distribution of H given B^H is a Dirac measure. This is apparent for instance from the p-th variation results in (5.1.6). Indeed, using the p-variations of a given path from B^H we could easily construct a mechanism for identifying H.

We could simply adjust p by increasing it when the p-variation of the given

[8] Data augmentation was originally described in [80] but expanded for our context in [71]

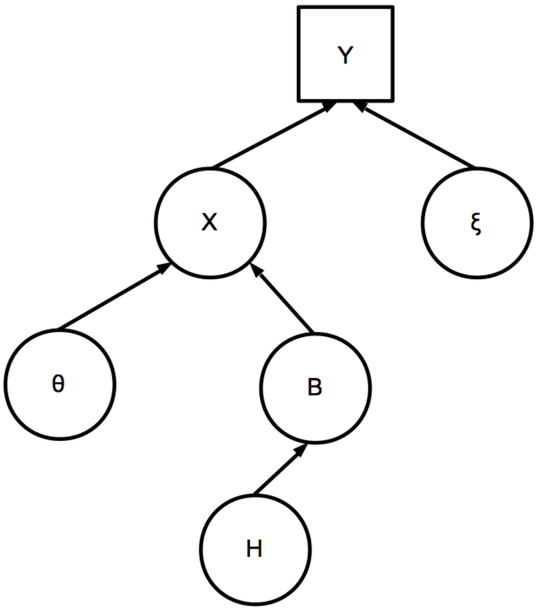

Figure 5.2: Dependency Graph of default model

B^H sample path is equal to zero and decreasing it when the p-variation diverges towards infinity, until finding the value of p for which the variation would be non-zero finite– thus, also identifying the value of the Hurst parameter H as $1/p$. In practice, a discretization of B^H will be used within the algorithm, so that the dependence with H will not be as extreme as above, but it will still be high. It is well documented that high correlations between latent variables and parameters can lead to very inefficient Gibbs samplers or other MCMC algorithms[9].

Thus, it is important to disentangle B^H and H and we can achieve this by using the Davies-Harte method of Section 5.1.1. Indeed, in Section 5.1.1 we described the Davies-Harte method that generates fBm sample paths using FFT. The method boils down to the $1-1$ linear mapping $Z \mapsto G_N = \mathcal{P}_{1:N}\{\,\delta^H\,P\sqrt{\Lambda_H}MZ\,\}$ in (5.1.13) that transforms a $2N$-dimensional vector $Z \sim N(0, I_{2N})$ into an fGN at N discrete time instances of step-size δ. That is, we have:

$$B^H = F(Z, H), \tag{5.2.3}$$

where F is a linear transform. So, we will use the $2N$-vector Z as a latent variable in our method instead of B^H– as Z and H are now a-priori independent (see the new hierarchical model structure in Figure 5.3).

Under this new model interpretation, we can write the posterior distribution

[9] See e.g. [75, 15, 16, 74, 61]

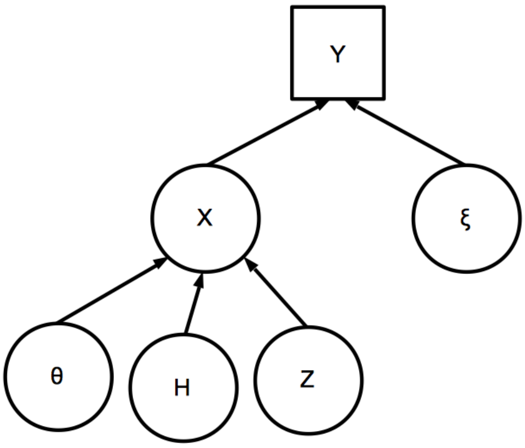

Figure 5.3: Dependency Graph 2, using transformation to de-couple dependencies

of interest as follows:

$$p(H, Z, \theta, \xi | Y) \propto p(Y | H, Z, \theta, \xi) p(H) p(Z) p(\theta) p(\xi) \ , \tag{5.2.4}$$

where, from (5.2.2) we have that:

$$p(Y | H, Z, \theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\xi^2}} e^{\frac{-(Y_{t_i} - X_{t_i})^2}{2\xi^2}} \ . \tag{5.2.5}$$

Having the full conditional distribution allows us to define an MCMC algorithm. In this case we will be using advanced HMC methods to do update the parameters H,θ and latent variables Z.

Note 5.2.2. *For the remainder of this Chapter we will assume that the error variance parameter ξ is fixed and known, so it doesn't have to be inferred from the data. Our presentation for the development of the method agrees with the numerical examples we show later in this Chapter where, indeed, we fixed ξ, as otherwise the MCMC trajectory got trapped and did not converge. We have left the issue of inferring ξ for future work.*

5.2.1 A HMC-within-Gibbs Sampler

We first present a Metropolis-within-Gibbs sampler where the updating of H, θ, Z is in an order established from their corresponding full conditional distributions. We apply the following approach: For the low-dimensional parameters H and θ we use a standard HMC as described in Table 1.1 in Section 1.1.3, whereas for the high-dimensional parameter Z, we exploit the fact that its target distribution is a change of measure from a Gaussian law[10] and apply the advanced HMC algorithm as described in Table 2.1 in Section 2.2. Notice that in the present context, the covariance matrix for the reference Gaussian law in the change of measure is simply the identity matrix I_{2N}. As we may remember, one important effect of the advanced HMC sampler is that its mixing time is mesh-free, that is: it will not get worse with increasing N.

Step 1	Update H from $p(H\|\theta, Z, Y) \propto p(Y\|H, Z, \theta)p(H)$ using standard HMC.
Step 2	Update θ from $p(\theta\|H, Z, Y) \propto p(Y\|H, Z, \theta)p(\theta)$ using standard HMC.
Step 3	Update Z from $p(Z\|H, \theta, Y) \propto p(Y\|H, Z, \theta)p(Z)$ using advanced HMC.

Table 5.1: A Metropolis-within-Gibbs sampler for the fBm-driven diffusion model in (5.2.1)-(5.2.2) using HMC updates.

We summarized the algorithm in Table 5.1. As said before, the implementation of HMC requires calculating the gradient of the logarithm of the target distribution. Assuming that the gradients of the log-priors $\log p(\theta)$ and $\log p(H)$ are easy to calculate, the only challenging term is $\log p(Y|H, Z, \theta)$. From the analytical expression in (5.2.20) we have that:

$$\log p(Y|H, Z, \theta) = -\sum_{i=1}^{n} \frac{(Y_{t_i} - X_{t_i}(Z, H, \theta))^2}{2\xi^2} \ , \tag{5.2.6}$$

where we have written $X_{t_i} = X_{t_i}(Z, H, \theta)$ to emphasize the dependence of the diffusion path X_{t_i} on all the involved variables (Z, H, θ). To simplify the expressions, we set:

$$\begin{aligned} \Phi(Z, H, \theta) &= -\log p(Y|H, Z, \theta) - \log p(H, \theta) \\ &= \frac{\sum_{i=1}^{N} (Y_i - X_{t_i}(Z, H, \theta))^2}{2\xi^2} - \log p(H, \theta) \ . \end{aligned} \tag{5.2.7}$$

So, the derivatives we would have to calculate are:

$$\nabla_Z \Phi(Z, H, \theta) \ , \quad \partial_H \Phi(Z, H, \theta) \ , \quad \nabla_\theta \Phi(Z, H, \theta) \ . \tag{5.2.8}$$

We will show the details for the calculation of the derivatives in (5.2.8), but before that, we present another algorithm which updates all parameters (Z, H, θ) jointly

[10] A-priori, the distribution of Z is simply the product of $2N$ standard Gaussians

within an HMC scheme, as this more effective sampler requires the calculation of the same quantities.

5.2.2 A Joint-Update HMC Sampler

The Gibbs sampler scheme sketched in Table 5.1 is not the only option available. Indeed, we may also choose to update jointly (Z, H, θ) within an HMC sampler. To simplify the expressions that follow we first set:

$$\theta' = (H, \theta) \ .$$

The motivation here is that, even if a-priori Z, H, θ are independent[11], a-posteriori strong correlations could arise as the data impose a lot of restrictions on the permitted joint values of Z, H, θ for the diffusions instances $X_{t_i}(Z, H, \theta)$ in order to get close to the data points Y_{t_i}. In the presence of such strong correlations a joint update scheme could be more effective than the Gibbs sampler described in Table 5.1, and indeed, this will be proven to be the case in the numerical examples later on in this Chapter.

But first some algorithmic development is needed as the advanced HMC sampler that jointly updates Z and θ' cannot be derived directly from the advanced HMC algorithm described in Section 2.2. This is because in that Section we described only the update for the random element defined as a change of measure from a Gaussian law (i.e. Z in the current context) with all other elements presumed fixed[12].

Compared to the advanced HMC described earlier in Section 2.2, we now have the extended locations and velocities:

$$v = (v_z, v_{\theta'}) \in \mathbb{R}^{2N+q}$$

and:

$$x = (z, \theta') \in \mathbb{R}^{2N+q}$$

where $2N$ is the dimension of the Z vector and q the dimension of θ'. Same as before we define the total energy function as:

$$H(x, v; M) = \Phi(x) + \tfrac{1}{2}\langle z, z\rangle + \tfrac{1}{2}\langle v, Mv\rangle, \tag{5.2.9}$$

[11] For Z, H this was induced after using the Davies-Harte transform

[12] The approach we follow has also been presented in a parallel work in [35]

for a user specified positive-definite mass matrix M and the term $\Phi(x) = \Phi(z, H, \theta)$ as defined in (5.2.7). As before, $\frac{1}{2}\langle v, Mv\rangle$ can be interpreted as the kinetic energy and $\Phi(x) + \frac{1}{2}\langle z, z\rangle$ as the potential energy. We can now define the relevant distribution on the joint (x, v)-space as follows:

$$Q(x, v; M) = \exp\big\{ - H(x, v; M)\big\} \ . \tag{5.2.10}$$

The relevant Hamiltonian equations on $\mathbb{R}^{2N+q}$ (as written for instance in (1.1.14)) are now expressed as follows:

$$\frac{dx}{dt} = v \ , \quad M\frac{dv}{dt} = -(z, 0)^\top - \nabla\Phi(x) \ . \tag{5.2.11}$$

A point of discussion is the choice of the user-defined matrix M. As per our discussion for the advanced HMC sampler in Section 2.2: for the portion of M that corresponds to the high-dimensional z-part of our space, the requirement to construct an algorithm with mesh-free mixing time as $N \to \infty$ leads to selecting the inverse covariance of the Gaussian prior for z, i.e. simply the identity matrix I_{2N}. This motivates the following specification for the complete mass matrix M:

$$M = \begin{pmatrix} I_{2N} & 0 \\ 0 & A \end{pmatrix} \ , \quad A = \mathtt{diag}\{a_i : 1 \le i \le q\} \ . \tag{5.2.12}$$

Under this choice of M we can re-write the Hamiltonian equations in (5.2.11) as follows:

$$\frac{dx}{dt} = v \ , \quad \frac{dv}{dt} = -(z, 0)^\top - M^{-1}\nabla\Phi(x) \ . \tag{5.2.13}$$

Note 5.2.3. *Recall that a good choice for M is one that resembles the inverse covariance of the target distribution. This intuition can guide the choice of the diagonal matrix A. Indeed, in the numerical implementations later on we choose the coefficients a_i to be close to the inverse of the corresponding posterior marginal variances, with the later estimated by preliminary runs of the algorithm.*

As with the derivation of advanced HMC in Section 2.2, we split (5.2.13) into a coupled system of equations:

$$\frac{dx}{dt} = 0 \ , \quad \frac{dv}{dt} = -M^{-1}\nabla\Phi(x) \ ; \tag{5.2.14}$$

$$\frac{dx}{dt} = v \ , \quad \frac{dv}{dt} = -(z, 0)^\top \ . \tag{5.2.15}$$

Both of these equations can be solved analytically, thus, we construct a numerical scheme for the original dynamics in (5.2.13) by synthesising steps from the solutions operators of (5.2.14) and (5.2.15). Indeed, the solvers for (5.2.14) and (5.2.15) are defined respectively as:

$$\begin{aligned}\Xi_t(x,v) &= \big(\,x,\, v - tM^{-1}\nabla\Phi(x)\,\big)\,;\\ \tilde{\Xi}_t(x,v) &= \big(\,(\cos(t)z + \sin(t)v_z,\, \theta' + tv_{\theta'}),\, (-\sin(t)z + \cos(t)v_z, v_{\theta'})\,\big)\ . \end{aligned} \tag{5.2.16}$$

The leapfrog integrator we develop for the original dynamics (5.2.13) is obtained by alternating these two operators as follows:

$$\Psi_h = \Xi_{h/2} \circ \tilde{\Xi}_h \circ \Xi_{h/2}\ , \tag{5.2.17}$$

for sufficiently small values of the step-size tuning parameter h. We can synthesize a number of $I = \lfloor T/h \rfloor$ applications of Ψ_h in (5.2.17) to traverse the Hamiltonian dynamics up to some time horizon $T > 0$. We denote by Ψ_h^I the complete synthesized operator. It is easy to verify that Ψ_h retains the volume preservation and symmetricity properties of the standard leapfrog operator[13], thus, the acceptance probability for the developed HMC method is the same as for standard HMC.

The complete advanced HMC sampler that updates jointly the variables z and θ' is summarised in Table 5.2.

5.2.3 Calculation of Derivatives

We have identified the derivatives needed for our advanced HMC samplers in (5.2.8). These can be found using the chain rule– with some caution so that computational costs remain $\mathcal{O}(N\log N)$. Recall that processes B^H and X are in practice considered on the regular grid $i\delta$ for $\delta = \ell/N$. Also, we take under careful consideration the details for the numerical scheme for X described in steps (i)-(iii) in Section 5.1.3. Thus, we have to keep in mind the composition $Z \mapsto G_N \mapsto (F_1, F_2, \ldots, F_N)$ defined therein. A direct application of the chain rule gives the following:

$$\begin{aligned}\nabla_Z \log p(Y|Z,H,\theta) &= \left(\frac{dG_N}{dZ}\right)^{\top}\left(\frac{dF}{dG_N}\right)^{\top}\nabla_F \log p(Y|F,\theta)\ ;\\ \nabla_\theta \log p(Y|Z,H,\theta,x_0) &= \left(\frac{dF}{d\theta}\right)^{\top}\nabla_F \log p(Y|F,\theta) + \nabla_\theta \log p(Y|F,\theta)\ ;\\ \partial_H \log p(Y|Z,H,\theta) &= \left(\frac{dG_N}{dH}\right)^{\top}\left(\frac{dF}{dG_N}\right)^{\top}\nabla_F \log p(Y|F,\theta)\ . \end{aligned} \tag{5.2.18}$$

[13] As specified in Theorem 1.1.2 in Section 1.1.3

Advanced Joint-Update HMC:

(i) *Start with an initial value* $x^{(0)} = (z^{(0)}, \theta'^{(0)})$ *with elements* $z^{(0)} \sim N(0, I_{2N})$ *and* $\theta'^{(0)} \sim p(\theta')$*, and set* $k = 0$*. Specify a mass-matrix* M *as in (5.2.12).*

(ii) *Given* $x^{(k)}$*, sample* $v^{(k)} \sim N(0, M^{-1})$ *and propose:*

$$x^* = \mathcal{P}_x \Psi_h^I(x^{(k)}, v^{(k)}) \ .$$

(iii) *Consider:*

$$a = 1 \wedge \exp\big\{ - \Delta H(x^{(k)}, v^{(k)})\big\}$$

for $\Delta H(x, v) = H(\Psi_h^I(x, v)) - H(x, v)$*.*

(iv) *Set* $x^{(k+1)} = x^*$ *with probability* a*; otherwise set* $x^{(k+1)} = x^{(k)}$*.*

(v) *Set* $k \to k + 1$ *and go to (ii).*

Table 5.2: Advanced joint updated HMC, with target distribution as specified in (5.2.4), with ξ assumed fixed and known.

where we have set:

$$\frac{dG_N}{dF} = (\partial G(i)/\partial F_j)_{ij} \in \mathbb{R}^{N\times N} \ ; \quad \frac{dG_N}{dZ} = (\partial G(i)/\partial Z_j)_{ij} \in \mathbb{R}^{N\times(2N)} \ ;$$
$$\frac{dF}{d\theta} = (\partial F_i/\partial \theta_j)_{ij} \in \mathbb{R}^{N\times p} \ ; \quad \frac{dG_N}{dH} = (dG(i)/dH)_i \in \mathbb{R}^{N} \ .$$

with p being the dimension of θ. We start from dG_N/dF. Recall the Euler approximation of F in (5.1.22). We now set:

$$f_i = -1 - b_F'(F_{i-1}, \theta)\, \delta \ , \quad i = 2, 3, \ldots, N,$$

and obtain immediately the following calculation:

$$\frac{dG_N}{dF} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ f_2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & f_3 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & f_N & 1 \end{pmatrix} .$$

Then, from the details of the Davies and Harte algorithm in Section 5.1.1, we have:

$$\frac{dG_N}{dZ} = \mathcal{P}_{1:N,1:(2N)}\{\,\delta^H P\sqrt{\Lambda_H}\,M\}\ ,$$

where $\mathcal{P}_{1:N,1:(2N)}$ denotes the projection of the $(2N)\times(2N)$-dimensional input matrix to its first N rows. The N rows of $dF/d\theta$ are obtained recursively via the Euler scheme (5.1.22) starting from:

$$\nabla_\theta F_1 = \nabla_\theta b_F(F_0,\theta)\,\delta\ ,$$

and for $i = 2,\dots,N$:

$$\nabla_\theta F_i = \nabla_\theta F_{i-1}\cdot\big(1 + b_F'(F_{i-1},\theta)\,\delta\big) + \nabla_\theta b_F(F_{i-1},\theta)\,\delta\ .$$

Making use again of the Davies and Harte method, we have that:

$$\frac{dG_N}{dH} = \delta^H\ \log(\delta)\,\mathcal{P}_{1:N}\{P\sqrt{\Lambda_H}\,M\,Z\} + \delta^H\mathcal{P}_{1:N}\big\{P\frac{d\Lambda_H^{1/2}}{dH}MZ\big\}\ .$$

Then, using the expression in (5.1.10) for the elements $\{\lambda_k\}_{k=0}^{2N-1}$ of the diagonal matrix Λ_H, we get:

$$\frac{d\lambda_k^{1/2}}{dH} = \tfrac{1}{2\lambda_k^{1/2}}\sum_{j=0}^{2N-1}\frac{dc_{0,j}}{dH}\exp\big(-2\pi i\,\tfrac{jk}{2N}\big)\ . \tag{5.2.19}$$

The remaining derivatives $dc_{0,j}/dH$ are easy to obtain via $d\gamma(k)/dH$ where $\gamma(\cdot)$ are the lagged autocovariances of fBm increments defined in (5.1.4) for $\delta = 1$. From there, we have that:

$$\tfrac{d\gamma(k)}{dH} = \begin{cases} 0\ ,\ \ k=0\ ; \quad \log(2)2^{2H}\ ,\ \ k=1\ ; \\[2ex] (k+1)^{2H}\log(k+1) + (k-1)^{2H}\log(k-1) - 2\log(k)k^{2H}\ ,\ \ k\geq 2\ . \end{cases}$$

Notice that the calculation of dG_N/dH requires $\mathcal{O}(N\log N)$ operations using FFT.

It remains to calculate $\nabla_F \log p(Y|F,\theta)$ and $\nabla_\theta \log p(Y|F,\theta)$. We recall here that we have:

$$p(Y|F,\theta) = \prod_{i=1}^{n}\frac{1}{\sqrt{2\pi\xi^2}}\,e^{\frac{-(Y_{t_i}-X_{t_i}(F_{t_i},\theta))^2}{2\xi^2}}\ . \tag{5.2.20}$$

For $1 \leq k \leq n$, and $j_k = \lfloor t_k/\delta \rfloor$ (and $j_0 \equiv 0$), we clearly have for $1 \leq j \leq N$:

$$\big\{\nabla_F \log p(Y|F,\theta)\big\}_i = \sum_{k=1}^{n} \tfrac{(Y_{t_k} - X_{t_k}(F_{t_k},\theta))}{\xi^2} \cdot \tfrac{\partial X_{t_k}}{\partial F_{t_k}} \cdot \mathrm{I}\left[\, j_k = i \,\right] \, . \qquad (5.2.21)$$

A very similar calculation is carried out for the derivative $\nabla_\theta \log p(Y|F,\theta)$ which we omit here for brevity.

5.3 Validity of Joint-Update Advanced HMC Sampler

As noted before, the advantage of an advanced HMC algorithm versus its standard counterpart is its mesh-free mixing time i.e. as N increases and h remains fixed, algorithmic convergence and mixing properties do not deteriorate (of course the computing cost will increase as $\mathcal{O}(N \log N)$). Indeed, this is exactly what we have proven in Section 2.2.1 for the version of advanced HMC presented there. We will now make a very similar proof for the case of the Joint-Update advanced HMC sampler presented in this Chapter, in Section 5.2.2. Same as with the proof in Section 2.2.1 we follow closely the derivations of [5] and [35].

In this context, we adopt a scenario where the variable z corresponds to an infinite-dimensional vector comprised a-priori of i.i.d. standard Gaussian random variables. That is, we now have $z \in \mathbb{R}^\infty$ and $\theta' \in \mathbb{R}^q$. The target distribution, say, $\Pi = \Pi(Z,\theta)$ corresponds to the posterior of Z,θ observations Y and assumed to be defined on the following space:

$$\mathcal{H} := \mathbb{R}^\infty \times \mathbb{R}^q \ ,$$

via the following change of measure:

$$\frac{d\Pi}{d\{\otimes_{i=1}^{\infty} N(0,1) \times Leb_q\}}(Z,\theta' \mid Y) \propto e^{-\Phi(Z,\theta')}, \qquad (5.3.1)$$

for the function $\Phi : \mathcal{H} \mapsto \mathbb{R}$ defined in (5.2.7). We also need the infinite-dimensional vector of partial derivatives $\nabla\Phi : \mathcal{H} \mapsto \mathcal{H}$. Then, the velocity component $v = (v_z, v_{\theta'})$ will also lie in the same space, $v \in \mathcal{H}$. The mass matrix M, which is specified in (5.2.12) for finite dimensions now has an infinite-dimensional identity matrix I_∞ at its upper-left block. We also consider the analogue of the bivariate target $Q(dx, dv)$ in (5.2.10) in infinite-dimensions corresponding, in x-direction, to the posterior of $x = (z, \theta')$ given the data Y. Notice that we also have that $\Xi_{h/2}, \tilde{\Xi}_h, \Psi_h : \mathcal{H} \times \mathcal{H} \mapsto \mathcal{H} \times \mathcal{H}$.

The main idea of the proof here (as also in the proof of Section 2.2.1) is that the leapfrog mapping of the advanced algorithm Ψ_h projects $(x_0, v_0) \sim Q(dx, dv)$ to a random variable (x_h, v_h) which has a distribution that is absolutely continuous with respect to $Q(dx, dv)$. This attribute implies the existence of a non-zero acceptance probability even in the current infinite-dimensional set-up corresponding to $N = \infty$. We can see this intuitively by looking at the specification of the individual maps $\Xi_{h/2}$, $\tilde{\Xi}_h$ which are synthesized to provide Ψ_h in (5.2.17). For operator $\tilde{\Xi}_h$, the above prescribed attribute of preserving the absolute continuity properties of $Q(dx, dv)$ is apparent, since $\tilde{\Xi}_h$ simply performs a rotation on the (z, v_z)-space which is invariant for $\prod_{i=1}^{\infty} N(0,1) \otimes \prod_{i=1}^{\infty} N(0,1)$, therefore preserving the absolute continuity properties of $Q(dx, dv)$. Then for step $\Xi_{h/2}$ in (5.2.17), the gradient $\nabla_z \Phi(z, \theta')$ needs to be in the Cameron-Martin space of $\prod_{i=1}^{\infty} N(0,1)$ for the translation $v \mapsto v - \frac{h}{2} M^{-1} \nabla \Phi(x)$ to preserve the absolute continuity properties of the v-marginal $Q(dv)$. This Cameron-Martin space is precisely the ℓ_2-space of squared summable infinite sequences (see e.g. Chapter 2 of [25]).

For the sake of completeness, I will include here the full proof for the well-posedness of the algorithm in infinite-dimensional as described also in the parallel work in [35]. We begin by defining the reference measure on the joint (x, v)-space as follows:

$$Q_0 = Q_0(dx, dv) = \left(\prod_{i=1}^{\infty} N(0,1) \otimes Leb_q \right) \otimes \left(\prod_{i=1}^{\infty} N(0,1) \otimes N_q(0, A^{-1}) \right) ,$$

so that the joint target distribution is expressed as:

$$Q(dx, dv) \propto \exp\{-\Phi(x)\} Q_0(dx, dv) \ .$$

Following closely the proof in Section 2.2.1, we consider the sequence of probability measures:

$$Q^{(i)} = Q \circ \Psi_h^{-i}, \quad 1 \leq i \leq I, \tag{5.3.2}$$

which corresponds precisely to the push-forward projection flow of the target measure $Q(dx, dv)$ under application of the leapfrog mappings. As in Proposition 2.2.1,we can obtain a recursive formula for the Radon-Nikodym derivatives $\frac{dQ^{(i)}}{dQ_0}$ for $i = 1, 2, \ldots, I$. Recall here that:

$$\mathcal{C} = M^{-1} = \begin{pmatrix} I_\infty & 0 \\ 0 & A \end{pmatrix} \ , \quad A = \mathtt{diag}\{a_i : 1 \leq i \leq q\} \ .$$

We also set:

$$g(x) := -\mathcal{C}^{1/2}\nabla\Phi(x), \quad x \in \mathcal{H} \ .$$

From the definition of Ψ_h we have the following recursion of probability measures:

$$Q^{(i)} = Q^{(i-1)} \circ \Xi_{h/2}^{-1} \circ \tilde{\Xi}_h^{-1} \circ \Xi_{h/2}^{-1}.$$

Given the assumption $\nabla_z\Phi(z,\theta) \in \ell_2$, we have that $Q_0 \circ \Xi_{h/2}^{-1}$ and Q_0 are absolutely continuos with respect to each other, with Radon-Nikodym derivative:

$$\frac{d\{Q_0 \circ \Xi_{h/2}^{-1}\}}{dQ_0}(x,v) = \exp\{\langle \tfrac{h}{2}g(x), \mathcal{C}^{-1/2}v\rangle - \tfrac{1}{2}|\tfrac{h}{2}g(x)|^2\} =: G(x,v). \qquad (5.3.3)$$

The stated assumption on $\nabla_z\Phi(x)$ ensures that the inner products appearing in the density will converge. The above result is simply an application of Proposition 1.4.1 in Section 1.4, which is a statement of Theorem 2.21 of [25]. Thus, we have:

$$\begin{aligned}\frac{dQ^{(i)}}{dQ_0}(x_i,v_i) &= \frac{dQ^{(i)}}{d\{Q_0 \circ \Xi_{h/2}^{-1}\}}\frac{d\{Q_0 \circ \Xi_{h/2}^{-1}\}}{dQ_0}(x_i,v_i) \\ &= \frac{dQ^{(i)}}{d\{Q_0 \circ \Xi_{h/2}^{-1}\}}(x_i,v_i) \times G(x_i,v_i) \ . \qquad (5.3.4)\end{aligned}$$

In the calculations that follow we make repeated use of the following standard property for the Radon–Nikodym derivative: if M_1, M_2 are probability measures on the measurable space $(E,\mathcal{E})$, and F is a measurable mapping $F : E \mapsto E'$ for some second measurable space $(E', E)'$, then we have that:

$$\frac{d\{M_1 \circ F^{-1}\}}{d\{M_1 \circ F^{-1}\}}(x) = \frac{dM_1}{dM_2}(F^{-1}(x)) \ .$$

Also, we notice here that $Q_0 \circ \tilde{\Xi}_h \equiv Q_0$ as the mapping $\tilde{\Xi}_h$ will rotate the infinite-dimensional products of independent standard Gaussians for the z, v_z components of Q_0, thus, will preserve their distribution. For the θ-component, $\tilde{\Xi}_h$ is just a linear mapping (previous value plus a constant), thus, it will translate the Lebesque measure and will also preserve it. With the above results in mind, we work as follows:

$$\begin{aligned}\frac{dQ^{(i)}}{d\{Q_0\circ\Xi^{-1}_{h/2}\}}(x_i,v_i) &= \frac{d\{Q^{(i)}\circ\Xi_{h/2}\}}{dQ_0}(\Xi^{-1}_{h/2}(x_i,v_i))\\ &= \frac{d\{Q^{(i)}\circ\Xi_{h/2}\circ\tilde{\Xi}_h\}}{dQ_0}(\tilde{\Xi}^{-1}_h(\Xi^{-1}_{h/2}(x_i,v_i)))\\ &= \frac{d\{Q^{(i)}\circ\Xi_{h/2}\circ\tilde{\Xi}_h\circ\Xi_{h/2}\}}{d\{Q_0\circ\Xi_{h/2}\}}(\Xi^{-1}_{h/2}(\tilde{\Xi}^{-1}_h(\Xi^{-1}_{h/2}(x_i,v_i))))\\ &= \frac{dQ^{(i-1)}}{d\{Q_0\circ\Xi_{h/2}\}}(x_{i-1},v_{i-1})\ .\end{aligned}$$

Using now the chain rule and, again, (5.3.3) we get that:

$$\begin{aligned}\frac{dQ^{(i-1)}}{d\{Q_0\circ\Xi_{h/2}\}}(x_{i-1},v_{i-1}) &= \frac{dQ^{(i-1)}}{dQ_0}(x_{i-1},v_{i-1})\,\frac{dQ_0}{d\{Q_0\circ\Xi_{h/2}\}}(x_{i-1},v_{i-1})\\ &= \frac{dQ^{(i-1)}}{dQ_0}(x_{i-1},v_{i-1})\,\frac{dQ_0\circ\Xi^{-1}_{h/2}}{Q_0}(\Xi_{h/2}(x_{i-1},v_{i-1}))\\ &\equiv \frac{dQ^{(i-1)}}{dQ_0}(x_{i-1},v_{i-1})\cdot G(x_{i-1},v_{i-1}+\tfrac{h}{2}\,\mathcal{C}^{1/2}g(x_{i-1}))\ .\end{aligned}\tag{5.3.5}$$

Thus, bringing together (5.3.4) and (5.3.5), overall we have shown that:

$$\frac{dQ^{(i)}}{dQ_0}(x_i,v_i) = \frac{dQ^{(i-1)}}{dQ_0}(x_{i-1},v_{i-1})\cdot G(x_i,v_i)\cdot G(x_{i-1},v_{i-1}+\tfrac{h}{2}\,\mathcal{C}^{1/2}g(x_{i-1}))\ .$$

Applying the above recursion repeatedly will give that:

$$\frac{dQ^{(I)}}{dQ_0}(x_I,v_I) = \frac{dQ}{dQ_0}(x_0,v_0)\times\prod_{i=1}^{I}G(x_i,v_i)G(x_{i-1},v_{i-1}+\tfrac{h}{2}\mathcal{C}^{1/2}g(x_{i-1}))\ ,\tag{5.3.6}$$

now, using the fact that:

$$\Psi_h = \Xi_{h/2}\circ\tilde{\Xi}_h\circ\Xi_{h/2},\tag{5.3.7}$$

and some long but otherwise straightforward algebraic calculations, we find that:

$$\begin{aligned}&\log\{G(x_i,v_i)\cdot G(x_{i-1},v_{i-1}+\tfrac{h}{2}\mathcal{C}^{1/2}g(x_{i-1}))\} =\\ &\qquad= \tfrac{1}{2}\langle x_i,Lx_i\rangle+\tfrac{1}{2}\langle v_i,Lv_i\rangle-\tfrac{1}{2}\langle x_{i-1},Lx_{i-1}\rangle-\tfrac{1}{2}\langle v_{i-1},Lv_{i-1}\rangle\ .\end{aligned}$$

Therefore, using this last equation within (5.3.6) and taking advantage of the induced cancellations, we obtain that:

$$\frac{dQ^{(I)}}{dQ_0}(x_I, v_I) = \exp\{\Delta H(x_0, v_0) - \Phi(x_I)\} \ . \tag{5.3.8}$$

So, we have proven that the leapfrog mappings preserve the absolute continuity properties of $Q(dx, dv)$ with the particular density for $Q^{(I)}(dx, dv)$ found in (5.3.8). Using this expression for $Q^{(I)}(dx, dv)$ and following the exact steps in the proof of Theorem 2.2.1, we can obtain that the Markov transitions on the x-coordinate determined by the joint-update advanced HMC method preserve the target distribution for x.

5.4 Results

In this Section we apply the algorithms that hitherto we have introduced in this Chapter. In particular, we apply the joint-update scheme as summarised in Table 5.2. The specific diffusion model that we will be using is the fractional Ornstein-Uhlenbeck process:

$$dX_t = \kappa(\mu - X_t)dt + \sigma dB_t^H,$$

observed with error, so that:

$$Y_{t_i} = X_{t_i} + N(0, \xi^2) \ .$$

Our aim is to infer parameters $\theta = (\kappa, \mu, \sigma, x_0)$ and H, given observations Y. The extra parameter x_0 is the starting value of the diffusion process.

Note 5.4.1. *The herewith suggested signal and data dynamics could be used to model the Chicago Board Options Exchange Market Volatility Index (VIX), as originally described in [12]. Since VIX data are computed by composing a series of indexes, the quoted values could be modelled as observations from the underlying stochastic volatility with some error. Additionally, fBm could be a good choice for modelling the underlying volatility process as suggested for instance in work [20].*

To test the algorithm, we first generate data Y with known parameters:

$$\kappa = 0.03; \quad \mu = -3; \quad \sigma = 0.08;$$
$$H = 0.85; \quad \xi = 1; \quad X_0 = 3 \ ,$$

for 500 observations, at regular intervals of step-size 1. We used a joint update scheme combined with a data augmentation scheme where $\delta = 0.05$ (making the total number of points generated per iteration 9980), we run it for $20,000$ MCMC iterations.

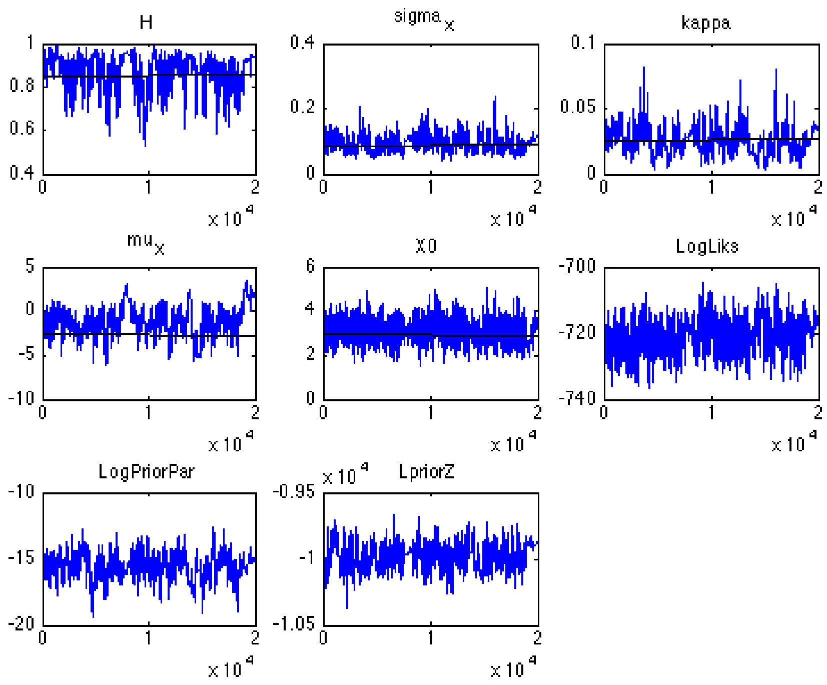

Figure 5.4: Trace Plots for Joint-Update HMC over 20,000 iterations. The black line shows the true parameters values.

Figure 5.4 shows HMC trace plots for all variables of interest, Figure 5.5 shows estimations of the marginal densities and bivariate traceplots for the MCMC points. The output suggests that the algorithm is very effective at exploring the posterior distribution in this case. The trace plots appear to have good mixing. To understand better the algorithmic performance, we ran the algorithm again but this time with parameter σ fixed. We can see from the trace plot in Figure 5.6, that we now have a better mixing for parameter H. We show the related density estimates in Figure 5.7.

Additionally, we repeated the same experiment with $H < 0.5$ using the following parameter values:

$$\kappa = 0.03; \quad \mu = -3; \quad \sigma = 0.08;$$
$$H = 0.35; \quad \xi = 1; \quad X_0 = 3 \ ,$$

for 500 observations, at regular intervals of step-size 1. We can see in Figure 5.8 and Figure 5.9 shows that there is some relatively bad mixing for variables H and σ. Fixing σ again leads to better results as you can see in Figures 5.10 and 5.11.

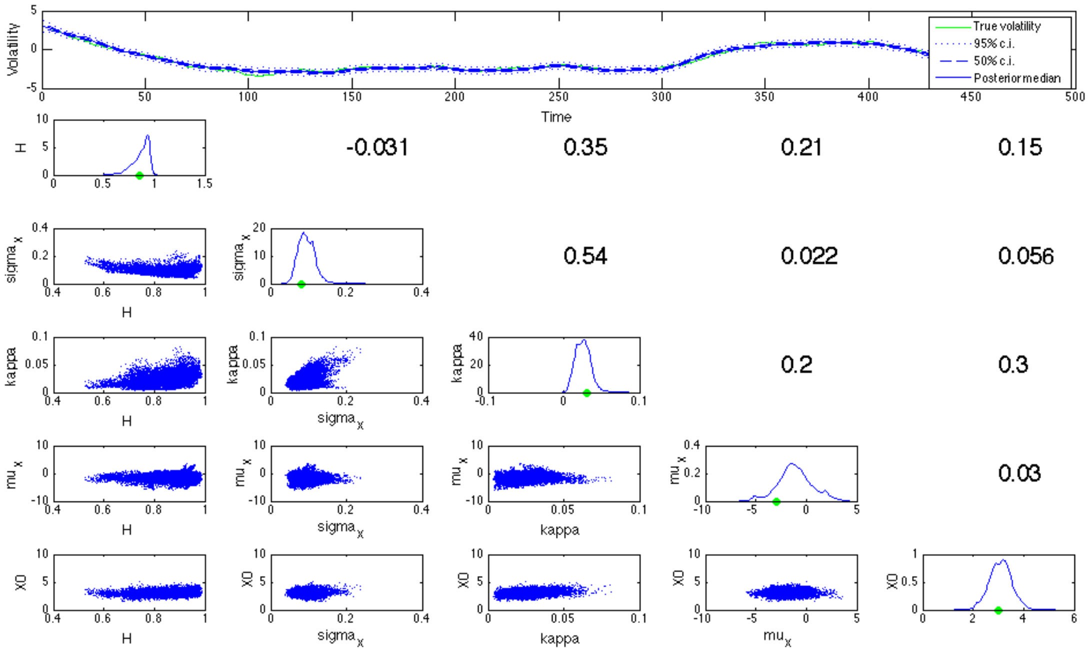

Figure 5.5: Distributions of simulated values. The green dots show the true parameter values.

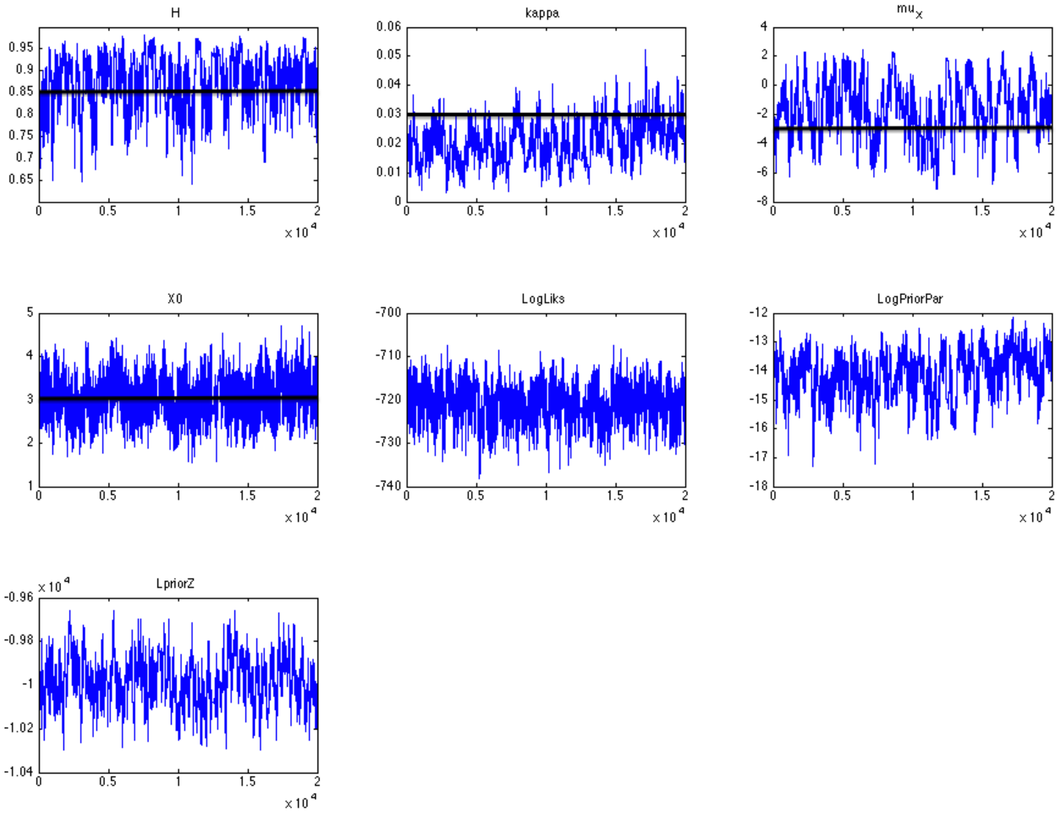

Figure 5.6: Trace Plots for Joint-Update HMC sampler with 20,000 iterations with σ fixed. The solid black lines show the true parameter values.

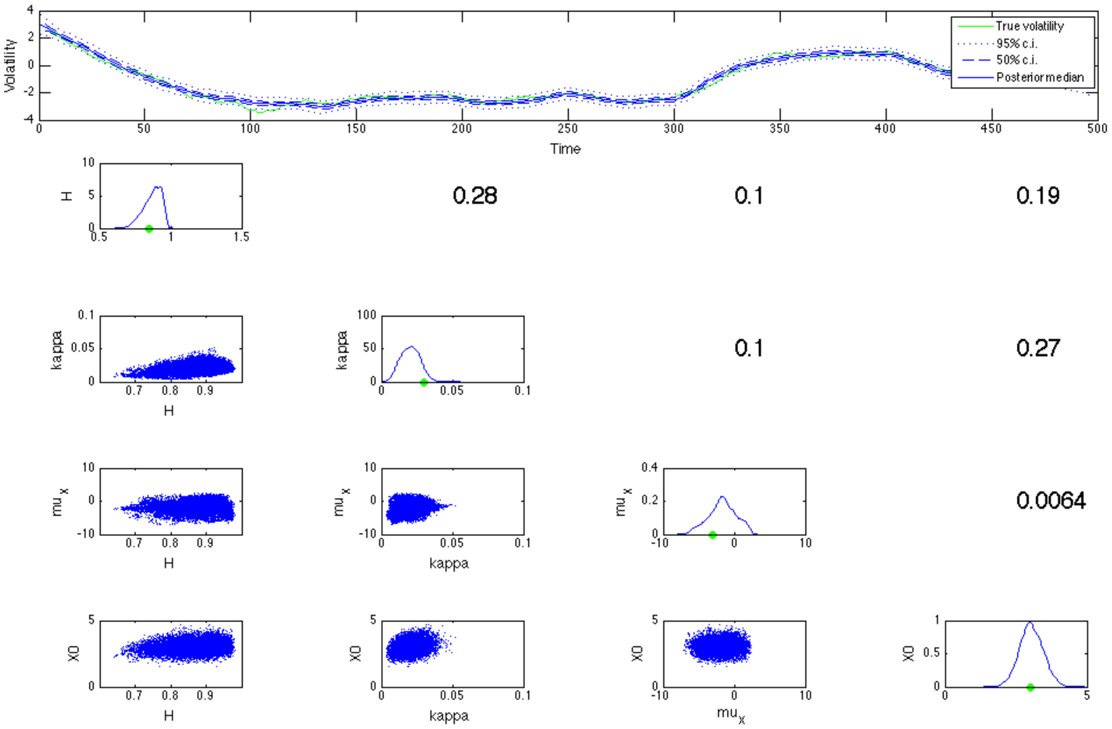

Figure 5.7: Distributions of simulated values with σ fixed. The green dots show the true parameter values.

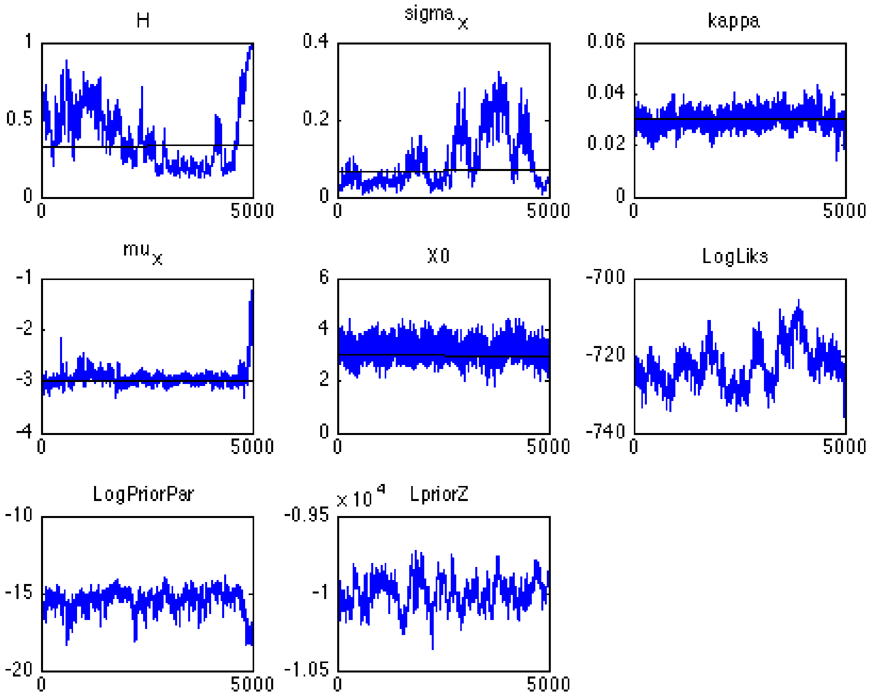

Figure 5.8: Traceplot of simulated values where line represents theoretical true values for $H = 0.35$.

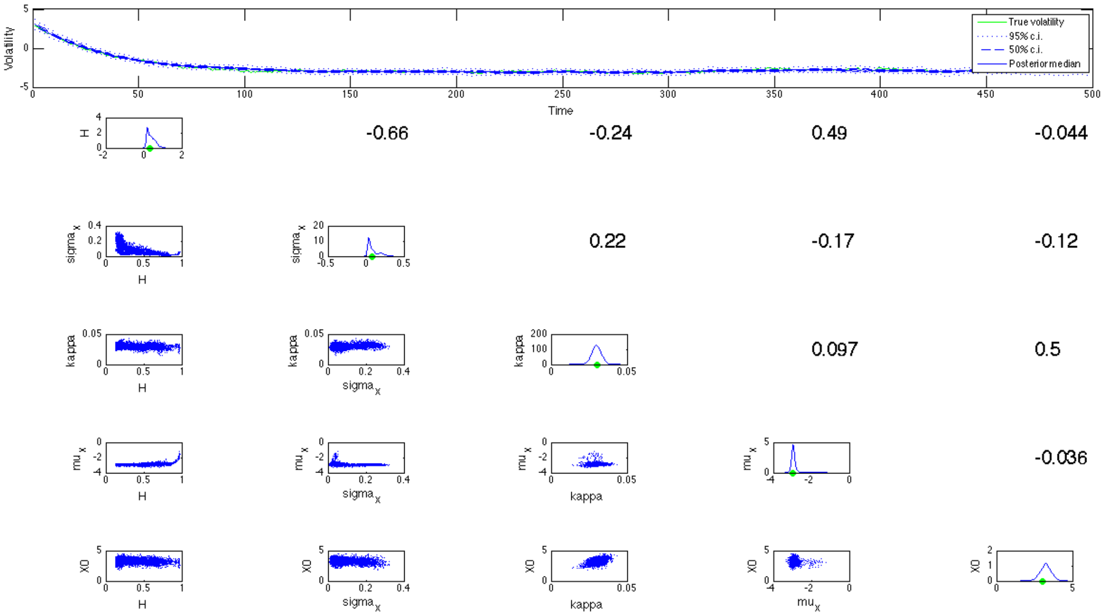

Figure 5.9: Distributions of simulated values where dot represents theoretical true values for $H = 0.35$.

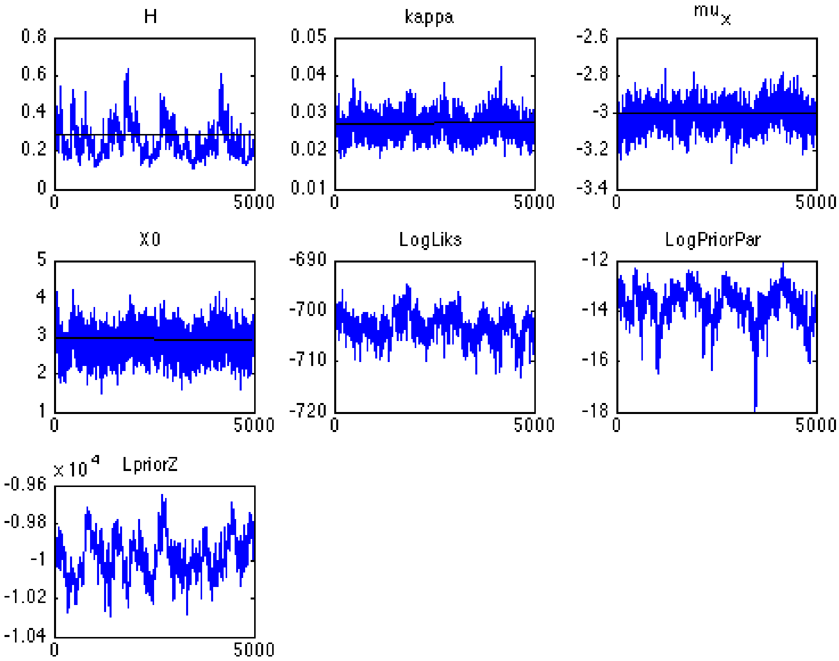

Figure 5.10: Traceplot of simulated values where line represents theoretical true values for $H = 0.35$ and fixed σ.

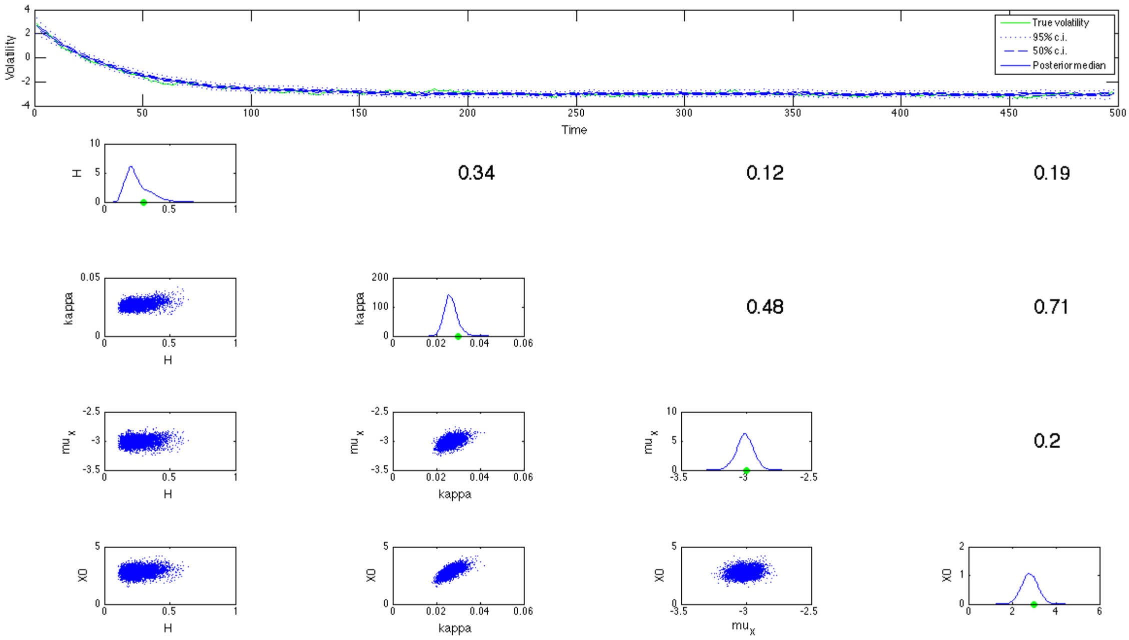

Figure 5.11: Distributions of simulated values where dot represents theoretical true values for $H = 0.35$ and fixed σ.

In general, the Joint-Update HMC sampler appears to be very effective when σ is fixed. When σ varies, a deeper exploration of the method will be useful to improve algorithmic performance.

5.5 Summary

In this Chapter we implemented a novel joint scheme update to sample from a path observed with error, where the path itself was driven by an fBm instead of a Wiener process. We defined fBm as a generalisation of Brownian motion allowing for dependence between innovations, this is done using the additional Hurst parameter H, such that:

- if $0 < H < \frac{1}{2}$, then the increments are negatively correlated;
- if $H = \frac{1}{2}$ then the increments are independent, and B^H is simply a Wiener process;
- if $\frac{1}{2} < H < 1$ then the increments are positively correlated (i.e. $\gamma_\delta(j) > 0$).

fBm is a building block for useful non-Markovian models in many real world applications where Wiener's no memory property may prove unrealistic for many practical applications. Due to its long-range dependence, using blocking strategies within MCMC algorithms adds another level of complexity since, unlike, the Markovian SDE case, blocks are correlated with each other. Thus, blocking may prove computationally intensive when using fBm. HMC superior scaling with respect to the path lengths versus MALA and RWM proved useful.

In previous chapters, we had defined our Advanced Algorithms to work for target distributions that can be defined as changes of measures w.r.t. a Gaussian measure. Here, we are inferring parameters from a SDE that is being driven by fBm where it is not immediately obvious that it can be defined w.r.t a Gaussian measure. We achieved this by a well-planned use of the Davies-Harte algorithm to provide the mapping between fBm and uncorrelated white noise that we used to decouple the a-priori involved model parameters from the high-dimensional latent variables. We examined other methods for simulating fBm but concluded that Davies-Harte algorithm provided a convenient mapping between Gaussian white noise and fBm.

Finally, we provided an example where all this properties came together by implementing a join-update algorithm to sample parameters from a path observed with error. The implemented algorithm worked well in most cases, but there were some issues with more complex models, specifically when it had to sample both the diffusion coefficient and the Hurst coefficient. This shows that there might be some issues with high correlation between those parameters but we leave it to the reader as a topic for further research.

Chapter 6

Conclusions and Further Work

6.1 Conclusions

The advanced MCMC algorithms defined in this thesis are relevant for target distributions defined as change of measures from Gaussian laws. This provides us with a lot of flexibility to apply these methods to a wider range of problems; including models driven by Stochastic Differential Equations (SDEs). In our thesis we focused on developing and testing MCMC algorithms for simulation and inference in a variety of SDE-driven models because SDEs are useful for modelling a wide variety of problems[1]. The continuous-time high-frequency nature of SDE sample paths means that traditional MCMC methods can often be unsuitable for the task at hand. The advanced MCMC algorithms shown in this thesis are well-defined on the infinite-dimensional path-space, thus, have superior properties in terms of computationally complexity when compared with standard MCMC methods. In this thesis we presented a number of proofs and results on the well-definition, suitability and computational efficiency of these algorithms.

More specifically, the advanced algorithms were well-defined for target distributions Π on general (separable) Hilbert spaces $\mathcal{H}$ that were defined as a change of measure with respect to a Gaussian one $\Pi_0 = N(0, \mathcal{C})$, i.e.:

$$\frac{d\Pi}{d\Pi_0}(x) = \exp\{-\Phi(x)\}, \tag{6.1.1}$$

where $\Phi(x)$ is a function defined on $\mathcal{H}$. As a result of the well-definition, advanced algorithms have *mesh-free* mixing properties, that is: their convergence time does not deteriorate when the dimension of the path increases (when discretized for computer purposes). Importantly, using advanced methods, SDEs are often discretized when using computational methods to sample from them, so

[1] See e.g. [65, 53] for example applications, sampling methods and mathematical properties

being *mesh-free* means that as the discretization becomes finer the mixing time remains unchanged. Our methods were built on earlier research by [7, 8, 78] which made a significant contribution on the development of several advanced MCMC algorithms. We expanded these contributions in several directions in this thesis.

Our contributions were in four main areas:

- First, we presented a new proof for the well-posedness of advanced Hybrid Monte Carlo (HMC), which is simpler and more direct when compared to the one in [7]. This allowed us to verify the validity of the required assumptions for well-posedness in several practical applications.

- Second, by comparing analytically and with numerical examples the computational costs of different advanced MCMC algorithms, we showed the very interesting result that both advanced Random Walk Metropolis (RWM) and Metropolis-adjusted Langevin Algorithm (MALA) have similar complexity when applied to 'long' diffusion paths, whereas HMC is more efficient than both of them. Thus, a well planned use of the derivative can have a big impact on the effectiveness of the selected computational method[2].

- Third, we demonstrated that the Golightly-Wilkinson transformation can be applied to a wider range of applications than the Lamperti when using HMC algorithms to sample from complex target distributions such as SDEs with general diffusion coefficients. We explored a range of uses for HMC. One of them by using HMC to sample from more complex target distributions, such as SDEs with general diffusion coefficients (as opposed to those with constant diffusion coefficients). This direction required a path tranform known as the Golightly-Wilkinson transformation, which effectively maps a Weiner-like process to an SDE with a general diffusion coefficient. Previous transformations, like the Lamperti, require the diffusion process to be reducible (see e.g. [2]), while the Golightly-Wilkinson applies in a much wider range of applications. We also used this transformation for prior decoupling when we employed HMC within a joint update algorithm.

- Four, we implemented a novel joint scheme update to sample from a path observed with error, where the path itself was driven by an fBm instead of a Wiener process. Here HMC's scaling properties proved desirable, since, the

[2] Our analytic results have been motivated by calculations in the PhD thesis [83], where interest lies in identifying algorithmic complexity with respect to the amount of 'non-linearity' in the drift for both advanced RWM and advanced MALA

non-Markovian properties of fBm made techniques like blocking unavailable. We achieved this by a well-planned use of the Davies-Harte algorithm to provide the mapping between fBm and uncorrelated white noise that we used to decouple the a-priori involved model parameters from the high-dimensional latent variables. A fBm is a generalization of Brownian motion allowing for dependence between innovations. It is a building block for useful non-Markovian models in many real world applications, this is because the no memory property of a Wiener process may prove unrealistic for many practical applications. Due to its long-range dependence, using blocking strategies within MCMC algorithms adds another level of complexity since, unlike, the Markovian SDE case, blocks are correlated with each other. Thus, blocking may prove computationally intensive when using fBm, and this was a setup where the HMC superior scaling with respect to the path lengths versus MALA and RWM proved useful.

Several algorithms have been developed to sample a fBm path, and some of them are based on the Cholesky decomposition of the related covariance matrix. In this thesis we focused on the Davies-Harte method. This method makes a clever use of a Fast Fourier Transform (FFT) to achieve an efficiency[3] of $O(N \log N)$ when simulating fBm, which is an important improvement over other algorithms. The Davies-Harte method provided a mapping between a fBm path and a vector of uncorrelated white noise, therefore, in the context of a joint-update algorithm, it was also used to decouple the a-priori involved model parameters from the high-dimensional latent variables.

Finally, we provided an example where all these properties came together to produce an effective sampling algorithm in the non-Markovian setup. We implemented a joint-update algorithm to sample parameters from a path observed with error, where the path itself was driven by a fBm instead of a Wiener process. Because of this, blocking is computationally intensive and consequently the advanced HMC proved more effective than RWM or MALA. Additionally, we used a modified version of the Golightly-Wilkinson transformation alongside the Davies-Harte algorithm to decouple the parameters from the latent variables a-priori. After completing the algorithm we showed numerically that it worked very effectively in the comparisons we continued to perform.

[3] N is the number of discretized times considered along the path

6.2 Further Work

In this thesis we have focused on MALA and, mainly, HMC algorithms. We have discussed advanced versions adapted to be effective on diffusion pathspaces or general settings where the target distributions are change of measure from Gaussian, and we have illustrated their superiority versus corresponding standard MCMC algorithms. We have also applied the method empirically in a number of diffusion models. But, still, there are many directions for further algorithmic advancements that future research could investigate, in this chapter we indicate some of the paths that further work could take.

HMC seems to be particularly effective at providing big steps in complex high-dimensional state spaces, thus, providing very good mixing, but such efficiency comes at a cost. One issue is that the algorithm involves important user-specified parameters, namely the time horizon for the Hamiltonian dynamics T, the step-size h and the mass matrix M. This has an effect on computational power. Some recent algorithms in the literature that try to address this issue: One that has attracted a lot of interest is the No U-turn Sampler (NUTS) in [50], which I will discuss in the sequel. Another issue is that HMC requires the specification of the mass matrix, denoted M in the previous chapters. Indeed, the choice of mass matrix M for standard HMC is an area of study all onto itself. We will discuss one fairly recent field of research on this matter involving Riemannian-manifold HMC methods [42]. In the case of fractional Brownian motion models we have observed that many times it is not reasonable to assume a constant Hurst parameter over long periods of time, so there is plenty of room for research in this issue alone.

In the sequel, we highlight a number of research directions that could be considered in the near future, they are related to the above discussion, and are relevant to the subject of this thesis.

6.2.1 Non-Constant Hurst Parameter

In Chapter 5 we developed a joint update scheme to infer the parameters of a fBm-driven diffusion model using VIX data, we assumed a modelling structure with unknown but fixed Hurst parameter H. Yet, one can reasonably expect that the Hurst parameter H will not be constant over a long enough time period of observations. Thus, one can try to develop a more realistic model allowing for changes in the value of H. One alternative can be to adopt a time-varying framework using an autoregressive model. That is, we can construct, for instance, the following model:

$$Y_{t_i} = X_{t_i} + N(0, \xi^2) \ ,$$

for iid Gaussian errors with variance $\xi^2 > 0$ and a fractional Ornstein-Uhlenbeck volatility process:

$$dX_t = \kappa(\mu - X_t)dt + \sigma dB_t^{H_t} \ . \tag{6.2.1}$$

for appropriate parameters (κ, μ, σ). The Hurst parameter could be modelled to change with time, say for instance, on a monthly basis so that (with a slight abuse of notation):

$$H_m = \mu_H + \varphi(H_{m-1} - \mu_H) + \varepsilon_m, \quad \varepsilon_t \sim N(0, \tau^2) \ ,$$

for parameters $\varphi \in (-1, 1)$ and $\mu_H \in (0, 1)$, $\tau^2 > 0$. In between months (continuing with this example) the value of the Hurst parameter can be assumed to be fixed. To complete the model, we need to specify the joint distribution structure of the segments of the SDE (6.2.1) over the different months. The obvious choice is to assume independency over the paths of the fractional Brownian motion with the different Hurst parameters. In this example we have decided to define H as an AR(1) model for illustration purposes, but other relevant time series models could have been chosen.

To perform Bayesian inference for this model we need to appropriately adjust the MCMC framework we described in Chapter 5. It seems that using the Davies and Harte sampling method in this context will be inappropriate, as using a stream of iid standard Gaussians $\{Z_i\}_{i=1}^{2N}$ to cover the complete time period under consideration, by using sub-blocks of the Z's for each of the different sub-periods of constant H, it would produce discontinuities in the conditional likelihood function $p(Y|Z, \theta)$ with θ denoting all model parameters. If such is the case, maybe a sequential-in-time generation of the fBm will be more appropriate, using, for instance, the Hosking method in [51], which, however, will be of cost $\mathcal{O}(N^2)$. In general, this direction is very interesting and is left for future research.

6.2.2 No U-Turn Sampler

The motivation for NUTS stems from the need to specify effectively the time horizon parameter T within HMC. Setting a small time horizon T risks inducing random walk behaviour, whereas setting a too large time horizon T risks having the Hamiltonian trajectory turning back towards its starting point, thus, wasting computational power. In Figure 6.1 we illustrate graphically the effect of choice of

T, with the green arrows representing the 'desired' Hamiltonian trajectory, while the red arrows are the unwanted leapfrog steps where the trajectory performs a U-turn and goes back to its starting position.

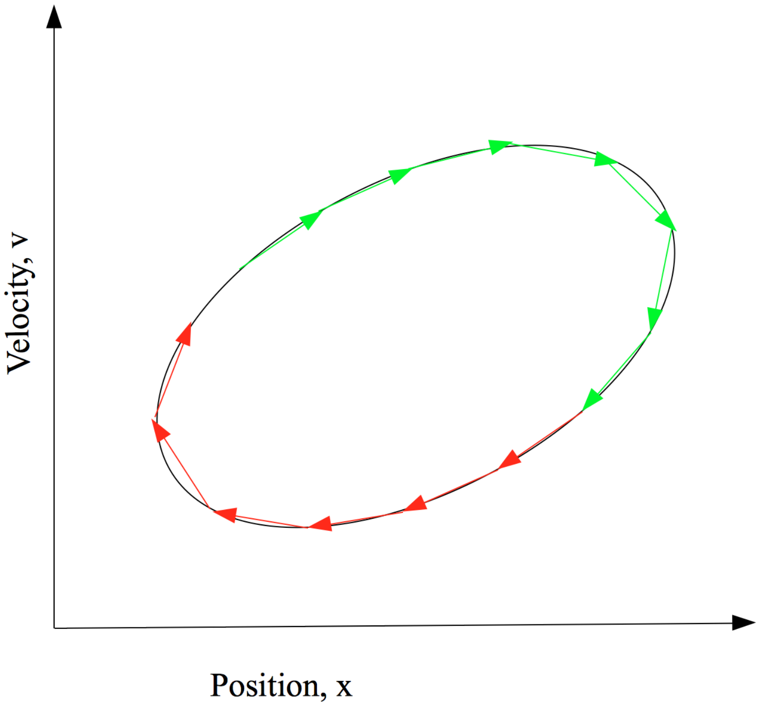

Figure 6.1: An example of too many Leapfrog steps.

We don't plan to cover the No U-Turn Sampler (NUTS) in full detail here, instead we encourage the reader to seek the source material [50]. Also, we recommend the incredibly detailed open-source package STAN in [77] accompanying the method and that provides a lot of details on the implementation of this algorithm. Nevertheless, we now provide a brief intuition on the main aspects of the algorithm.

NUTS, originally introduced in [50], aims to improve upon standard HMC by automatically selecting (on-the-fly) a varying time horizon T. The basic idea about when to stop the leapfrog steps is very simple. We want to continue applying leapfrog steps as long as the current location, say x', is getting further away from the starting location x. Correspondingly, we want to terminate the algorithm when this distance starts decreasing. To decide this, the algorithm uses the dot product of the current velocity vector v' against the vector of the difference between the initial position and current location $(x' - x)$. This is because this inner product corresponds precisely to the rate of change of the the squared distance, that is:

$$\frac{d}{dt}\frac{(x'-x)\cdot(x'-x)}{2} = (x'-x)\cdot\frac{d}{dt}(x'-x) = (x'-x)\cdot v' \ . \tag{6.2.2}$$

When this measure becomes negative the square distance starts decreasing, signalling the point where the Hamiltonian trajectory should be terminated.

It is certainly tempting to simply run a standard HMC, and then for each step keep track of $(x' - x) \cdot \rho'$ over a number of leapfrog steps which are stopped when the inner product becomes negative. Then accept/reject the proposed location via the standard HMC acceptance probability. This may ensure that the trajectory will avoid a U-turn, however such algorithm will typically not have the desired invariant distribution and will give wrong samplers.

The main contribution of [50] is precisely that the authors have developed an algorithm that ensures that the leapfrog integration of the Hamiltonian dynamics stops when starting a U-turn, while at the same time making sure that the target distribution is the invariant distribution of the induced dynamics. Without going into full details, we briefly mention here that NUTS uses the leapfrog integration to trace out a path forward or backwards in fictitious time, first running forwards or backwards 1 step, then forwards or backwards 2 steps, then forwards or backwards 4 steps, and so on. This doubling process implicitly produces a balanced binary tree where every leaf node is a position and velocity state (x, v). This doubling process is stopped when the sub-trajectory from the leftmost to the rightmost nodes of any balanced subtree of the overall binary tree starts to double back onto itself. At this point NUTS stops the simulation and samples from among the set of points computed during the simulation[4].

NUTS algorithm provides a very interesting contribution to the HMC machinery. Since the Hamiltonian trajectories are stopped before they can double-back, there is a good argument that the induced method could run at least as efficiently as a finely tuned standard HMC. This is certainly confirmed in the numerical study shown in Section 4.4 of [50], where at its worst NUTS performed as well as HMC and at its best, performed three times better. We want to emphasize, that besides the improved mixing, an important aspect of NUTS is that it allows for the specification of the time horizon parameter on-the-fly.

At the moment NUTS has been setup with the framework of a standard HMC algorithm. For the purposes of this thesis, it would certainly be of interest to develop a NUTS version for the advanced HMC, so that the method also becomes effective for high-dimensional target distributions defined as change of measure from Gaussian laws. Indeed, it would be interesting to bring some of the improvements that NUTS can deliver within the context of sampling diffusion sample

[4] A pseudocode detailing further the algorithm can be found in Section 3 of [50]

paths. It is unclear if one would experience the same increase of efficiency as seen in [50], so a numerical study would be needed. We leave this direction open for future research.

6.2.3 Riemannian Manifold MCMC

We now look at the issue of the specification of the mass matrix M within the HMC algorithm. In the derivation of advanced HMC we chose a mass matrix $M = \Sigma^{-1}$, with Σ being the covariance operator of the reference Gaussian measure. This selection is effective at delivering a method with mesh-free mixing time, however, it is blind to the data and does not adjust to the actual covariance structure of the posterior distribution. Recent methodological developments for standard HMC have looked at challenging target distributions with complex local-correlation structures. Such developments exploit mathematical theory on Riemannian manifolds and are based on Hamiltonian dynamics with non-constant mass matrix. This methodology introduced in [42] shows to be effective for target distributions as prescribed above. One apparent direction of research here is to combine the strengths of an advanced with a manifold HMC, with the aim of constructing manifold-based algorithms which promise to be robust when addressing high-dimensional target distributions defined as change of measure from Gaussian laws.

A first preliminary attempt in such research direction has already been made in the recent work of [4]. Indeed, the method in [4] looks at the Manifold Metropolis-adjusted Langevin algorithm (MMALA), originally introduced in [42], and suggests an advanced version of it which is robust in high-dimensions (labeled ∞-MMALA). We will cover here the basics of MMALA and ∞-MMALA very briefly, for a more detail explanation the reader is advised to consult the source material in [42] and [4] respectively.

Same as with our standard setup for high-dimensional target distributions, we wish to simulate a distribution defined on a Hilbert space $\mathcal{H}$ as follows:

$$\Pi(x) \propto \exp\{l(x)\} = \exp\big\{ -\Phi(x) - \tfrac{1}{2}\langle x-\mu, L(x-\mu)\rangle \big\} \ , \qquad (6.2.3)$$

for a prior mean $\mu \in \mathcal{H}$, a mapping $\Phi : \mathcal{H} \mapsto \mathbb{R}$, with $L = \Sigma^{-1}$ being the inverse of the covariance matrix of the reference Gaussian measure. Similarly to MALA, its manifold version MMALA uses the dynamics of the Langevin SDE but this time as defined on the manifold space generated by an appropriately chosen metric tensor $G(x)$. In particular, the analytical expression for the SDE on the manifold is as follows:

$$dx = \tfrac{1}{2}\tilde{\nabla} l(x)\, dt + d\tilde{b}\ , \tag{6.2.4}$$

with $\tilde{\nabla} = G^{-1}(x)\nabla$ corresponding to the analogue of differentiation along the manifold and $d\tilde{b}$ denoting infinitesimal increments of a Brownian motion on the manifold. We assume here that the state-space is $x \in \mathbb{R}^N$ so that $G(x)^{-1}$ is a symmetric positive-definite matrix in $\mathbb{R}^{N\times N}$. Making use of the detailed expression for a Brownian motion on a manifold specified by the tensor $G(x)$, the work in [42] shows that the manifold dynamics in (6.2.4) can be equivalently be expressed in terms of the following standard SDE on the Euclidean space $\mathbb{R}^N$:

$$dx = G(x)^{-1}\big\{\, \tfrac{1}{2}\nabla l(x) + \tfrac{1}{2}\nabla \log |G(x)| + \nabla\ \big\} dt + G(x)^{-1/2} db\ , \tag{6.2.5}$$

where db now represents increments of standard Brownian motion and $|G(x)|$ is the determinant of $G(x)$. One now needs to discretize the continuous-time Langevin SDE to come up with a proposal for the MCMC algorithm. MMALA considers a standard Euler scheme. Also, it is suggested in [42] that one can keep only the term $\frac{1}{2}\, G(x)^{-1} l(x)$ from the drift function in (6.2.5) and still end up with a powerful method, thus avoiding the expensive computation of the remaining drift terms. To develop the advanced algorithm ∞-MMALA, when discretizing the SDE (6.2.5) we take a semi-implicit approach similar to the one for advanced MALA in Section 2.1, that is, we have (using also a single drift term as explained above):

$$\begin{aligned} x' - x = \tfrac{1}{2} G(x)^{-1}\big\{ - G(x)\tfrac{x'+x}{2} + G(x)x + \nabla l(x)\ \big\} h \\ + \sqrt{h}\, N(0, G(x)^{-1})\ , \end{aligned} \tag{6.2.6}$$

for a step-size $h > 0$, which can be equivalently written as:

$$x' = \tfrac{1-h/4}{1+h/4}\, x + \tfrac{h/2}{1+h/4}\, S(x) + \tfrac{\sqrt{h}}{1+h/4}\, N(0, G(x)^{-1}) \tag{6.2.7}$$

where we have set:

$$S(x) = -G(x)^{-1}\{\nabla\Phi(x) - (G(x) - L)x - L\mu\}\ . \tag{6.2.8}$$

We stop here, momentarily only, for the development of ∞-MMALA, and discuss briefly the choice of $G(x)$. An interesting thing to notice is that the choice $G(x) = L$ yields exactly ∞-MALA, thus ∞-MMALA can be thought of as a generalization of the advanced MALA algorithm described in the main part of this thesis, now allowing for a non-constant mass matrix. An often effective approach, which was suggested in [42], is to use the expected Fisher information as the metric tensor:

$$\begin{aligned} -E_{Y|x}\nabla^2 l(x) &= E_{Y|x}\nabla_x^2\Phi(x;Y) + L \\ &= E_{Y|x}\left[\nabla_x\Phi(x;Y)\{\nabla_x\Phi(x;Y)\}^\top\right] + L \ , \end{aligned} \tag{6.2.9}$$

where we write $\Phi(x;Y) = \Phi(x)$ to emphasise the dependence of Φ on some data Y when given a particular model. In the context of high-dimensional $x \in \mathbb{R}^N$ this choice of geometric tensor can sometimes lead to large computational costs as a function of N, thus, it is important to keep in mind that one should try to effectively balance improving algorithmic mixing with increased computational costs.

We can now return to (6.2.7) and complete the discussion for ∞-MMALA. We have yet to illustrate that proposal (6.2.7) will have a positive acceptance probability even in infinite-dimensions. The work in [4] is focused exactly on this point. The proof provided there follows a very similar logic used to prove the well-posedness of ∞-MALA in [8] and also shown in Chapter 2 of this thesis, building upon the generalised definition of the Metropolis-Hasting ratio in [81]. Briefly, we define the bivariate probability measure on $\mathcal{H} \times \mathcal{H}$:

$$\mu(dx, dx') = \Pi(dx)Q(x, dx') \ ,$$

with $Q(x,\cdot)$ denoting the transition probability law rising via the dynamics in (6.2.7), and the corresponding symmetric measure $\mu^\top(dx, dx') = \mu(dx', dx)$. Following [81], if $\mu \simeq \mu^T$ (with '$\simeq$' denoting absolute continuity between probability measures) then the acceptance probability is well-defined and equal to:

$$1 \wedge \frac{d\mu^\top}{d\mu}(x, x') \ . \tag{6.2.10}$$

So, it remains to specify the conditions under which $\mu \simeq \mu^\top$ and find the precise Radon-Nikodym derivative $(d\mu^\top/d\mu)(x, x')$. The analytical derivations are shown in [4].

The development of the ∞-MMALA algorithm marks a stepping stone toward the development of other high-dimensional robust manifold methods. The next logical step would be to adapt the more complicated Riemann Manifold Hybrid Monte Carlo (RMHMC) so that it is robust in increasing dimensions, which would be referred to as ∞-RMHMC or Advanced RMHMC. This has not been done in the literature yet, mainly due to the relative complexity of the acceptance probability in high-dimensions. It is also unclear if the increase in computational costs related with manifold methods will yield sufficient improvement in mixing times. Additionally, there is plenty of room for experimentation with combinations of various algorithms. It would be interesting, for instance, to see if high-dimensional manifold methods can be used with algorithms like NUTS, and if substantial increases in efficiency can be achieved in such direction.

Bibliography

[1] Odd O. Aalen and Håkon K. Gjessing, *Survival models based on the Ornstein-Uhlenbeck process*, Lifetime Data Anal. **10** (2004), no. 4, 407–423.

[2] Yacine Aït-Sahalia, *Closed-form likelihood expansions for multivariate diffusions*, Ann. Statist. **36** (2008), no. 2, 906–937.

[3] Jasbir Arora, *Practical mathematical optimization: An introduction to basic optimization theory and classical and new gradient-based algorithms*, Structural and Multidisciplinary Optimization **31** (2006), 249–249.

[4] Alexandros Beskos, *A stable manifold MCMC method for high dimensions*, Statistics & Probability Letters **90** (2014), 46–52.

[5] Alexandros Beskos, Konstantinos Kalogeropoulos, and Erik Pazos, *Advanced MCMC methods for sampling on diffusion pathspace*, Stochastic Processes and their Applications **123** (2013), no. 4, 1415–1453.

[6] Alexandros Beskos, Natesh Pillai, Gareth Roberts, Jesus-Maria Sanz-Serna, and Andrew Stuart, *Optimal tuning of the Hybrid Monte Carlo algorithm*, Bernoulli **19** (2013), no. 5A, 1501–1534.

[7] Alexandros Beskos, FJ Pinski, JM Sanz-Serna, and AM Stuart, *Hybrid Monte Carlo on Hilbert spaces*, Stochastic Processes and their Applications **121** (2011), no. 10, 2201–2230.

[8] Alexandros Beskos, Gareth Roberts, Andrew Stuart, and Jochen Voss, *MCMC methods for diffusion bridges*, Stoch. Dyn. **8** (2008), no. 3, 319–350.

[9] Francesca Biagini, Yaozhong Hu, Bernt Øksendal, and Tusheng Zhang, *Stochastic calculus for fractional Brownian motion and applications*, Probability and its Applications (New York), Springer-Verlag London, Ltd., London, 2008.

[10] Patrick Billingsley, *Probability and measure*, third ed., Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, Inc., New York, 1995, A Wiley-Interscience Publication.

[11] Vladimir I. Bogachev, *Gaussian measures*, Mathematical Surveys and Monographs, vol. 62, American Mathematical Society, Providence, RI, 1998.

[12] Menachem Brenner and Dan Galai, *New financial instruments for hedging changes in volatility*, Financial Analysts Journal (1989), 61–65.

[13] Stephen P Brooks and Gareth O Roberts, *Convergence assessment techniques for Markov Chain Monte Carlo*, Statistics and Computing **8** (1998), no. 4, 319–335.

[14] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng, *Handbook of Markov Chain Monte Carlo*, CRC Press, 2011.

[15] Bradley P Carlin, Nicholas G Polson, and David S Stoffer, *A Monte Carlo approach to nonnormal and nonlinear state-space modelling*, Journal of the American Statistical Association **87** (1992), no. 418, 493–500.

[16] Chris K Carter and Robert Kohn, *On Gibbs sampling for state space models*, Biometrika **81** (1994), no. 3, 541–553.

[17] Patrick Cheridito, *Regularizing fractional Brownian motion with a view towards stock price modelling*, Ph.D. thesis, SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH, 2001.

[18] Siddhartha Chib, Michael K Pitt, and Neil Shephard, *Likelihood based inference for diffusion driven state space models*, Por Clasificar (2006), 1–33.

[19] Jean-Paul Chiles and Pierre Delfiner, *Geostatistics: Modelling spatial uncertainty*, **497** (2009).

[20] Alexandra Chronopoulou and Frederi G Viens, *Estimation and pricing under long-memory stochastic volatility*, Annals of Finance **8** (2012), no. 2-3, 379–403.

[21] ———, *Stochastic volatility and option pricing with long-memory in discrete and continuous time*, Quantitative Finance **12** (2012), no. 4, 635–649.

[22] Mary Kathryn Cowles and Bradley P. Carlin, *Markov Chain Monte Carlo convergence diagnostics: a comparative review*, J. Amer. Statist. Assoc. **91** (1996), no. 434, 883–904.

[23] John C Cox, Jonathan E Ingersoll Jr, and Stephen A Ross, *A theory of the term structure of interest rates*, Econometrica: Journal of the Econometric Society (1985), 385–407.

[24] Peter F Craigmile, *Simulating a class of stationary Gaussian processes using the Davies-Harte algorithm, with application to long memory processes*, Journal of Time Series Analysis **24** (2003), no. 5, 505–511.

[25] Giuseppe Da Prato and Jerzy Zabczyk, *Stochastic equations in infinite dimensions*, vol. 152, Cambridge university press, 2014.

[26] Robert B Davies and DS Harte, *Tests for Hurst effect*, Biometrika **74** (1987), no. 1, 95–101.

[27] Laurent Decreusefond et al., *Stochastic analysis of the fractional Brownian motion*, Potential analysis **10** (1999), no. 2, 177–214.

[28] Jérôme Dedecker and Paul Doukhan, *A new covariance inequality and applications*, Stochastic processes and their applications **106** (2003), no. 1, 63–80.

[29] P Dellaportas and Adrian FM Smith, *Bayesian inference for generalized linear and proportional hazards models via Gibbs sampling*, Applied Statistics **42** (1993), no. 3, 443–459.

[30] Bernard Delyon and Ying Hu, *Simulation of conditioned diffusion and application to parameter estimation*, Stochastic processes and their applications **116** (2006), no. 11, 1660–1675.

[31] Aurélien Deya, Andreas Neuenkirch, Samy Tindel, et al., *A Milstein-type scheme without Lévy area terms for SDEs driven by fractional Brownian motion*, **48** (2012), no. 2, 518–550.

[32] Ton Dieker, *Simulation of fractional Brownian motion*, MSc thesis, University of Twente, Amsterdam, The Netherlands (2004).

[33] CR Dietrich and Garry Neil Newsam, *Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix*, SIAM Journal on Scientific Computing **18** (1997), no. 4, 1088–1107.

[34] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth, *Hybrid Monte Carlo*, Physics letters B **195** (1987), no. 2, 216–222.

[35] Joseph Dureau, Alexandros Beskos, and Konstantinos Kalogeropoulos, *Bayesian inference for partially observed sdes driven by fractional Brownian motion*, arXiv preprint arXiv:1307.0238 (2013).

[36] Garland B Durham and A Ronald Gallant, *Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes*, Journal of Business & Economic Statistics **20** (2002), no. 3, 297–316.

[37] Eric Fournié, Jean-Michel Lasry, Jérôme Lebuchoux, Pierre-Louis Lions, and Nizar Touzi, *Applications of Malliavin calculus to Monte Carlo methods in finance*, Finance Stoch. **3** (1999), no. 4, 391–412.

[38] Crispin W Gardiner, *Handbook of stochastic methods: For physics, chemistry and the natural sciences*, vol. 25, 1986.

[39] Izrail Moiseevich Gelfand and Sergeĭ Vasil'evich Fomin, *Calculus of Variations*, Courier Dover Publications, 2000.

[40] Charles J Geyer, *Practical Markov Chain Monte Carlo*, Statistical Science (1992), 473–483.

[41] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in practice*, Interdisciplinary Statistics, Chapman & Hall, London, 1996.

[42] Mark Girolami and Ben Calderhead, *Riemann manifold Langevin and Hamiltonian Monte Carlo methods*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), no. 2, 123–214.

[43] Tilmann Gneiting, *Simple tests for the validity of correlation function models on the circle*, Statistics & probability letters **39** (1998), no. 2, 119–122.

[44] Andrew Golightly and Darren J Wilkinson, *Bayesian inference for nonlinear multivariate diffusion models observed with error*, Comput. Statist. Data Anal. **52** (2008), no. 3, 1674–1693.

[45] Geoffrey R. Grimmett and David R. Stirzaker, *Probability and random processes*, third ed., Oxford University Press, New York, 2001.

[46] Martin Hairer, Andrew M Stuart, Jochen Voss, et al., *Analysis of SPDEs arising in path sampling. II. The nonlinear case*, Ann. Appl. Probab. **17** (2007), no. 5-6, 1657–1706.

[47] Martin Hairer, Andrew M Stuart, Jochen Voss, Petter Wiberg, et al., *Analysis of SPDEs arising in path sampling. I. The Gaussian case*, Commun. Math. Sci. **3** (2005), no. 4, 587–603.

[48] Keith W Hastings, *Monte Carlo sampling methods using Markov Chains and their applications*, Biometrika **57** (1970), no. 1, 97–109.

[49] Steven L Heston, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, Review of financial studies **6** (1993), no. 2, 327–343.

[50] Matthew D Hoffman and Andrew Gelman, *Forthcoming. "the no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo"*, Journal of Machine Learning Research (2014).

[51] Jonathan RM Hosking, *Fractional differencing*, Biometrika **68** (1981), no. 1, 165–176.

[52] Ioannis Karatzas and Steven E. Shreve, *Brownian motion and stochastic calculus*, second ed., Graduate Texts in Mathematics, vol. 113, Springer-Verlag, New York, 1991.

[53] Peter E. Kloeden and Eckhard Platen, *Numerical solution of stochastic differential equations*, Applications of Mathematics (New York), vol. 23, Springer-Verlag, Berlin, 1992.

[54] Samuel C Kou, *Stochastic modelling in nanoscale biophysics: Subdiffusion within proteins*, The Annals of Applied Statistics **2** (2008), no. 2, 501–535.

[55] Will E Leland, Murad S Taqqu, Walter Willinger, and Daniel V Wilson, *On the self-similar nature of ethernet traffic*, ACM SIGCOMM Computer Communication Review, vol. 23, ACM, 1993, pp. 183–193.

[56] Martin Lysy and Natesh S Pillai, *Statistical inference for stochastic differential equations with memory*, arXiv preprint (2013).

[57] Benoit B Mandelbrot and John W Van Ness, *Fractional Brownian motions, fractional noises and applications*, SIAM review **10** (1968), no. 4, 422–437.

[58] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller, *Equation of state calculations by fast computing machines*, The journal of chemical physics **21** (1953), no. 6, 1087–1092.

[59] Yuliya S. Mishura, *Stochastic calculus for fractional Brownian motion and related processes*, Lecture Notes in Mathematics, vol. 1929, Springer-Verlag, Berlin, 2008.

[60] R. M. Neal, *Probabilistic inference using Markov Chain Monte Carlo methods*, Tech. report, University of Toronto, 1993.

[61] Radford M Neal, *Suppressing random walks in Markov Chain Monte Carlo using ordered overrelaxation*, Learning in graphical models, Springer, 1998, pp. 205–228.

[62] Andreas Neuenkirch, Samy Tindel, and Jérémie Unterberger, *Discretizing the fractional Lévy area*, Stochastic Processes and Their Applications **120** (2010), no. 2, 223–254.

[63] Ilkka Norros, Petteri Mannersalo, and Jonathan L Wang, *Simulation of fractional Brownian motion with conditionalized random midpoint displacement*, Advances in Performance Analysis **2** (1999), no. 1, 77–101.

[64] Ivan Nourdin and Thomas Simon, *On the absolute continuity of one-dimensional SDEs driven by a fractional Brownian motion*, Statistics & probability letters **76** (2006), no. 9, 907–912.

[65] Bernt Oksendal, *Stochastic differential equations: An introduction with applications*, Springer, 2003.

[66] B. L. S. Prakasa Rao, *Statistical inference for fractional diffusion processes*, Wiley Series in Probability and Statistics, John Wiley & Sons, Ltd., Chichester, 2010.

[67] Brian D. Ripley, *Stochastic simulation*, Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, John Wiley & Sons, Inc., New York, 1987.

[68] Gareth O Roberts, *Markov Chain concepts related to sampling algorithms*, Markov Chain Monte Carlo in practice, Springer, 1996, pp. 45–57.

[69] Gareth O Roberts and Sujit K Sahu, *Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler*, Journal of the Royal Statistical Society. Series B (Methodological) **59** (1997), no. 2, 291–317.

[70] Gareth O Roberts, Laura M Sangalli, et al., *Latent diffusion models for survival analysis*, Bernoulli **16** (2010), no. 2, 435–458.

[71] Gareth O Roberts and Osnat Stramer, *On inference for partially observed nonlinear diffusion models using the Metropolis-Hastings algorithm*, Biometrika **88** (2001), no. 3, 603–621.

[72] Gareth O Roberts and Richard L Tweedie, *Exponential convergence of Langevin distributions and their discrete approximations*, Bernoulli (1996), 341–363.

[73] L Chris G Rogers and David Williams, *Diffusions, Markov processes, and martingales. Vol. 2*, Cambridge Mathematical Library, Cambridge University Press, Cambridge, 2000, Itô calculus, Reprint of the second (1994) edition.

[74] Neil Shephard, *Partial non-Gaussian state space*, Biometrika **81** (1994), no. 1, 115–131.

[75] Neil Shephard and Michael K Pitt, *Likelihood analysis of non-Gaussian measurement time series*, Biometrika **84** (1997), no. 3, 653–667.

[76] Albert N Shiryaev and N Kruzhilin, *Essentials of stochastic finance: Facts, models, theory*, vol. 23, World scientific Singapore, 1999.

[77] Stan Development Team, *Rstan: the R interface to stan, version 2.4*, 2014.

[78] Andrew M. Stuart, Jochen Voss, and Petter Wiberg, *Fast communication conditional path sampling of SDEs and the Langevin MCMC method*, Commun. Math. Sci. **2** (2004), no. 4, 685–697.

[79] Héctor J Sussmann, *On the gap between deterministic and stochastic ordinary differential equations*, The Annals of Probability **6** (1978), no. 1, 19–41.

[80] Martin A Tanner and Wing Hung Wong, *The calculation of posterior distributions by data augmentation*, Journal of the American statistical Association **82** (1987), no. 398, 528–540.

[81] Luke Tierney, *A note on metropolis-hastings kernels for general state spaces*, Annals of Applied Probability **8** (1995), 1–9.

[82] George E Uhlenbeck and Leonard Salomon Ornstein, *On the theory of the Brownian motion*, Phys. Rev. **36** (1930), no. 5, 823–841.

[83] David White, *A comparison of MCMC methods for conditioned diffusions*, Ph.D. thesis, Mathematics Institute, University of Warwick, 2009.

[84] Walter Willinger, Murad Taqqu, Will Leland, and Daniel Wilson, *Self-similarity in high-speed packet traffic: Analysis and modelling of ethernet traffic measurements*, Statistical science **10** (1995), no. 1, 67–85.

[85] Andrew TA Wood and Grace Chan, *Simulation of stationary Gaussian processes in [0, 1]*, Journal of computational and graphical statistics **3** (1994), no. 4, 409–432.