MCMC Methods for Functions: Modifying Old Algorithms to Make Them Faster

2nd May 2017

Abstract

Markov Chain Monte Carlo methods on function spaces are useful, for example to solve inverse problems. Classical methods suffer from poor performance on function space, which makes modifications of them necessary. This essay provides an overview of certain dimension-independent methods. Discussed are applications, examples, theoretical underpinnings, and the mathematical properties behind these methods, furthermore the performance of them is analysed.

Contents

1 Introduction

This essay aims to explore and assess Markov Chain Monte Carlo (MCMC) methods that work on function spaces (extending to other infinite-dimensional, separable Hilbert spaces). The necessesity of this arises as standard MCMC methods suffer from poor performance under mesh-refinement. New 'robust' methods are needed, which are stable under mesh-refinement, and thus not require adjustment with every dimension added. Amongst the settings in which these methods can be applied are inverse problems. This is a very active field of research. One of the Bayesian contributions to this field of research will be discussed in this essay, namely MCMC methods on function spaces. This work can then be used to construct confidence intervals for possible initial states of an inverse problem, by sampling from the posterior distribution for these initial states given data.

This essay is structured as follows: settings in which these methods are useful will be discussed in section 2, where a very basic but illustrative example will be used to introduce the concept and utility of MCMC methods on function spaces, before turning to more realistic applications, the aforementioned inverse problems. Some probabilistic background will be provided in section 3, which will be necessary to understand and assess these methods. Next, two MCMC methods wil be described and analysed in section 4, where the title of this essay will be justified, the origin of these methods will be explained, and also a few remarks on possible modifications will be given - and indeed the most recent developments concerning them will be stated. The proofs of theorems 4, 5, and 6 in that section are largely the authors own work. Computational experiments will be found in section 5. These experiments should be seen as explanatory rather than as an applicable tool. To use the program in practice, adjustments to the respective setting need to be made. We conclude with a summary and an outlook.

2 Setting

In this section we assess the usefulness of MCMC methods on function spaces. We do this first by looking at a somewhat artificial example, in order to illustrate how these methods work, and to offer some insight into possible other applications. We then turn to the more advanced application of solving inverse problems.

2.1 Density estimation

We elaborate the example mentioned in [6]. Let us assume that we have data given by independent draws from a random variable that has a density function proportional to $\mathcal{N}(-3,1)\mathbb{1}_{[-10,10]} + \mathcal{N}(3,1)\mathbb{1}_{[-10,10]}$. Given only the data, we may be interested in finding the density function of this random variable. Non-Bayesian approaches for this task can be found in [4]. Instead of assuming a parametrical model, a mixture of normals say, we assume no prior knowledge whatsoever about the density. We may therefore try to infer the Fourier series[1] of the density, which

[1] We could just as well take any other series representation of the function, as long as the series is garantued to converge. Throughout this essay, the reader should think of the Fourier series as the most

is given by

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{2\pi n(x-10)}{20}\right) + b_n \sin\left(\frac{2\pi n(x-10)}{20}\right),$$

where we already adjusted the series to the given interval $[-10, 10]$. See [9] for a detailed analysis of Fourier series. The truncated series sums only over the first N terms which leads to $2N+1$ coefficients to be estimated in total. A MCMC method may now be used to sample these coefficients a_n and b_n, which may be arbitrarily many. The first questions arising are:

As a computer can't handle infinitely many terms, how many terms shall we use in practise? Shall we used a fixed N, or allow for differing N?

Shall all coefficients be similarly important, or shall we put some kind of 'importance weight' on certain terms?

While both questions will be discussed and answered in the subsequent sections, the reader should note that - generally speaking - the first coefficients of a Fourier series are more important than later ones, as the sequence of coefficients converges to 0. Therefore, it makes sense to see the first ones as 'more important' and - fortunately - truncating the series does not lead to an inferior method. Indeed, when given n data points, it wouldn't make sense to use more than n coefficients.

The example of density estimation is used for illustrative purposes, and the reader may look at the computational experiments in section 5, where the above mentioned density is estimated by the two methods introduced in section 4.

2.2 Inverse problems

Before delving into inverse problems, let us first recall what we did in the previous subsection. We took the Fourier representation of a function, and used a MCMC method to sample the coefficients.
This idea can be applied to inverse problems as well: intuitively speaking, we posit some function f_0 as the initial condition for such a problem, a partial differential equation (PDE) for example. We then use the PDE and the function f_0 to solve the forward problem, and can compare the solution of this forward problem to the data observed. Changing some coefficients in the Fourier series of f_0 gives us some proposal function $\tilde{f}$, which will lead to another solution and, as in a classical Metropolis-Hastings algorithm, we accept or reject this proposal with a certain acceptance probability, generally accepting the proposal $\tilde{f}$ if it is 'more likely' than f_0, or otherwise rejecting it with a probability somewhat proportional to 'how less likely' $\tilde{f}$ is compared to f_0, thus giving us a new iterate f_1.
Doing this often enough will yield a posterior distribution in the function space, where more mass is found on functions that are more likely to be the true initial state, as the data are more likely to have been genereated by these functions.

obvious representation of a function, though others are possible. Also see the section dealing with the Karhunen-Loève expansion.

Let us look at one example. This example is taken from [16], where a thorough discussion can be found. We consider the inverse problem for a diffusion coefficient. We are given a two-point boundary value problem

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}x}\left(k(x)\frac{\mathrm{d}p}{\mathrm{d}x}\right) &= 0, \\ p(0) = p_0, \quad p(1) &= p_1, \end{aligned} \tag{1}$$

and aim to recover the true diffusion coefficient $k(x)$. We assume $p_1 > p_0 > 0$. Given some erroneous measurements y_k at the points $0 < x_1 < x_2 < ... < x_q < 1$, and we assume the errors to be normal. Thus, we can write $y_k = p(x_k) + \eta_k$, where $p(x_k)$ is the 'true' values, and the η_k are i.i.d. zero-mean Gaussians with variance γ^2.
Note the following: given some function $k(x)$, we can solve the forward problem (1), and find the $p(x_k)$ corresponding to that diffusion coefficient. This is achieved as follows: to ensure positivity of $k(x)$ (a non-positive coefficient function doesn't make physical sense), firstly define $u(x) = \log(k(x))$, a one-to-one correspondence for positive k. Then the solution to the PDE given by (1) is

$$p(x) = (p_1 - p_0)\frac{\int_0^x \exp(-u(z))\mathrm{d}z}{\int_0^1 \exp(-u(z))\mathrm{d}z} + p_0.$$

In order to compare this to our observations, we define the observation operator $\mathcal{G}(u(x)) = (p(x_1), ..., p(x_q))^*$. The star is used to distinguish between the true values and the ones obtained by solving the forward problem.

Here, one can see the general technique when using the Bayesian framework to solve an inverse problem: a forward operator is used to solve the forward problem, given some proposal for the initial state. Depending on the problem and the data observed, one chooses an appropriate observation operator $\mathcal{G}(u)$, which will be used in MCMC algorithms, to compare the observations y to the solutions from the forward problem. The observations might be erroneous, which is taken into consideration in our observation model

$$y = \mathcal{G}(u) + \varepsilon, \tag{2}$$

where we normally assume Gaussian noise $\varepsilon \sim \mathcal{N}(0, Q_{obs})$. Q_{obs} denotes the covariance matrix of the noise. We will see in the next section how this algorithm is defined, and how the observation operator is crucial to defining the acceptance probabilities.

In our example, we assume that one has chosen coefficients of the Fourier series of $k(x)$. We start our MCMC method from these coefficients, i.e. from $k_0(x)$. Depending on the MCMC method, we propose a new $\tilde{k}(x)$, having coeffiecients which are usually close to those from k_0. We then use $\mathcal{G}(\log(k_0(x)))$, $\mathcal{G}(\log(\tilde{k}(x)))$ and the observations $\{y_k\}_{k=1}^q$ to decide whether to accept or reject the proposal. Repeating this step will then lead to a method whereby (if properly defined) we should end up with a probability distribution for each coefficient of the Fourier series, and should thus be able to define confidence intervals, to obtain the most likely coefficients (and thereby the most likely $k(x)$), and to infer anything else

possible with standard MCMC methods.

Let us now turn to the more technical part of this essay. In order to understand how these MCMC methods actually work, how they propose a new state, how the acceptance probabilities are defined, and which advantages and/or disadvantages they might have, some theoretical background is required. This background is provided in the next subsection, where we try to make some sense of the rather technical aspects needed.

3 Probabilistic interlude

In order to understand section 4, this section will provide the reader with some theorems in probability theory that will be crucial for the understanding of the MCMC methods on function spaces and their analysis in the subsequent section. Firstly, it will be justified to use a series representation for infinite Gaussian measures by making use of the Karhunen-Loève expansion, and secondly, we will state the ingenious Feldman-Hajek theorem, which states that two Gaussian measures on infinite Hilbert spaces are either mutually singular or equivalent, and furthermore it states three conditions that hold if, and only if, those two measures are equivalent.

3.1 The Karhunen-Loève expansion

As shown in [1], the following theorem holds:

Proposition 1 (Karhunen-Loève expansion)**.** *Let $X(t)$, $t \in [a,b]$, a and b finite, be a continuous-parameter second-order random process with zero mean and continuous covariance function $R(t,s)$. Then we may write*

$$X(t) = \sum_{k=1}^{\infty} Z_k e_k(t), \quad a \leq t \leq b,$$

where e_k are eigenfunctions of the integral operator A

$$A(x(\cdot))(t) = \int_a^b R(t,s)x(s)ds, \quad a \leq t \leq b,$$

i.e. $A(e_k(\cdot))(t) = \lambda_k e_k(t)$. The eigenfunctions form an orthornormal basis for the space spanned by the eigenfunctions corresponding to the non-zero eigenvalues. The Z_k are given by

$$\int_a^b X(t)e_k(t)dt,$$

and are independent, zero mean random variables with variance λ_k. The series $\sum_{k=1}^{\infty} Z_k e_k(t)$ converges in mean square to $X(t)$ uniformly in t.

The result we will use in this essay is the following, also proved in [1]:

Proposition 2. *If $X(t)$ is a Gaussian process, then the Z_k of the Karhunen-Loève expansion of $X(t)$ are independent Gaussian random variables.*

As we know the variance of these Gaussian random variables by the first proposition, we find that we can write

$$X(t) = \sum_{k=1}^{\infty} \sqrt{\lambda_k} g_k e_k(t),$$

where $g_k \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$. Note that this is equivalent to

$$X(t) = \sum_{k=1}^{\infty} \lambda_k g_k e_k(t),$$

by choosing λ_k^2 as the eigenvalues with an abuse of notation. From here on, one should always see λ_k^2 to be the eigenvalues, as is common in the literature (see e.g. [6]). This is because we can also consider

$$X(t) = \sum_{k=1}^{\infty} \lambda_k g_k e_k(t) = \sum_{k=1}^{\infty} \xi_k e_k(t), \tag{3}$$

where the ξ_ks are independent normals, $\xi_k \sim \mathcal{N}(0, \lambda_k^2)$. We will from here on also assume the covariance operator to be trace-class, i.e. the sum of the eigenvalues $\sum_{k=1}^{\infty} \lambda_k^2$ (with the new notation) is finite.
Both notations in (3) will be used for the MCMC methods.

So far the Karhunen-Loève expansion was quite theoretical and the reader might not yet see how it is useful for the MCMC methods on function spaces. That should become clearer in this paragraph. For any MCMC method we need to define a prior and on a function space the first thing that might come to one's mind is a (centered) Gaussian process, defined by the covariance operator $\mathcal{C}$, thus $\mu_0 = \mathcal{N}(0, \mathcal{C})$. If we want to draw a sample $u(t)$ from μ_0, we can now use the Karhunen-Loève theorems to realise that we might as well sample one-dimensional Gaussians, and then sum them according to the Karhunen-Loève expansion. Unfortunately, the expansion is an infinite sum, but as it converges, we can truncate it after d_u terms and hope that the remaining terms don't have a huge impact on $u(t)$.

Some remarks on this are now made and the reader should reconsider them after having read section 4. It is generally hard to decide, where to truncate the series expansion. There are some more advanced techniques than simlpy truncating the series after d_u terms, these techniques are found in [6] and we will only give an overview here:
The first modification from simply truncating the series expansion at a fixed d_u is, that one allows d_u to be a random variable itself, e.g. a Poisson random variable. Conditional on d_u, $u(t)$ is still a Gaussian random variable in the function space. We may also give every basis function an individual 'on/off switch', while still controlling the number of active, i.e. non-zero, terms. In each MCMC step, we first

pick d_u (by using a constant number or by using some random variable) and then pick d_u terms to be active, e.g. by using a Bernoulli random variable, and turning the first term on or off with probability a half (or something else), then the second one, etc., until we have the d_u active terms, which we will then sample.

Summarising this subsection, the Karhunen-Loève expansion allows us to draw samples from function space priors μ_0, given some centered Gaussian process with covariance operator $\mathcal{C}$. We will throughout this paper assume that the eigenfunctions and eigenvalues are known, though other possibilities for sampling from this prior are possible, see [6]. We suggest that the reader may think of the eigenfunctions to be the Fourier basis functions for illustrative purposes, even though the (orthonormal) eigenfunctions are uniquely determined by the covariance operator and will generally not coincide with the Fourier basis functions.

3.2 The Feldman-Hajek theorem

In this subsection, we will state the Feldman-Hajek theorem, which characterises equivalence of Gaussian measures on Hilbert spaces; it is proved in [8], where one also finds a more detailed description and more properties of Gaussian measures on both Banach and Hilbert spaces. First note that we call a measure μ on a Hilbert space a Gaussian measure if, for arbitrary $h \in H$, there exists some $m \in \mathbb{R}$ and $q \geq 0$, such that $\mu(\{x \in H : \langle h, x\rangle \in A\}) = \mathcal{N}(m, q)(A)$. As in the finite-dimensional case, we can uniquely characterise this measure by a mean $m \in H$ and a covariance operator $Q : H \to H$. We may thus write $\mathcal{N}(m, Q)$ for this measure.

The reader should recall that in the Karhunen-Loève expansion we wrote $\mu := \mu_0 = \mathcal{N}(0, \mathcal{C})$ as an infinite sum of orthogonal Gaussians, which defines a unique Gaussian measure on the Hilbert space. If we are given another measure, $\nu = \mathcal{N}(m_\nu, Q_\nu)$, one might be interested in whether these two measures are equivalent, whether one is singular with respect to the other, or whether they are mutually singular, as we know from standard MCMC theory that we need equivalent measures in order to properly define the acceptance probability. The surprising theorem in Hilbert spaces is that μ and ν are either equivalent or mutually singular, and we can furthermore find necessary and sufficient conditions for the equivalence:

Proposition 3 (Feldman-Hajek)**.** *The following statement holds.*
(1) Gaussian measures $\mu = \mathcal{N}(m_1, Q_1)$, $\nu = \mathcal{N}(m_2, Q_2)$ are either mutually singular or equivalent.
(2) They are equivalent if, and only if, the following conditions hold.
(i) $Q_1^{1/2}(H) = Q_2^{1/2}(H) =: H_0$.
(ii) $m_1 - m_2 \in H_0$.
(iii) The operator $(Q_1^{-1/2}Q_2^{1/2})(Q_1^{-1/2}Q_2^{1/2})^* - I$ is a Hilbert-Schmidt operator on $\overline{H_0}$.

The reader is referred to [8] for the proof, but we will give an intuitive explanation for the criteria. Recall that two measures are mutually singular if they have disjoint supports.
Now, the condition (i) tells us that for two measures to be equivalent, the images of their covariance operators have to match. In particular, if one measure

does degenerate in one direction, i.e. if one eigenvalue, the k-th one say, in the Karhunen-Loève expansion of μ is 0, then the other measure also has to degenerate in the same direction, i.e. the variance into the direction of $e_{\mu,k}$ has to equal 0. $e_{\mu,k}$ denotes the k-th eigenfunction of the expansion of μ.
Condition (ii) then says that if both μ and ν degenerate, in the k-th coordinate of μ say, then in that direction their means have to match, which already implies that we may choose $e_{\nu,k} = e_{\mu,k}$. If their means in the k-th direction wouldn't match, the sets $\{x \in H : \langle x, e_{\mu,k}\rangle = m_{1,k}\}$ and $\{x \in H : \langle x, e_{\nu,k}\rangle = m_{2,k}\}$, where $m_{i,k}$ is the k-th coordinate of m_i, are disjoint, thus μ and ν would be mutually singular, as their supports are subsets of the respective sets.
The last condition is the most abstract one, telling us that, as $k \to \infty$, the eigenfunctions of μ and ν have to match and the corresponding eigenvalues, i.e. the variances in the directions of the eigenfunctions, also have to match. This becomes clearer, when noting that condition (iii) is equivalent to

$$\sum_{i,j=1}^{\infty} (r_{ij} - \delta_{ij})^2 < \infty,$$

where $r_{ij} = \frac{\langle Q_2 e_{\mu,i}, e_{\mu,j}\rangle}{\sqrt{\lambda_{\mu,i}\lambda_{\mu,j}}}$ and δ_{ij} the Kronecker-delta. The equivalence is obtained by looking at the spectral decomposition of the operator in condition (iii), which is possible by the spectral theorem for compact operators (note that any Hilbert-Schmidt operator is compact). For $i = j$, the sum can only be finite if Q_2 has eigenvalue $\lambda_{\nu,i} = \lambda_{\mu,i}$ corresponding to the eigenfunction $e_{\nu,i} = e_{\mu,i}$, or at least doesn't differ too much, more precisely, the sum $\sum_{i=1}^{\infty}(r_{ii} - 1)^2$ must be finite. For $i \neq j$, the sum can only converge if $e_{\mu,i}$ and $e_{\mu,j}$ get uncorrelated under Q_2, as $i, j \to \infty$.

Summarising, this subsection told us that two Gaussian measures are either mutually singular or equivalent, and that they are equivalent if, and only if, their eigenvalues and eigenfunctions in the Karhunen-Loève expansion don't differ too much, as specified in the Feldman-Hajek theorem.

4 MCMC methods on function spaces

This section, which discusses methods proposed in [6], starts with an intuitive introduction, by generalising the standard Random Walk Metropolis-Hastings method from finite to infinite dimensions. We realise that this method doesn't work in infinite dimensions (or at least not without major, computational expensive adjustments). The second subsection then introduces another MCMC method on function space that is stable under mesh-refinement, i.e. we can use an arbitrary number of non-zero coefficients in the series representation of the function we want to sample, without having to adjust the method in any way. However, that method has a few downsides as well and adjustments are also discussed.
The theorems 4, 5, and 6, which are stated in this section, are modified versions of theorems found in [6]. The proofs are the authors own work, unless stated otherwise.

4.1 Random Walk Metropolis-Hastings

We assume that the reader is familiar with MCMC methods on finite-dimensional spaces, in particular with the Metropolis-Hastings algorithm. Otherwise [15] gives a brief introduction and [10] offers an exhaustive discussion of the method with many explanatory examples.
We take the standard random walk method in finite dimensions and generalise it to define the standard random walk (SRW) method in infinite dimensions as follows:

Algorithm 1: SRW

Set $k = 0$ and pick $u^{(0)}$;
while *true* **do**
  Propose $v^{(k)} = u^{(k)} + \xi^{(k)}$, $\xi^{(k)} \sim \mathcal{N}(0, I)$.
  Set $u^{(k+1)} = v^{(k)}$ with probability $a(u^{(k)}, v^{(k)})$.
  Set $u^{(k+1)} = u^{(k)}$ otherwise. Set $k \to k + 1$.
end

We ignore the question how the acceptance probability is defined for the moment. One realises quickly that this method will lead to an ill-posed one, as the proposal v will not have a well-defined norm, the sum of the i.i.d. $\mathcal{N}(0, 1)$ random variables will almost surely not converge. This may be adjusted, however, by adding a preconditioner that ensures convergence. Let $\mathcal{C}$ be the covariance operator of the prior, i.e. $u^{(0)} \sim \mathcal{N}(0, \mathcal{C})$, then in each step of the algorithm we propose to add another $\mathcal{N}(0, \mathcal{C})$ random variable to our current position. This random variable has finite norm. We also introduce a tuning parameter β, such that the (preconditioned) SRW method becomes

Algorithm 2: pSRW

Set $k = 0$ and pick $u^{(0)}$;
while *true* **do**
  Propose $v^{(k)} = u^{(k)} + \beta\xi^{(k)}$, $\xi^{(k)} \sim \mathcal{N}(0, \mathcal{C})$.
  Set $u^{(k+1)} = v^{(k)}$ with probability $a(u^{(k)}, v^{(k)})$.
  Set $u^{(k+1)} = u^{(k)}$ otherwise. Set $k \to k + 1$.
end

In this method, one may play around with β, but we will show that for fixed β, this MCMC method is not defined on function space. The reason for this is, as we let the dimension $d_u \to \infty$, the acceptance probability is not defined and thus the method is not independent of the dimension d_u, i.e. the active terms in the Karhunen-Loève expansion. It would be possible to adjust β to the d_u used, but especially when using a variable d_u this would considerably slow down the perfomance.
It is shown in [5] that the acceptance probability is given by $\min\{1, I(u) - I(v)\}$, where $I(u) = \Phi(u) + \frac{1}{2}|\mathcal{C}^{-\frac{1}{2}}u|^2$. Here, and anywhere else in this essay, $\Phi(u)$ is some real-valued potential, which can be interpreted as a function, whose negative exponential is proportional to the Radon-Nikodym derivative of the posterior with respect to the prior whenever it exists: $\frac{\mathrm{d}\mu}{\mathrm{d}\mu_0}(u) \propto \exp(-\Phi(u))$. Note that this is just a reformulation of Bayes' formula $\frac{\mathrm{d}\mu}{\mathrm{d}\mu_0}(u) \propto L(u)$, for fixed data and where L is the likelihood for the data given u.

We will now prove the following theorem, which shows that the (preconditioned) SRW method is not independent of d_u, a property that is also known as 'stable under mesh-refinement' or 'robust'. Note that in this theorem, $\beta = \sqrt{2\delta}$, which is due to it being a special case of the stochastic differential equation (SDE) we discuss in the subsequent subsection[2].

Theorem 4. *Consider the proposal $v|u \sim q(u,\cdot)$ defined by $v = u + \sqrt{2\delta\mathcal{K}}\xi_0$, where $\mathcal{K} \in \{\mathcal{C}, I\}$ and $\xi_0 \sim \mathcal{N}(0, I)$, and the resulting measure $\eta(du, dv) = q(u, dv)\mu(du)$ on $X \times X$. For both choices of $\mathcal{K}$ the measure $\eta^\perp = q(v, du)\mu(dv)$ is not absolutely continuous with respect to η. Thus, the MCMC method is not defined on function space.*

Proof. $u = u_0$ is a draw from $\mathcal{N}(0, \mathcal{C})$. Here, and in the rest of this essay, we will identify u with its Karhunen-Loève expansion $\sum \xi_i e_i$, and refer to $u_i = \xi_i$ as the i-th coordinate. For a truncated u, the acceptance probabilty is given by $\min\{1, I(u) - I(v)\}$, where $I(u) = \Phi(u) + \frac{1}{2}|\mathcal{C}^{-\frac{1}{2}}u|^2$.
The i-th coordinate of u is $\mathcal{N}(0, \lambda_i^2)$ distributed, such that the i-th coordinate of $\mathcal{C}^{-\frac{1}{2}}u$ is $\mathcal{N}(0,1)$ distributed, as the i-th eigenvalue of $\mathcal{C}^{-\frac{1}{2}}$ is λ_i^{-1} and $\lambda_i^{-1}u_i \sim \mathcal{N}(0, \lambda_i^{-2}\lambda_i^2) = \mathcal{N}(0,1)$. Let these i.i.d. standard normal variables be g_i. Let d_u be the largest non-zero coefficient of the truncated series. The expectation for $\frac{1}{2}|\mathcal{C}^{-\frac{1}{2}}u|^2$ then becomes

$$\begin{aligned}
\mathbb{E}\frac{1}{2}\left|\mathcal{C}^{-\frac{1}{2}}u\right|^2 &= \frac{1}{2}\mathbb{E}\sum_{i=1}^{d_u}\left(\frac{1}{\lambda_i}u_i\right)^2 \\
&= \frac{1}{2}\mathbb{E}\sum_{i=1}^{d_u} g_i^2 \\
&= \frac{1}{2}\sum_{i=1}^{d_u} 1.
\end{aligned}$$

For the not-truncated version, i.e. when letting $d_u \to \infty$, this expectation is infinite, such that the acceptance probability is not well-defined. □

It is quite disappointing that the standard random walk method, which works perfectly well in finite dimensions, doesn't generalise to infinite dimensions. Fortunately, a rather small modification leads to a well-defined method. This will be discussed in the next subsection.

4.2 (Preconditioned) Crank-Nicolson

This subsection deals with the preconditioned Crank-Nicolson (pCN) method and its modifications. It is a MCMC method that indeed works on function spaces.

[2]The interested reader may compare the SDE (4) to $\frac{\mathrm{d}u}{\mathrm{d}s} = \sqrt{2\mathcal{K}}\frac{\mathrm{d}b}{\mathrm{d}s}$. The discretisation of this simpler SDE yields the proposal $v = u + \sqrt{2\delta\mathcal{K}}\xi_0$.

4.2.1 Derivation and properties

The method can be derived from discretising a stochastic differential equation (SDE), which works as follows: Consider the SDE

$$\frac{\mathrm{d}u}{\mathrm{d}s} = -\mathcal{K}(\mathcal{L}u + \gamma D\Phi(u)) + \sqrt{2\mathcal{K}}\frac{\mathrm{d}b}{\mathrm{d}s}, \tag{4}$$

where $\mathcal{K} \in \{\mathcal{C}, I\}$, $\mathcal{L} = \mathcal{C}^{-1}$ is the precision operator, and b is a standard Brownian motion. This SDE has the nice property that for both $\gamma = 0$ and $\gamma = 1$ we know that the invariant measures are μ_0 and μ respectively, see [8]. For this essay however, we will only consider the case where $\gamma = 0$, thus the SDE (4) becomes

$$\frac{\mathrm{d}u}{\mathrm{d}s} = -\mathcal{K}\mathcal{L}u + \sqrt{2\mathcal{K}}\frac{\mathrm{d}b}{\mathrm{d}s}. \tag{5}$$

As just stated, this SDE has invariant measure μ_0, so discretising it should lead to a discrete time chain, which also has invariant measure μ_0, which we will then be able to use for a MCMC method on function spaces.
The simplest method for discretising a differential equation are the forward Euler and backward Euler method, see [17, Ch. 12], but an entire family of discretisations is given by a combination of these two. For $\theta \in [0, 1]$ a discretisation of (5) is given by

$$v - u = -\delta\mathcal{K}\mathcal{L}((1-\theta)u + \theta v) + \sqrt{2\mathcal{K}\delta}\xi_0, \tag{6}$$

where u is the current position, v the next position, δ the time difference between those two steps, and ξ_0 is a standard Normal random variable. Note that $\theta = 0$ is the forward Euler method, $\theta = 1$ gives the backward Euler method, and $\theta = \frac{1}{2}$ is the Crank-Nicolson method. Rearranging (6) yields (under the assumption that $I + \delta\theta\mathcal{K}\mathcal{L}$ is invertible)

$$v = (I + \delta\theta\mathcal{K}\mathcal{L})^{-1}((I - \delta(1-\theta)\mathcal{K}\mathcal{L})u + \sqrt{2\delta\mathcal{K}}\xi_0). \tag{7}$$

We now choose $\mathcal{K} = \mathcal{C}$, $\theta = \frac{1}{2}$, and define $\beta = \frac{\sqrt{2\delta}}{1+\delta/2}$. The proposal in (7) becomes

$$v = \sqrt{1-\beta^2}u + \beta\xi,$$

where $\xi \sim \mathcal{N}(0, \mathcal{C})$. This now allows us to formulate the following MCMC method on function space:

Algorithm 3: pCN

Set $k = 0$ and pick $u^{(0)}$;
while *true* **do**
  Propose $v^{(k)} = \sqrt{1-\beta^2}u^{(k)} + \beta\xi^{(k)}$, $\xi^{(k)} \sim \mathcal{N}(0, \mathcal{C})$.
  Set $u^{(k+1)} = v^{(k)}$ with probability $a(u^{(k)}, v^{(k)})$.
  Set $u^{(k+1)} = u^{(k)}$ otherwise. Set $k \to k + 1$.
end

Here, the acceptance probability is given by $a(u, v) = \min\{1, \exp(\Phi(u) - \Phi(v))\}$, by standard MCMC theory and the next theorem. Also note that Φ is still a real-valued potential, and note that $\Phi(u) \propto -\log L(u)$ if posterior and prior are

absolutely continuous with respect to each other, see [6].
It may be asked why we choose $\theta = \frac{1}{2}$ in the pseudo-code, but this choice is not arbitrary. The following theorem explains this choice, stating that only $\theta = \frac{1}{2}$ leads to a well-defined MCMC method on function space. For other choices of θ the acceptance probabilities aren't defined, as the measures η and $\eta^{\perp}$ are mutually singular by the Feldman-Hajek theorem from the preceeding section.

Theorem 5. *Let $\mu_0(X) = 1$, let the real-valued potential Φ satisfy Assumption 6.1(2) in [6] and assume that μ and μ_0 are equivalent as measures with the Radon-Nikodym derivative $\frac{d\mu}{d\mu_0}(u) \propto \exp(-\Phi(u))$. Consider the proposal $v|u \sim q(u, \cdot)$ defned by*

$$v = (I + \delta\theta\mathcal{K}\mathcal{L})^{-1}((I - \delta(1-\theta)\mathcal{K}\mathcal{L})u + \sqrt{2\delta\mathcal{K}}\xi_0) \tag{8}$$

and the resulting measure $\eta(du, dv) = q(u, dv)\mu(du)$ on $X \times X$.

For both $\mathcal{K} = I$ and $\mathcal{K} = \mathcal{C}$ the measure $\eta^{\perp} = q(v, du)\mu(dv)$ is equivalent to η if and only if $\theta = \frac{1}{2}$.

Furthermore, if $\theta = \frac{1}{2}$, then

$$\frac{d\eta^{\perp}}{d\eta}(u, v) = \exp(\Phi(u) - \Phi(v)).$$

Proof. We use the fact that two centered product Gaussian laws $\Pi_{i=1}^{\infty}\mathcal{N}(0, \sigma_i)$ and $\Pi_{i=1}^{\infty}\mathcal{N}(0, \tau_i)$ are equivalent if and only if $\sum_{i=1}^{\infty}\left(\frac{\sigma_i}{\tau_i} - 1\right)^2 < \infty$, see [3, Lemma A.1.].
We know that $u \sim \mathcal{N}(0, \mathcal{C})$, thus the i-th coordinate is distributed $u_i \sim \mathcal{N}(0, \lambda_i^2)$ by the Karhunen-Loéve Expansion. Every coordinate of the noise is a $\mathcal{N}(0, 1)$ random variable. To determine the law of v, we look at each coordinate of v as proposed by (8) by itself and obtain, for all i,

$$v_i = \frac{1 - \delta(1-\theta)[\lambda_i^2]\lambda_i^{-2}}{1 + \delta\theta[\lambda_i^2]\lambda_i^{-2}}u_i + \frac{\sqrt{2\delta[\lambda_i^2]}}{1 + \delta\theta[\lambda_i^2]\lambda_i^{-2}}g_i, \tag{9}$$

where $g_i \sim \mathcal{N}(0, 1)$ and $[\lambda_i^2] = 1$ if $\mathcal{K} = I$ and $[\lambda_i^2] = \lambda_i^2$ if $\mathcal{K} = \mathcal{C}$. Thus

$$\begin{aligned} v_i &\sim \mathcal{N}\left(0, \left(\frac{1 - \delta(1-\theta)[\lambda_i^2]\lambda_i^{-2}}{1 + \delta\theta[\lambda_i^2]\lambda_i^{-2}}\right)^2 \lambda_i^2 + \left(\frac{\sqrt{2\delta[\lambda_i^2]}}{1 + \delta\theta[\lambda_i^2]\lambda_i^{-2}}\right)^2\right) \\ &= \mathcal{N}\left(0, \left(\frac{1 - \delta(1-\theta)[\lambda_i^2]\lambda_i^{-2}}{1 + \delta\theta[\lambda_i^2]\lambda_i^{-2}}\right)^2 \lambda_i^2 + \frac{2\delta[\lambda_i^2]}{(1 + \delta\theta[\lambda_i^2]\lambda_i^{-2})^2}\right) \\ &= \mathcal{N}\left(0, \left[\left(\frac{1 - \delta(1-\theta)[\lambda_i^2]\lambda_i^{-2}}{1 + \delta\theta[\lambda_i^2]\lambda_i^{-2}}\right)^2 + \frac{2\delta[\lambda_i^2]}{(1 + \delta\theta[\lambda_i^2]\lambda_i^{-2})^2\lambda_i^2}\right]\lambda_i^2\right). \end{aligned}$$

We devide the variance of v_i by the variance of u_i (i.e. λ_i^2) as suggested by condition (iii) in theorem 3 and obtain

$$a_i := \left(\frac{1-\delta(1-\theta)[\lambda_i^2]\lambda_i^{-2}}{1+\delta\theta[\lambda_i^2]\lambda_i^{-2}}\right)^2 + \frac{2\delta[\lambda_i^2]}{(1+\delta\theta[\lambda_i^2]\lambda_i^{-2})^2\lambda_i^2},$$

this simplifies to either

$$\left(\frac{1-\delta(1-\theta)}{1+\delta\theta}\right)^2 + \frac{2\delta}{(1+\delta\theta)^2} = \frac{1+2\delta\theta+\delta^2(1-\theta)^2}{1+2\delta\theta+\delta^2\theta^2} \tag{10}$$

if $\mathcal{K}=\mathcal{C}$ or to

$$\left(\frac{1-\delta(1-\theta)\lambda_i^{-2}}{1+\delta\theta\lambda_i^{-2}}\right)^2 + \frac{2\delta}{(1+\delta\theta\lambda_i^{-2})^2\lambda_i^2} = \left(\frac{\lambda_i^2-\delta(1-\theta)}{\lambda_i^2+\delta\theta}\right)^2 + \frac{2\delta\lambda^2}{(\lambda_i^2+\delta\theta)^2}$$

if $\mathcal{K}=I$ which converges to

$$\frac{\delta^2(1-\theta)^2}{\delta^2\theta^2} = \frac{(1-\theta)^2}{\theta^2} \tag{11}$$

as $i\to\infty$, remembering that $\lambda_i\to 0$, by the definition of $\mathcal{C}$. Now to use the proposition stated at the beginning of the proof, we need θ such that (10) and (11) are 1 respectively, as then $\sum_i (Var(v_i)/Var(u_i)-1)^2$ is finite. In both cases, it is obvious that only $\theta=\frac{1}{2}$ satisfies this condition, showing that the product measures are equivalent if and only if $\theta=\frac{1}{2}$.
For the last claim, see [16]. One uses the assumptions on Φ to show that the Radon-Nikodym derivative is well-defined. □

In order for pCN to be a well-working method, we need to be able to choose δ in a way that we can tune the method. A first step for this is to show that, as $\delta\to 0$, the average acceptance probability (also known as the acceptance ratio) converges to 1. This is shown by the next theorem:

Theorem 6. *Let μ_0 be a Gaussian measure on a Hilbert space $(X,\|\cdot\|)$ with $\mu_0(X)=1$ and let μ be an equivalent measure on X given by the Radon-Nikodym derivative $\frac{d\mu}{d\mu_0}(u)\propto\exp(-\Phi(u))$, satisfying Assumptions 6.1(1) and 6.1(2) in [6]. Then both the pCN and CN algorithms (using the proposal (8) with $\mathcal{K}=\mathcal{C}$ and $\mathcal{K}=I$, respectively) with fixed δ are defined on X and, furthermore, the acceptance probability satisfies*

$$\lim_{\delta\to 0}\mathbb{E}^{\eta}a(u,v)=1.$$

Proof. For fixed δ one checks in the preceeding theorem that the measures η and $\eta^{\perp}$ are absolutely continuous with respect to each other, and using the acceptance probability $a(u,v)=\min\{1,\exp(\Phi(u)-\Phi(v))\}$ one checks in [16] that one obtains a well-defined algorithm.

For the final claim, one looks at the change in each coordinate as in the proof of theorem 5, see (9).

We first look at the slightly simpler case $\mathcal{K} = \mathcal{C}$, where equation (9) simplifies to

$$v_i = \frac{1-\delta\frac{1}{2}}{1+\delta\frac{1}{2}}u_i + \frac{\sqrt{2\delta\lambda_i^2}}{1+\delta\frac{1}{2}}g_i = \frac{2-\delta}{2+\delta}u_i + \frac{\sqrt{8\delta\lambda_i^2}}{2+\delta}g_i.$$

The second term is a noise with distribution $\mathcal{N}(0, \frac{8\delta\lambda_i^2}{(2+\delta)^2})$ which converges to a degenerated normal, i.e. the noise term converges to 0. For the other part we now consider the sum of the proposals v_i

$$v = \sum_{i=1}^{\infty} v_i = \sum_{i=1}^{\infty} \frac{2-\delta}{2+\delta}u_i = \frac{2-\delta}{2+\delta}\sum_{i=1}^{\infty} u_i = \frac{2-\delta}{2+\delta}u,$$

which converges to u as $\delta \to 0$. Now using the Assumption 6.1(2), we notice that

$$|\Phi(u) - \Phi(v)| \leq K(r)\|u - v\| \overset{\delta\to 0}{\longrightarrow} 0,$$

and therefore $\Phi(u) - \Phi(v) \to 0$, such that $\exp(\Phi(u) - \Phi(v)) \to 1$ by continuity of the exponential function, which then implies $a(u, v) \to 1$. Now looking at the expectation one sees

$$\lim_{\delta\to 0} \mathbb{E}^{\eta} a(u, v) = \lim_{\delta\to 0} \int_{X\times X} a(u, v)q(u, dv)\mu(du) = \int_{X\times X} q(u, dv)\mu(du) = 1$$

by the dominated convergence theorem, as $a(u, v)$ is bounded by the integrable function $f \equiv 1$, and $a(u, v) \to 1$.

Now we look at the unconditioned case $\mathcal{K} = I$. Here, the proposal (9) for the i-th coordinate of v simplifies to

$$v_i = \frac{1-\delta\frac{1}{2}\lambda_i^{-2}}{1+\delta\frac{1}{2}\lambda_i^{-2}}u_i + \frac{\sqrt{2\delta}}{1+\delta\frac{1}{2}\lambda_i^{-2}}g_i = \frac{2-\delta\lambda_i^{-2}}{2+\delta\lambda_i^{-2}}u_i + \frac{\sqrt{2\delta}}{1+\delta\frac{1}{2}\lambda_i^{-2}}g_i,$$

where the second term is smaller than $\sqrt{2\delta}$ for i large enough, so smaller than $c\sqrt{2\delta}$ for an appropriate constant c. Then, as $\delta \to 0$, this term also degenerates as in the preconditioned case. For the other term, note that we're only interested in showing that, for any $\varepsilon > 0$, there exists some $\tilde{\delta}$ such that $\|u - v\| < \varepsilon$ whenever $\delta < \tilde{\delta}$. Since $\sum_{i=N}^{\infty} u_i \to 0$ as $N \to \infty$, we choose N large enough, such that the remaining series is smaller than $\varepsilon/2$. The first N terms however can be bounded by $\varepsilon/2$ as well, simply be choosing δ small enough, which is possible, as for each term individually, $v_i \to u_i$ as $\delta \to 0$. Thus we again obtain $\|u - v\| \to 0$ and we conclude as in the first case. □

We have now established that the pCN method is well-defined on function spaces. Now, before we discuss some downsides of the method and possible improvements, let us take a step back and summarise in simple words, why the pCN method works. Starting the method with $u \sim \mathcal{N}(0, \mathcal{C})$ and picking $\xi \sim \mathcal{N}(0, \mathcal{C})$ independent from u, we have that $v = \sqrt{1-\beta^2}u + \beta\xi$ is distributed $\mathcal{N}(0, (1-\beta^2)\mathcal{C} + \beta^2\mathcal{C}) = \mathcal{N}(0, \mathcal{C})$. Thus, in every step the distribution of the prior does indeed not change. What preceeded has shown that other choices of θ in the discretisation of the SDE (5) do not define a well-posed MCMC method on function space and the Feldman-Hajek theorem was used to proof this.

4.2.2 Possible modifications

We now turn to answer the following questions:
- Are there any simple modifications to decorrelate the samples obtained faster, to reduce the effective sample size[3] needed?
- If information is given about which coefficients in the Karhunen-Loève expansion are more important, can we formulate a method that considers this to improve the performance? How can we do this and where could we get this information from?

To answer these questions, we outline some of the ideas presented in [13] by K. J. H. Law, in [7] by T. Cui et al., and in [2] by A. Beskos et al., of which the last one has been published in 2017, and is (as for today and as far as the author is informed) the most recent development for MCMC methods on function spaces. The authors mentioned above propose a bunch of methods that modify the pCN method by using additional information, using operator-weighted proposals in the first, likelihood informed proposals in the second, and geometry information in the third paper, which in a way is a generalisation of the first two. In order to understand these ideas, it may be useful to be familiar with Hamiltonian Monte Carlo methods on Riemann manifolds, see [11], as the underlying principles of those methods are the same as the ones presented here.

To understand what is going on in these papers, we first observe that the standard pCN method introduced earlier in this essay is prior-biased: Only the prior determines, which parameters (i.e. coefficients in the Karhunen-Loève expansion) are 'more important' than others, larger steps will be made in directions where the eigenvalues of the covariance matrix are large. However, it might be that the likelihood suggests to put more importance on other parameters. This would ask for a method that then takes larger steps in those parameters to allow for a faster exploration of the parameter space and thus better mixing times.
The reasoning behind this is straight-forward: By Bayes rule, the posterior is proportional to the likelihood times the prior, and therefore both should be considered when proposing good MCMC methods. In [13], operator-weighted proposals are introduced. Instead of having a fixed β in the pCN proposal[4] $v = \sqrt{1-\beta^2}u_n + \beta\xi$, it is shown that one may instead use an operator[5]:

$$v = B_n u_n + \sqrt{I - B_n^2}\xi.$$

This operator may change with every time step, if $B_n = B$ is independent of n we call B a preconditioner. B_n allows us to put different weights on different directions, which is certainly a nice thing for itself, but only the next developments made this idea incredibly useful.

In [7], the authors propose following idea: One divides the space X into two subspaces, the finite-dimensional *likelihood-informed subspace* (LIS) and its complementing space (CS), which exists by basic Hilbert space theory. On the CS, one

[3] It is assumed that the reader is familiar with the notion of the effective sample size, a problem common to all MCMC methods. A first introduction can be found in [15].

[4] Note that this proposal is equivalent to $v = \beta u_n + \sqrt{1-\beta^2}$.

[5] Law actually uses B_n^2 instead of our B_n.

will still use the pCN method introduced above, on the LIS better methods exist. The name comes from the fact that one can identify certain parameters, in which the likelihood dominates the prior and the posterior is strongly determined by the likelihood, thus the name LIS. As stated earlier, the log-likelihood is proportional to the potential Φ if posterior and prior are absolutely continuous with respect to each other: $-\log(L(u)) \propto \Phi(u)$. Thus, the Hessian of $\Phi(u)$ contains information about the likelihood and indeed Cui et al. show that the LIS can be derived from the dominant eigenvectors of the Hessian of a finite-dimensional approximation of Φ. Note that the LIS is a local space, as it normally depends on u (as the Hessian of Φ does). However, one can globalise the local LIS by combining information from many points in the posterior. If n is the total number of observations, the global LIS is shown to be not more than n-dimensional, and often is even lower dimensional. [7] then proposes methods that are equivalent to the pCN on the CS and are operator-weighted on the LIS, using gradient information of Φ and weighting parameters according to their importance given by the likelihood.

The authors of [11] improve and generalise this idea even further. They also take the standard pCN method and modify it by taking advantage of gradient information of the potential Φ, which is called ∞-MALA (infinite-dimensional Metropolis-adjusted Langevin algorithm, by the underlying Langevin SDE). Similar to finite-dimensional Hamiltonian Monte Carlo, they then take both the pCN and ∞-MALA methods, but modify it as follows using Hamiltonian dynamics:
Firstly, a single 'leapfrog step' is a single step by a forward Euler scheme, used to solve the Hamiltonian differential equation with mass matrix equal to the inverse of the preconditioner $\mathcal{K}$ in the pCN method,

$$\frac{\mathrm{d}^2 u}{\mathrm{d}t^2} + \mathcal{K}(\mathcal{C}^{-1}u + D\Phi(u)) = 0.$$

These dynamics preserve the posterior μ for any integration time. As in the finite-dimensional case, it allows to propose quite large steps by executing a fixed number of leapfrog steps. The combination of using gradient information depending on the current position, and performing numerous leapfrog steps to obtain one step in the Metropolis-Hastings chain leads to the most advanced MCMC method discussed in this essay. Girolami et al. test this method and show that it outperforms any other method, and is especially good for target distributions with complex and non-Gaussian structures.

5 Computational experiments

The computational experiments conducted are solely to understand how both the pCN and SRW methods work and perform on (infinite) dimensional spaces, and why the pCN is robust under mesh-refinement while the SRW method isn't. We will discuss the experiments in detail and compare the findings to the theoretical performance discussed above.

The general setting for the experiments was the same as in section 2.1. We aim to retrieve the 'true' distribution which is proportional to $\mathcal{N}(-3,1)\mathbb{1}_{[-10,10]} + \mathcal{N}(3,1)\mathbb{1}_{[-10,10]}$. We are given *dimy* samples, in our program we usually chose

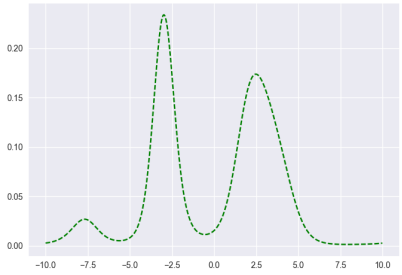
$dimy = 30$ unless stated otherwise.

The reader should note that we picked the Fourier basis functions instead of the Karhunen-Loève expansion, however it is shown in [14] that a Karhunen-Loève expansion for standard Brownian motion is given by $g_0 t + \sum_{k=1}^{\infty} \frac{\sqrt{2}}{\pi k} \sin(\pi k t) g_k$, where g_k are i.i.d. standard Gaussians. This can be rewritten as $g_0 t + \sum_{k=1}^{\infty} \frac{\sqrt{2}}{\pi} \sin(\pi k t) \xi_k$, where ξ_k is a $\mathcal{N}(0, 1/k^2)$ Gaussian random variable. Thus, the Karhunen-Loève expansion coincides with the sin-terms in the Fourier series. The author decided to also include the cos-terms of the Fourier expansion, hoping that the corresponding coefficients would turn out to be close to 0. This conjecture indeed holds, all the experiments have returned the parameters corresponding to the cos-terms found to be relatively close to 0, while the parameters corresponding to the sin-terms dominated the series.

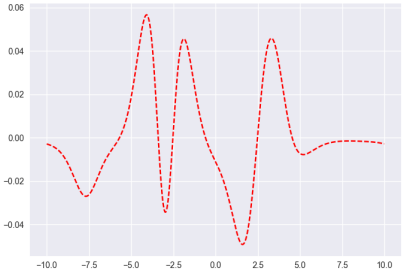
The first experiment showed how the acceptance ratio changed when refining the mesh. As the theory suggests, for fixed β the acceptance ratio for the SRW method tends towards 0, while the acceptance ratio for the pCN method stays more or less constant. Both methods have first been tuned to have acceptance ratios of approximately 0.5 with 11 active terms in the series expansion. For the SRW method, β was chosen to be 0.25, and for the pCN method $\beta = 0.2$ proved to give the desired acceptance ratio. Note that the acceptance ratio for our experiments is defined as the number of accepted steps divided by the total number of steps, where we only considered the last $N/2$ steps, ignoring the burn-in phase. N is the total number of steps executed, and $N = 200$ was chosen for this experiment. To get better results, the experiments can be re-run with more steps, but 200 turned out to be sufficient for these illustrative purposes. For the SRW method, the acceptance ratios were 0.48, 0.33, 0.18, and 0.08 for 11, 31, 51, and 71 active terms respectively. This clearly shows that, under mesh-refinement and for fixed β, the acceptance ratio decreases. Also, it happened that an error occured when the number of active terms was chosen to be too large: when calculating the acceptance probabilties, the function $\exp(I(u) - I(v))$ returned infinity, as the computer couldn't handle the exponent. As proposed in theorem 4, $I(u) - I(v)$ gets larger with increasing dimension $dimu$, and eventually the computer is no longer able to calculate the exponential. This happened when $I(u) - I(v)$ was around 300 or greater. For the pCN method, the respective acceptance probabilities were 0.52, 0.35, 0.44, and 0.38. While this is certainly not constant, neither does it exhibit exhibits the decreasing tendency that the SRW method shows.

Another effect mentioned earlier was shown as well: it doesn't make sense to pick $dimu > dimy$. This will only lead to overfitting and - as mentioned in section 2.1 - one doesn't gain any information.

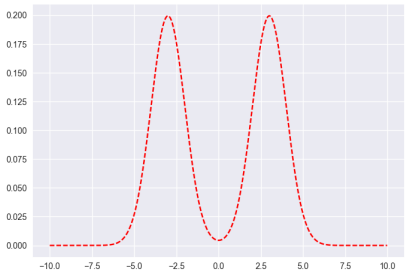
Lastly, some nice pictures:
For the pCN estimate in figure (1), we used $dimu = 11$, $dimy = 30$, $N = 200$, and then took the average over all samples calculated by the pCN method. For the pCN estimate in figure (2), we used $dimu = 21$, $dimy = 100$, and $N = 400$.



(a) Mean density calculated using pCN

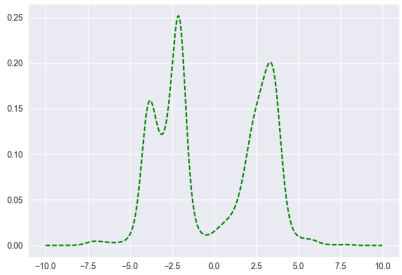


(b) Mean density minus the true denisty

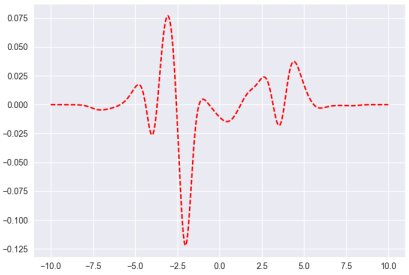


(c) True denisty

Figure 1: Density estimation by pCN



(a) Mean density calculated using pCN



(b) Mean density minus the true denisty

Figure 2: Density estimation by pCN with more non-zero terms

6 Summary and outlook

Summarising, this essay has discussed MCMC methods that work on function spaces, mainly elaborating on the pCN method. It has been shown that the SRW method isn't robust under mesh-refinement. A few applications of MCMC methods on functions spaces have been mentioned, and those methods are considered especially useful for inverse problems. Modifications of the pCN method have been discussed. In the future, the author expects something like an infinite-dimensional version of the No-U-Turn-Sampler proposed in [12]. This suggests a method (in finite dimensions) that automatically adjusts the tuning parameter, β in our algorithm, and at the same time overcomes the problem of the Hamiltonian Monte Carlo method of deciding how many leapfrog steps should be executed.

Acknowledgements

I would like to thank Dr Sumeetpal Singh for supervising this essay, and my parents, my sister, and my brother for constant support. Furthermore, thanks to Chris, Miri, Oli, Rylan, Shaun, and Tine for many fruitful discussions.

Appendix - Code

This is the code used for the pCN experiments above:

```
import matplotlib.pyplot as plt
import numpy as np
import math
import scipy.integrate as integrate
import seaborn as sns

'''Sample draws'''
def draw_sample():
    global dimy
    sample = []
    while len(sample)<dimy:
        p = np.random.uniform()
        if p<0.5:
            new_sample = np.random.normal(-3,1)
        else:
            new_sample = np.random.normal(3,1)
        if new_sample>-10 and new_sample<10:
            sample.append(new_sample)
    return sample

'''Define function for integration'''
def f(x,xi):
    f = u(xi,x)
    return np.exp(f)

'''Define true density function for integration'''
def f_true(x):
    y1=x-3
    y2=x+3
    f = np.exp(-((y1*y1)/2))+np.exp(-((y2*y2)/2))
    return f

'''Define function Phi'''
def Phi(xi):
    global y
    global dimy
    phiu = 0
    for j in range(dimy):
        rho = np.exp(u(xi,y[j]))
        integral_u = integrate.quad(lambda x: f(x,xi), -10,10)
        rho = rho/integral_u[0]
        phiu = phiu+np.log(rho)
    return -phiu

'''Define acceptance probability'''
```

```
def acceptance_prop(xi_u, xi_v):
    accept_prop = math.exp(Phi(xi_u)-Phi(xi_v))
    return min(1, accept_prop)

'''Define u(xi,x)'''
def u(xi,x):
    global dimu
    u = 0
    for i in range(dimu):
        if i%2 == 0:
            u = u+xi[i]*np.cos(2*np.pi*(i/2)*(x-10)/(10*2))
        else:
            u = u+xi[i]*np.sin(2*np.pi*((i+1)/2)*(x-10)/(10*2))
    return u

'''MCMC'''
N = 200
dimu = 11  #accuracy of u
dimy = 30  #number of samples given
xi = np.zeros(dimu)
xi[0] = 1

y = draw_sample()


beta = 1/5
mu = []
mu.append(1)
for i in range(int((len(xi)-1)/2)):
    t = 1/(i+1)
    mu.append(t)
    mu.append(t)
C = np.diag(mu)

samples = []
average = np.zeros(dimu)
acc_ratio = 0
x = []
for i in range(N):
    xi_proposal = np.sqrt(1-beta*beta)*xi+beta*np.random.multivariate_normal(np
    a = acceptance_prop(xi,xi_proposal)
    if a < 1:
        uni = np.random.uniform()
        if uni < a:
            xi = xi_proposal
            if i > N/2:
                samples.append(xi)
                x.append(len(samples))
```

```
                average = average + xi
                acc_ratio = acc_ratio + 1
    else:
        xi = xi_proposal
        if i > N/2:
            samples.append(xi)
            x.append(len(samples))
            average = average + xi
            acc_ratio = acc_ratio+1

samples = np.array(samples)
average = average/len(x)
acc_ratio = acc_ratio/(N/2)

fig1 = plt.figure()
ax1 = fig1.add_subplot(111)
ax1.scatter(x, samples[:, 0], alpha=0.5, s=1)

fig2 = plt.figure()
ax1 = fig2.add_subplot(111)
sns.distplot(samples[:, 0])

fig3 = plt.figure()
ax1 = fig3.add_subplot(111)
ax1.scatter(x, samples[:, 1], alpha=0.5, s=1)

fig4 = plt.figure()
ax1 = fig4.add_subplot(111)
sns.distplot(samples[:, 1])

fig5 = plt.figure()
ax1 = fig5.add_subplot(111)
ax1.scatter(x, samples[:, 8], alpha=0.5, s=1)

fig6 = plt.figure()
ax1 = fig6.add_subplot(111)
sns.distplot(samples[:, 8])

t2 = np.arange(-10,10, 0.02)
fig10 = plt.figure()
ax1 = fig10.add_subplot(111)
integral_avg = integrate.quad(lambda x: f(x,average), -10,10)
plt.plot(t2, f(t2,average)/integral_avg[0], 'g--')

fig11 = plt.figure()
ax1 = fig11.add_subplot(111)
integral_true = integrate.quad(lambda x: f_true(x), -10,10)
plt.plot(t2, f_true(t2)/integral_true[0]-f(t2,average)/integral_avg[0], 'r--')
```

```
plt.show

print (acc_ratio)
print ('Worked.')
```

This is the code used for the SRW experiments:

```
import matplotlib.pyplot as plt
import numpy as np
import math
import scipy.integrate as integrate
import seaborn as sns

'''Sample draws'''
def draw_sample():
    global dimy
    sample = []
    while len(sample)<dimy:
        p = np.random.uniform()
        if p<0.5:
            new_sample = np.random.normal(-3,1)
        else:
            new_sample = np.random.normal(3,1)
        if new_sample>-10 and new_sample<10:
            sample.append(new_sample)
    return sample

'''Define function for integration'''
def f(x,xi):
    f = u(xi,x)
    return np.exp(f)

'''Define function Phi'''
def Phi(xi):
    global y
    global dimy
    phiu = 0
    for j in range(dimy):
        rho = np.exp(u(xi,y[j]))
        integral_u = integrate.quad(lambda x: f(x,xi), -10,10)
        rho = rho/integral_u[0]
        phiu = phiu+np.log(rho)
    return -phiu

'''Define function C(u)'''
def Cop(xi):
    C=0
    for i in range(dimu):
        C=C+(i+1)*xi[i]*xi[i]
```

```
    return C

'''Define function I'''
def I(xi):
    return Phi(xi)+1/2*Cop(xi)

'''Define acceptance probability'''
def acceptance_prop(xi_u, xi_v):
    accept_prop = math.exp(I(xi_u)-I(xi_v))
    return min(1, accept_prop)

'''Define u(xi,x)'''
def u(xi,x):
    global dimu
    u = 0
    for i in range(dimu):
        if i%2 == 0:
            u = u+xi[i]*np.cos(2*np.pi*(i/2)*(x-10)/(10*2))
        else:
            u = u+xi[i]*np.sin(2*np.pi*((i+1)/2)*(x-10)/(10*2))
    return u


'''MCMC'''
N = 200
dimu = 11  #accuracy of u
dimy = 30  #number of samples given
xi = np.zeros(dimu)
xi[0] = 1
y = draw_sample()


beta = 1/4
mu = []
for i in range(len(xi)):
    t = 1/(i+1)
    mu.append(t)
C = np.diag(mu)

samples = []
average = np.zeros(dimu)
acc_ratio = 0
x = []
for i in range(N):
    xi_proposal = xi+beta*np.random.multivariate_normal(np.zeros(dimu),C)
    a = acceptance_prop(xi,xi_proposal)
    if a < 1:
        uni = np.random.uniform()
```

```
        if uni < a:
            xi = xi_proposal
            if i > N/2:
                samples.append(xi)
                x.append(len(samples))
                average = average + xi
                acc_ratio = acc_ratio+1
    else:
        xi = xi_proposal
        if i > N/2:
            samples.append(xi)
            x.append(len(samples))
            average = average + xi
            acc_ratio = acc_ratio+1

samples = np.array(samples)
average = average/len(x)
acc_ratio = acc_ratio/(N/2)

fig1 = plt.figure()
ax1 = fig1.add_subplot(111)
ax1.scatter(x, samples[:, 0], alpha=0.5, s=1)

fig2 = plt.figure()
ax1 = fig2.add_subplot(111)
sns.distplot(samples[:, 0])

fig3 = plt.figure()
ax1 = fig3.add_subplot(111)
ax1.scatter(x, samples[:, 1], alpha=0.5, s=1)

fig4 = plt.figure()
ax1 = fig4.add_subplot(111)
sns.distplot(samples[:, 1])

fig5 = plt.figure()
ax1 = fig5.add_subplot(111)
ax1.scatter(x, samples[:, 8], alpha=0.5, s=1)

fig6 = plt.figure()
ax1 = fig6.add_subplot(111)
sns.distplot(samples[:, 8])

fig7 = plt.figure()
ax1 = fig7.add_subplot(111)
ax1.scatter(x, samples[:, 9], alpha=0.5, s=1)

fig8 = plt.figure()
```

```
ax1 = fig8.add_subplot(111)
sns.distplot(samples[:, 9])

t2 = np.arange(-10,10, 0.02)
fig10 = plt.figure()
ax1 = fig10.add_subplot(111)
integral_avg = integrate.quad(lambda x: f(x,average), -10,10)
plt.plot(t2, f(t2,average)/integral_avg[0], 'r--')
plt.show

print (acc_ratio)
print ('Worked.')
```

References

[1] RB Ash. Information theory. 1965. *Interscience, New York*.

[2] Alexandros Beskos, Mark Girolami, Shiwei Lan, Patrick E Farrell, and Andrew M Stuart. Geometric mcmc for infinite-dimensional inverse problems. *Journal of Computational Physics*, 335:327–351, 2017.

[3] Alexandros Beskos, Gareth Roberts, Andrew Stuart, and Jochen Voss. Mcmc methods for diffusion bridges. *Stochastics and Dynamics*, 8(03):319–350, 2008.

[4] Timothy Cannings. Topics in statistical theory (lecture notes), *Part III lecture, University of Cambridge*, 2016.

[5] Simon L Cotter. *Applications of MCMC methods on function spaces*. PhD thesis, University of Warwick, 2010.

[6] Simon L Cotter, Gareth O Roberts, Andrew M Stuart, David White, et al. Mcmc methods for functions: modifying old algorithms to make them faster. *Statistical Science*, 28(3):424–446, 2013.

[7] Tiangang Cui, Kody JH Law, and Youssef M Marzouk. Dimension-independent likelihood-informed mcmc. *Journal of Computational Physics*, 304:109–137, 2016.

[8] Giuseppe Da Prato and Jerzy Zabczyk. *Stochastic equations in infinite dimensions*. Cambridge university press, 2014.

[9] Phil PG Dyke and PP Dyke. *An introduction to Laplace transforms and Fourier series*. Springer, 2001.

[10] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

[11] Mark Girolami and Ben Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

[12] Matthew D Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[13] Kody JH Law. Proposals which speed up function-space mcmc. *Journal of Computational and Applied Mathematics*, 262:127–138, 2014.

[14] Richard Nickl. Gaussian processes (lecture notes), *Part III lecture, University of Cambridge*, 2016.

[15] Christian P Robert and George Casella. The metropolis—hastings algorithm. In *Monte Carlo Statistical Methods*, pages 231–283. Springer New York, 1999.

[16] Andrew M Stuart. Inverse problems: a bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

[17] Endre Süli and David F Mayers. *An introduction to numerical analysis*. Cambridge university press, 2003.