Journal of Computational Physics 332 (2017) 492–503

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

On an adaptive preconditioned Crank–Nicolson MCMC algorithm for infinite dimensional Bayesian inference

Zixi Hu [a], Zhewei Yao [a], Jinglai Li [b,*]

[a] Department of Mathematics, Zhiyuan College, Shanghai Jiao Tong University, Shanghai 200240, China
[b] Institute of Natural Sciences, Department of Mathematics, and MOE Key Laboratory of Scientific and Engineering Computing, Shanghai Jiao Tong University, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 1 April 2016
Received in revised form 5 October 2016
Accepted 15 November 2016
Available online 7 December 2016

Keywords:
Bayesian inference
Infinite dimensional inverse problems
Adaptive Markov Chain Monte Carlo

ABSTRACT

Many scientific and engineering problems require to perform Bayesian inference for unknowns of infinite dimension. In such problems, many standard Markov Chain Monte Carlo (MCMC) algorithms become arbitrary slow under the mesh refinement, which is referred to as being dimension dependent. To this end, a family of dimensional independent MCMC algorithms, known as the preconditioned Crank–Nicolson (pCN) methods, were proposed to sample the infinite dimensional parameters. In this work we develop an adaptive version of the pCN algorithm, where the covariance operator of the proposal distribution is adjusted based on sampling history to improve the simulation efficiency. We show that the proposed algorithm satisfies an important ergodicity condition under some mild assumptions. Finally we provide numerical examples to demonstrate the performance of the proposed method.

1. Introduction

In many real-world inverse problems, the unknowns that one wants to estimate are functions of space and/or time. Solving such problems with the Bayesian approaches [11,23], often requires to perform Markov Chain Monte Carlo (MCMC) simulations in function spaces. Namely one first represents the unknown function with a finite-dimensional parametrization, for example, by discretizing the function on a pre-determined mesh grid, and then performs MCMC simulations in the resulting finite dimensional space. It has been known that standard MCMC algorithms, such as the random walk Metropolis–Hastings (RWMH), can become arbitrarily slow as the discretization mesh of the unknown is refined [19,21,3,15]. That is, the mixing time of an algorithm can increase to infinity as the dimension of the discretized parameter approaches to infinity, and in this case the algorithm is said to be *dimension-dependent*. To this end, a very interesting line of research is to develop *dimension-independent* MCMC algorithms by requiring the algorithms to be well-defined in the function spaces. In particular, a family of dimension-independent MCMC algorithms, known as the preconditioned Crank Nicolson (pCN) algorithms, were presented in [6] by constructing a Crank–Nicolson discretization of a stochastic partial differential equation (SPDE) that preserves the reference measure.

The sampling efficiency of the pCN algorithm can be improved by incorporating the data information in the proposal design, and a popular way to achieve this goal is the adaptive MCMC methods. Simply speaking, the adaptive MCMC algorithms improve the proposal based on the sampling history from the targeting distribution (cf. [1,2,20] and the references

* Corresponding author.
E-mail addresses: hzx@sjtu.edu.cn (Z. Hu), zyaosjtu@gmail.com (Z. Yao), jinglaili@sjtu.edu.cn (J. Li).

http://dx.doi.org/10.1016/j.jcp.2016.11.024
0021-9991/

therein) as the iterations proceed. A major advantage of the adaptive methods is that they only require the ability to evaluate the likelihood functions, which makes them particularly convenient for problems with black-box models. In a recent work [9], we develop an adaptive independence sampler MCMC algorithm for the infinite dimensional problems. A main difficulty of independence sampler MCMC algorithms is that the efficiency of such algorithms depends critically on the ability of the chosen proposal, often in a parametrized form, to approximate the posterior in the entire state space, and the algorithm may perform very poorly if the proposal can not well approximate the posterior distribution. In this respect, random walk based algorithms may be more convenient to use, as they do not require such a "global proposal". In this work, we present an adaptive random walk MCMC based on the preconditioned Crank–Nicolson (pCN) algorithm in [6]. Specifically, we adaptively adjust the covariance operator of the proposal to improve the sampling efficiency. We parametrize the covariance operator in a specific form that has been used in [17,9], and we provide an algorithm that can efficiently update the parameter values as the iteration proceeds. By design, the acceptance probability of our algorithm is well defined and thus the algorithm is dimension independent. Moreover, we can show that the algorithm satisfies some important ergodicity conditions in the infinite dimensional setting. Note that, another existing adaptive MCMC algorithm for infinite dimensional problems is the dimension independent adaptive Metropolis (DIAM) proposed in [5]. The DIAM is also based on the pCN algorithm, but our method preserves an important feature of the standard pCN algorithm, i.e., the acceptance probability being independent on the proposal distribution, while the DIAM method does not.

We note that, an alternative class of methods improve the sampling efficiency by guiding the proposal with the local derivative information of the likelihood function. Such derivative based methods include: the stochastic Newton MCMC [14,16], the Riemann manifold Hamiltonian MC [4], the operator-weighted proposal method [13], the dimension-independent likelihood-informed MCMC [7], the generalized pCN algorithm [22], and so on. We reinstate that in this work we are focused on the type of problems where the derivative information is difficult to obtain, and thus those derivative based methods are not in our scope.

The rest of the paper is organized as the following. In section 2 we describe the setup of infinite dimensional inference problems and present our adaptive algorithm in detail. In section 3 we provide several numerical examples to demonstrate the performance of the proposed algorithm. Finally we offer some concluding remarks in section 4.

2. The adaptive pCN algorithm

2.1. Bayesian inferences in function spaces

We present the standard setup of the Bayesian inverse problem following [23]. We consider a separable Hilbert space X with inner product $\langle\cdot,\cdot\rangle_X$. Our goal is to estimate the unknown $u \in X$ from data $y \in Y$ where Y is the data space and y is related to u via a likelihood function $\exp(-\Phi^y(u))$. In the Bayesian inference we assume that the prior μ_0 of u, is a (without loss of generality) zero-mean Gaussian measure defined on X with covariance operator $\mathcal{C}_0$, i.e. $\mu_0 = N(0, \mathcal{C}_0)$. Note that $\mathcal{C}_0$ is symmetric positive and of trace class. The range of $\mathcal{C}_0^{\frac{1}{2}}$,

$$E = \{u = \mathcal{C}_0^{\frac{1}{2}} x \,|\, x \in X\} \subset X,$$

which is a Hilbert space equipped with inner product [8],

$$\langle\cdot,\cdot\rangle_E = \langle \mathcal{C}_0^{-\frac{1}{2}}\cdot, \mathcal{C}_0^{-\frac{1}{2}}\cdot\rangle_X,$$

is called the Cameron–Martin space of measure μ_0. In this setting, the posterior measure μ^y of u conditional on data y is provided by the Radon–Nikodym derivative:

$$\frac{d\mu^y}{d\mu_0}(u) = \frac{1}{Z}\exp(-\Phi^y(u)), \tag{2.1}$$

with Z being a normalization constant, which can be interpreted as the Bayes' rule in the infinite dimensional setting. In what follows, without causing any ambiguity, we shall drop the superscript y in Φ^y and μ^y for simplicity, while keeping in mind that these items depend on the data y. For the inference problem to be well-posed, one typically requires the functional Φ to satisfy the Assumptions (6.1) in [6]. It is known that there exists a complete orthonormal basis $\{e_j\}_{j\in\mathbb{N}}$ on X and a sequence of non-negative numbers $\{\alpha_j\}_{j\in\mathbb{N}}$ such that $\mathcal{C}_0 e_j = \alpha_j e_j$ and $\sum_{j=1}^{\infty}\alpha_j < \infty$, i.e., $\{e_j\}_{k\in\mathbb{N}}$ and $\{\alpha_j\}_{k\in\mathbb{N}}$ being the eigenfunctions and eigenvalues of $\mathcal{C}_0$ respectively ([8], Chapter 1). For convenience's sake, we assume that the eigenvalues are in a descending order: $\alpha_j \geq \alpha_{j+1}$ for any $j \in \mathbb{N}$. $\{e_j\}_{j=1}^{\infty}$ are known as the Karhunen–Loève (KL) modes associated with $\mathcal{N}(0, \mathcal{C}_0)$.

2.2. The Crank–Nicolson algorithms

We start by briefly reviewing the family of Crank–Nicolson (CN) algorithms for infinite dimensional Bayesian inferences, developed in [6]. Simply speaking the algorithms are based on the stochastic partial differential equation (SPDE)

$$\frac{du}{ds} = -\mathcal{K}\mathcal{L}u + \sqrt{2\mathcal{K}}\frac{db}{ds}, \tag{2.2}$$

where $\mathcal{L} = \mathcal{C}_0^{-1}$ is the precision operator for μ_0, $\mathcal{K}$ is a positive operator, and b is a Brownian motion in X with covariance operator the identity. The proposal is then derived by applying the CN discretization to the SPDE (2.2), yielding,

$$v = u - \frac{1}{2}\delta\mathcal{K}\mathcal{L}(u+v) + \sqrt{2\mathcal{K}\delta}\xi_0, \tag{2.3}$$

for a white noise ξ_0 and $\delta \in (0,2)$. In [6], two choices of $\mathcal{K}$ are proposed, resulting in two different algorithms. First, one can choose $\mathcal{K} = \mathcal{I}$, the identity, obtaining:

$$(2\mathcal{C} + \delta\mathcal{I})v = (2\mathcal{C} - \delta\mathcal{I})u + \sqrt{8\delta}w,$$

where $w \sim \mathcal{N}(0, \mathcal{C}_0)$, which is known as the plain CN algorithm. Alternatively one can choose $\mathcal{K} = \mathcal{C}_0$, resulting in the pCN proposal:

$$v = (1-\beta^2)^{\frac{1}{2}}u + \beta w, \tag{2.4}$$

where

$$\beta = \frac{\sqrt{8\delta}}{2+\delta}.$$

It is easy to see that $\beta \in [0,1]$. In both CN and pCN algorithms, the acceptance probability is

$$a(v,u) = \min\{1, \exp\Phi(u) - \Phi(v)\}. \tag{2.5}$$

2.3. The adaptive algorithm

To derive the new algorithm, we rewrite the proposal Eq. (2.3) as

$$v = \frac{(I - \frac{1}{2}\delta\mathcal{K}\mathcal{L})}{(I + \frac{1}{2}\delta\mathcal{K}\mathcal{L})}u + \frac{\sqrt{2\delta\mathcal{K}}}{(I + \frac{1}{2}\delta\mathcal{K}\mathcal{L})}\xi_0. \tag{2.6}$$

Now we do a substitution. Namely we let

$$\frac{\sqrt{2\delta\mathcal{K}}}{(I + \frac{1}{2}\delta\mathcal{K}\mathcal{L})} = \beta\sqrt{\mathcal{B}}, \tag{2.7}$$

and by some simply calculation, we can verify that

$$\frac{(I - \frac{1}{2}\delta\mathcal{K}\mathcal{L})}{(I + \frac{1}{2}\delta\mathcal{K}\mathcal{L})} = \sqrt{(\mathcal{I} - \beta^2\mathcal{B}\mathcal{L})}. \tag{2.8}$$

Substitute Eqs. (2.7) and (2.8) into Eq. (2.6), and we obtain a new proposal:

$$v = (\mathcal{I} - \beta^2\mathcal{B}\mathcal{L})^{\frac{1}{2}}u + \beta w_{\mathcal{B}} \tag{2.9}$$

where $w_{\mathcal{B}} \sim \mathcal{N}(0, \mathcal{B})$. This proposal can be understood as a special case of the generalized pCN or the operator weighted proposal. The major difference is that in those two methods, the operator is determined by the derivative information of the likelihood function, while in our algorithm, it is determined with an adaptive method. Before discussing the details of how to determine the operator $\mathcal{B}$, we first show that under mild conditions, the proposal (2.9) results in well-defined acceptance probability in a function space:

Proposition 1. *Suppose operator $\mathcal{B}$ is symmetric positive and of trace class. Let $q(u,\cdot)$ be the proposal distribution associated to Eq. (2.9). Define measures $\eta(du,dv) = q(u,dv)\mu(du)$ and $\eta^{\perp}(du,dv) = q(v,du)\mu(dv)$ on $X \times X$. If $\mathcal{B}$ commutes with $\mathcal{C}_0$, $\eta^{\perp}$ is absolutely continuous with respect to η, and*

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

Proof. Define $\eta_0(du,dv) = q(u,dv)\mu_0(du)$. The measure η_0 is Gaussian. From $\mathcal{B}$ and $\mathcal{C}_0$ are commutable, we have

$$\mathbb{E}^{\eta_0} v \otimes v = (\mathcal{I} - \beta^2\mathcal{B}\mathcal{L})\mathcal{C}_0 + \beta^2\mathcal{B} = \mathcal{C}_0 = \mathbb{E}^{\eta_0} u \otimes u.$$

Then η_0 is symmetric in u, v. Now

$$\eta(du, dv) = q(u, dv)\mu(du), \quad \eta_0(du, dv) = q(u, dv)\mu_0(du),$$

and μ, μ_0 are equivalent. It follows that η and η_0 are equivalent and

$$\frac{d\eta}{d\eta_0}(u, v) = \frac{d\mu}{d\mu_0}(u) = \frac{1}{Z}\exp(-\Phi(u)).$$

Since η_0 is symmetric in u, v we also have that $\eta^\perp$ and η_0 are equivalent and that

$$\frac{d\eta^\perp}{d\eta_0}(u, v) = \frac{1}{Z}\exp(-\Phi(v)).$$

Since equivalence of measures is transitive it follows that η and $\eta^\perp$ are equivalent and

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

It follows immediately from the detailed balance condition that the associated acceptance probability of proposal (2.9) is also given by Eq. (2.5). We reinstate that an important feature of the original pCN algorithm is that its acceptance probability only depends on the function Φ, and as a result the discretization dimensionality has no impact on the acceptance probability up to the numerical error in evaluating Φ. The proposal (2.9) preserves this important feature.

Now we discuss how to specify the operator $\mathcal{B}$, and we start with assuming $\mathcal{B}$ an appropriate parametrized form. Note that an essential condition in Proposition 2.3 is that $\mathcal{B}$ must commute with $\mathcal{C}_0$. To satisfy this condition, it is convenient to design a $\mathcal{B}$ that has common eigenfunctions with $\mathcal{C}_0$. Namely, we write $\mathcal{B}$ in the form of

$$\mathcal{B}\cdot = \sum_{j=1}^{\infty} \lambda_j \langle e_j, \cdot\rangle e_j, \tag{2.10}$$

with λ_j being the coefficients. It is easy to see that $\mathcal{B}$ is a symmetric operator with eigenvalue–eigenfunction pair $\{\lambda_j, e_j\}_{j=1}^{\infty}$, which implies that $\mathcal{B}$ and $\mathcal{C}_0$ commute.

A well-adopted rule in designing efficient MCMC algorithms is that the proposal covariance should be close to the covariance operator of the posterior [21,10]. Now suppose the posterior covariance is $\mathcal{C}$, and one can determine the proposal covariance $\mathcal{B}$ by solving

$$\min_{\{\lambda_j\}_{\lambda=1}^{\infty}} \|\mathcal{B} - \mathcal{C}\|_{HS}, \tag{2.11}$$

where $\|\cdot\|_{HS}$ is the Hilbert–Schmidt norm defined as $\|\mathcal{A}\|_{HS}^2 = \mathrm{Tr}(\mathcal{A}^*\mathcal{A})$ where $\mathcal{A}$ is any bounded operator on X and $\mathcal{A}^*$ is the adjoint of $\mathcal{A}$. By some basic algebra, we can show that the optimal solution of Eq (2.11) is

$$\lambda_j = \langle \mathcal{C}e_j, e_j\rangle$$

for $j = 1...\infty$. Since $\mathcal{C}$ is the posterior covariance, for any v and $v' \in X$, we have [8],

$$\langle \mathcal{C}v, v'\rangle = \int \langle v, u - m\rangle\langle v', u - m\rangle\mu(du), \tag{2.12}$$

where m is the mean of μ. Using Eq. (2.12), we can derive that

$$\lambda_j = \int (x_j - u_j)^2 \mu(du), \tag{2.13}$$

where $x_j = \langle m, e_j\rangle$ and $u_j = \langle u, e_j\rangle$ for $j = 1...\infty$.

In practice, the posterior covariance $\mathcal{C}$ is not directly available, and so here we determine the operator $\mathcal{B}$ with an adaptive MCMC algorithm. Simply speaking, the adaptive algorithm starts with an initial guess of $\mathcal{B}$ and then adaptively updates the $\mathcal{B}$ based on the sample history. Estimating all eigenvalues from the sample history is not practical due to the finite sample size. Here we make a finite-dimension reduction: namely, only the first J eigenvalues are given in the form of Eq. (2.13) which is further estimated from the sample history and the rest of them are taken to be fixed. In particular we let

$$\lambda_j = \begin{cases} \int (x_j - u_j)^2 \mu(du) & \text{for } j \le J \\ \alpha_j & \text{for } j > J. \end{cases} \tag{2.14}$$

The argument that we compute λ_j as is in Eq. (2.14) may become more clear if we look at the projections of the proposal onto each eigenmodes:

$$\langle v, e_j\rangle = \begin{cases} (1 - \beta^2\lambda_i/\alpha_i)^{\frac{1}{2}} u_j + \beta w_j \quad \text{where } w_j \sim \mathcal{N}(0, \lambda_j) & \text{for } j \le J \\ (1 - \beta^2)^{\frac{1}{2}} u_j + \beta w_j \quad \text{where } w_j \sim \mathcal{N}(0, \alpha_j) & \text{for } j > J. \end{cases} \tag{2.15}$$

Eq. (2.15) shows the basic scheme of the algorithm: it performs an adaptive pCN for the KL modes $j \leq J$ with the proposal covariance adapted to approximate that of the posterior, and a standard pCN for all $j > J$. The intuition behind our algorithm is based on the assumption that the (finite-resolution) data is only informative about a finite number of KL modes of the prior. In particular, the data can not provide information about the modes that are highly oscillating (associated with small eigenvalues) and for those modes, the posterior is approximately the prior. In this case, for the modes that are informed by the data, we shall adjust the eigenvalues to approximate the posterior covariance; for those that are not, the best strategy is to simply use the covariance of the prior (which is also the posterior). Finally we note that the same covariance structure has been used in a number of existing works such as [17].

Now we discuss how to update the values of λ_j from posterior samples for $i = 1 \ldots J$. To this end, suppose we have a set of posterior samples $\{u^n\}_{i=0}^n$, and the values of parameters λ_j are estimated using the sample average approximation of Eq. (2.13):

$$x_j^n = \frac{1}{n+1} \sum_{i=0}^{n} \langle u^i, e_j \rangle, \tag{2.16a}$$

$$s_j^n = \sum_{i=0}^{n} (u_j^n)^2, \tag{2.16b}$$

$$\lambda_j^n = \frac{1}{n+1} \sum_{i=0}^{n} (x_j^n - u_j^i)^2 + \epsilon^2, \tag{2.16c}$$

for $j = 1 \ldots J$. Here ϵ is a small constant, introduced to ensure the stability of the algorithm, i.e., to keep λ_j^n from becoming arbitrarily small. For efficiency's sake, we can rewrite Eq (2.16) in a recursive form

$$x_j^n = \frac{n}{n+1} x_j^{n-1} + \frac{1}{n+1} \langle u^n, e_j \rangle, \tag{2.17a}$$

$$s_j^n = s_j^{n-1} + (u_j^n)^2, \tag{2.17b}$$

$$\lambda_j^n = \frac{1}{n+1} s_j^n - (x_j^n)^2, \tag{2.17c}$$

for $j = 1 \ldots J$ and $n > 0$. Note here that, in principle the estimated λ_j^n from samples can be arbitrarily large, which causes issues as $(\mathcal{I} - \beta^2 \mathcal{B}\mathcal{L})$ must not be negative. Thus we let $\lambda_j^n = \min\{\lambda_j^n, \alpha_j\}$ for $j = 1 \ldots J$, and as a result $\lambda_j \leq \alpha_j$ for $j = 1 \ldots J$. It is easy to see that the operator $\mathcal{B}$ resulting from $\{\lambda_j^n\}_{j=1}^J$ is symmetric positive and of trace class. Finally we note that, it is not robust to estimate the parameter values with a very small number of samples, and to address the issue, we first draw a certain number of samples with a standard pCN algorithm before starting the adaptation. We describe the complete adaptive pCN (ApCN) algorithm in Algorithm 1.

Finally an important issue in the implementation is to determine the number of adapted eigenvalues J. Here we propose to let $J = \min\{j \in \mathbb{N}\}$ such that,

$$\frac{\sum_{i=1}^{j} \alpha_j}{\sum_{i=1}^{\infty} \alpha_i} > \rho,$$

where $0 < \rho < 1$ is a prescribed number (e.g. $\rho = 0.99$).

2.4. Ergodicity analysis

It is well known that, the adaptation may destroy the ergodicity of the algorithm, and as a result the chain constructed may not converge to the target distribution. It has been suggested by Roberts and Rosenthal [20] that, an adaptive MCMC algorithm has the correct asymptotic convergence, provided that it satisfies the Diminishing Adaptation (DA) condition, which, loosely speaking, requires the transition probabilities to converge as the iteration proceeds, and the Containment condition. As the latter is regarded as merely a technical condition which is satisfied for virtually all reasonable adaptive schemes [20], it often suffices to prove an adaptive algorithm satisfies the DA condition. Next we show that the proposed ApCN algorithm satisfies the DA condition under a minor modification. Namely, we change Eq. (2.1) to be

$$\frac{d\mu^y}{d\mu_0}(u) = \begin{cases} \frac{1}{Z} \exp(-\Phi(z)), & \|u\|_X \leq R \\ 0, & \|u\|_X > R, \end{cases} \tag{2.18}$$

where R is a prescribed positive constant. We want emphasize here that, just like the work [10], the purpose of the modification is to simplify our proof here, and practically speaking, its impact on the inference results should be negligible, provided that R is taken to be sufficiently large.

Algorithm 1 The adaptive pCN algorithm.

Initialize $u^0 \in S$;
for $n = 0$ to N' **do**
  Propose v using Eq (2.4);
  Draw $\theta \sim U[0, 1]$
  Let $a := \min\{1, \exp[\Phi(u^n) - \Phi(v)]\}$;
  if $\theta \leq a$ **then**
    $u^{n+1} = v$;
  else
    $u^{n+1} = u^n$;
  end if
end for
Compute $\{x_j^{N'}, s_j^{N'}, \lambda_j^{N'}\}_{j=1}^J$ using Eq. (2.16) and samples $\{u^i\}_{i=1}^{N'}$;
for $j = 1$ to J **do**
  $\lambda_j = \min\{\lambda_j, \alpha_j\}$;
end for
for $n = N'$ to N **do**
  Compute $\mathcal{B}$ from Eqs. (2.10) with $\{\lambda_j^n\}_{j=1}^J$;
  Propose v using Eq (2.9);
  Draw $\theta \sim U[0, 1]$
  Let $a := \min\{1, \exp[\Phi(u^n) - \Phi(v)]\}$;
  if $\theta \leq a$ **then**
    $u^{n+1} = v$;
  else
    $u^{n+1} = u^n$;
  end if
  Compute $\{x_j^{n+1}, s_j^{n+1}, \lambda_j^{n+1}\}_{j=1}^J$ using Eqs. (2.17);
end for

Let us now set up some notations. Assume that $\mathcal{B}_n(u_0, u_1, \cdots, u_{n-2}, u)$ is the operator $\mathcal{B}$ at iteration n computed with $u_0, u_1, \cdots, u_{n-2}, u$ through Algorithm 1. For simplicity, we define

$$\mathcal{B}_{n,\zeta_{n-2}}(u) = \mathcal{B}_n(u_0, u_1, \cdots, u_{n-2}, u), \quad \text{where} \quad \zeta_{n-2} = (u_0, u_1, \cdots, u_{n-2}),$$

and let $\{\lambda_{n,i}\}_{i=1}^{\infty}$ be the eigenvalues of $\mathcal{B}_{n,\zeta_{n-2}}(u)$. We define $q_{n,\zeta_{n-2}}(u; dv)$ to be the proposal distribution associated to

$$v = (\mathcal{I} - \beta^2 \mathcal{B}_n(\zeta_{n-2}, u)\mathcal{L})^{\frac{1}{2}} u + \beta w$$

where $w \sim N(0, \mathcal{B}_{n,\zeta_{n-2}}(u))$, and

$$Q_{n,\zeta_{n-2}}(u, dv) = a(u, v) q_{n,\zeta_{n-2}}(u, dv) + \delta_u(dv)(1 - \int a(u, z) q_{n,\zeta_{n-2}}(u, dz))$$

where $a(\cdot, \cdot)$ is given by Eq. (2.5). It can be verified that

$$\sigma_{n,i} = (1 - \beta^2 \frac{\lambda_{n,i}}{\alpha_i})^{\frac{1}{2}}, \tag{2.19}$$

are the eigenvalues of $(\mathcal{I} - \beta^2 \mathcal{B}_{n,\zeta_{n-2}}(u)\mathcal{L})^{\frac{1}{2}}$. We then have the following theorem:

Theorem 1 *(DA condition). There is a fixed positive constant γ such that*

$$\sup_{u \in X} \| Q_{n,\zeta_{n-2}}(u, \cdot) - Q_{n+1,\zeta_{n-1}}(u, \cdot) \| \leq \frac{\gamma}{n}$$

for any ζ_{n-1} and ζ_{n-2} such that ζ_{n-1} is a direct continuation of ζ_{n-2}. Here $\| \cdot \|$ is the total variation norm.

We provide the proof of the theorem in the Appendix.

3. Numerical examples

3.1. An ODE example

Our first example is a simple inverse problem where the forward model is governed by an ordinary differential equation (ODE):

$$\frac{\partial x(t)}{\partial t} = -u(t)x(t)$$

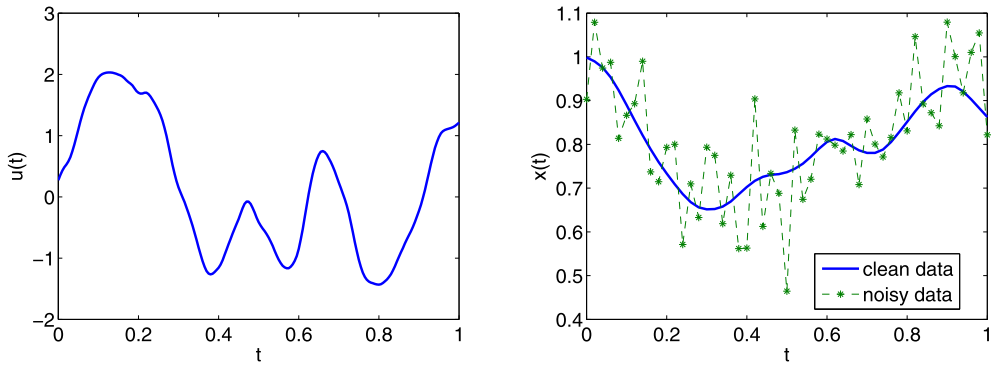

Fig. 1. (For the ODE example.) The truth (left) and the data simulated with it (right).

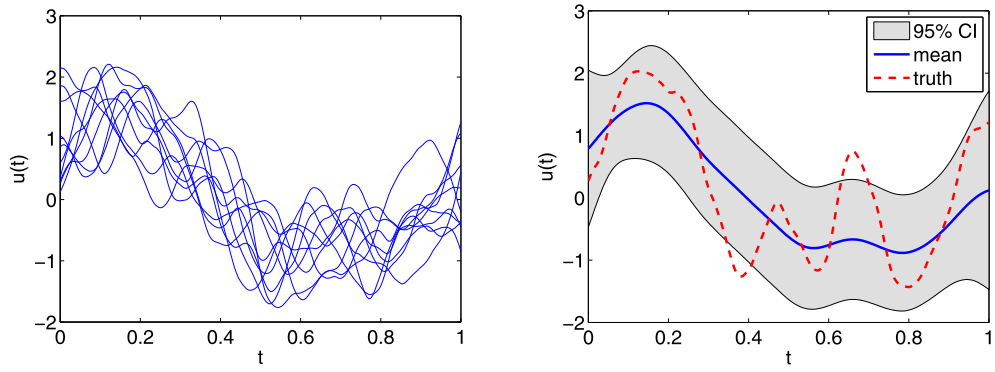

Fig. 2. (For the ODE example.) Left: 10 randomly drawn samples from the posterior. Right: the posterior mean and the 95% confidence interval.

with a prescribed initial condition. Suppose that we observe the solution $x(t)$ several times in the interval $[0, T]$, and we want to infer the unknown coefficient $u(t)$ from the observed data.

In our numerical experiments, we let the initial condition be $x(0) = 1$ and $T = 1$. Now suppose that the solution is measured every $T/100$ time unit from 0 to T and the error in each measurement is assumed to be an independent Gaussian $N(0, 0.1^2)$. The prior is taken to be a zero mean Gaussian with Matérn covariance [18]:

$$K(t_1, t_2) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu}\frac{d}{l})^{\nu} B_{\nu}(\sqrt{2\nu}\frac{d}{l}),$$

where $d = |t_1 - t_2|$, $\Gamma(\cdot)$ is the Gamma function, and $B_{\nu}(\cdot)$ is the modified Bessel function. A random function with the Matérn covariance is $[\nu - 1]$ mean square (MS) differentiable. Several authors suggest that the Matérn covariances can often provide a better model for many real-world physical processes than the popular squared exponential covariances [18]. In this example, we choose $l = 1$, $\sigma = 1$, and $\nu = 5$ implying second order MS differentiability. In the numerical tests, we represent the unknown with 501 grid points. We use synthetic data that is generated by applying the forward model to a true coefficient u and then adding noise to the result. The true coefficient is randomly drawn from the prior distribution. Both the truth and the simulated data are shown in Fig. 1. We perform the proposed adaptive pCN algorithm with 1×10^6 samples and another 5×10^4 pCN samples are used in the pre-run. We set the stepsize $\beta = 1/5$, and we choose $\rho = 0.99$ resulting in $J = 14$, i.e., 14 eigenvalues being adapted.

We show the simulation results in Fig. 2: in the left figure, we show 10 randomly chosen MCMC samples from the posterior, and in the right figure, we plot the posterior mean, as well as the 95% confidence interval, both computed with the MCMC samples. To illustrate the diminishing of the adaption, we plot the 1st and the 14th eigenvalues against the iterations in Fig. 3, and the plots indicate that both parameters tend to converge to certain fixed values as the iterations proceed. For comparison, we also draw 1.05×10^6 samples from the posterior with a standard pCN algorithm where the step size is again taken to be $\beta = 1/5$. In Fig. 4, we compare the autocorrelation function (ACF) of the samples drawn by the two methods at $t = 0.4$ (left) and $t = 0.8$ (right), and the ACF results show that the adaptive pCN method performs better than standard pCN. We then compute the ACF of lag 100 at all the grid points, and show the results in Fig. 5 (left), and we can see that, the ACF of the chain generated by the ApCN is clearly lower than that of the standard pCN at all the grid points. The effective sample size (ESS) is another popular measure of the sampling efficiency of MCMC [12]. ESS is computed by

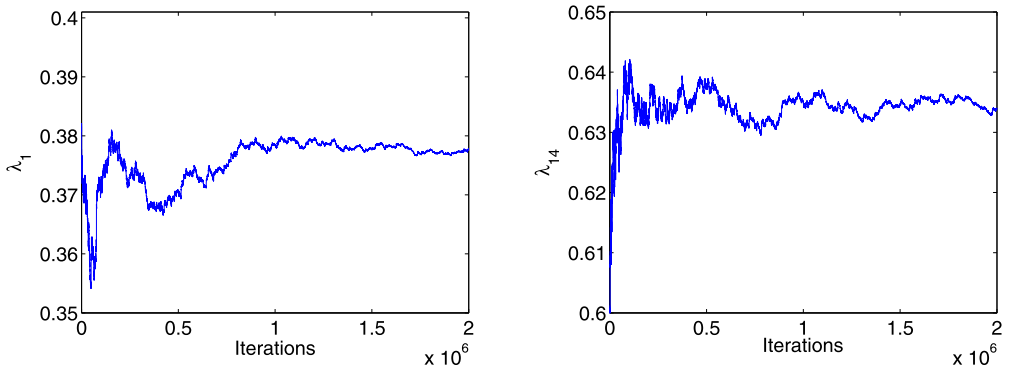

Fig. 3. (For the ODE example.) The eigenvalues λ_1 (left) and λ_{14} (right) plotted against the number of iterations.

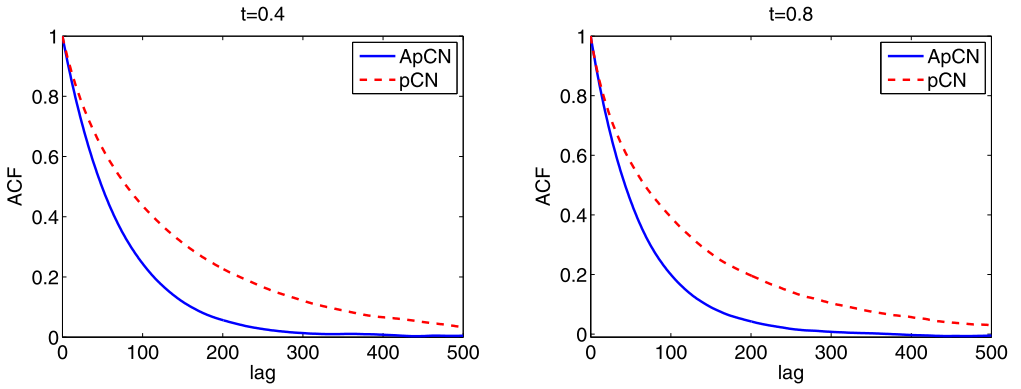

Fig. 4. (For the ODE example.) Autocorrelation functions (ACF) for the pCN and the ApCN methods at $t = 0.2$ and $t = 0.8$.

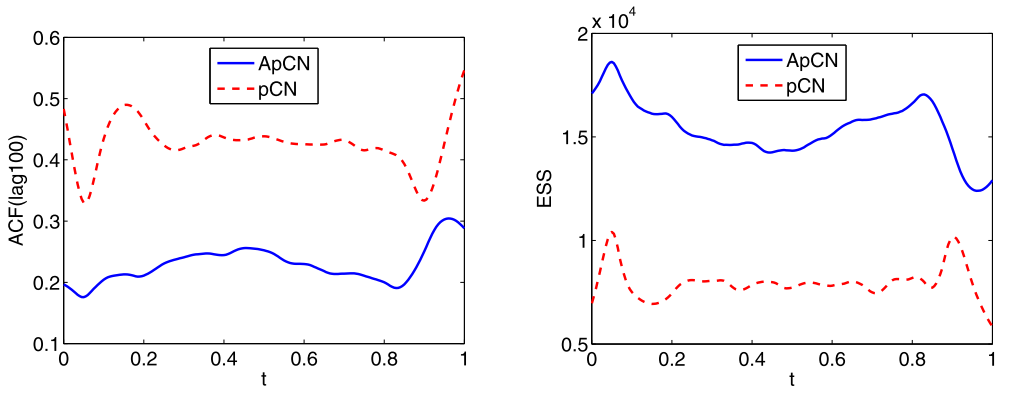

Fig. 5. (For the ODE example.) Left: ACF (lag 100) at each grid point. Right: ESS at each grid point.

$$\mathrm{ESS} = \frac{N}{1 + 2\tau},$$

where τ is the integrated autocorrelation time and N is the total sample size, and it gives an estimate of the number of effectively independent draws in the chain. We compute the ESS of the unknown u at each grid point and show the results in Fig. 5 (right). The results show that the ApCN algorithm produces much more effectively independent samples than the standard pCN.

3.2. Estimating the Robin coefficient

In the second example, we consider a one-dimensional heat conduction equation in the region $x \in [0, L]$,

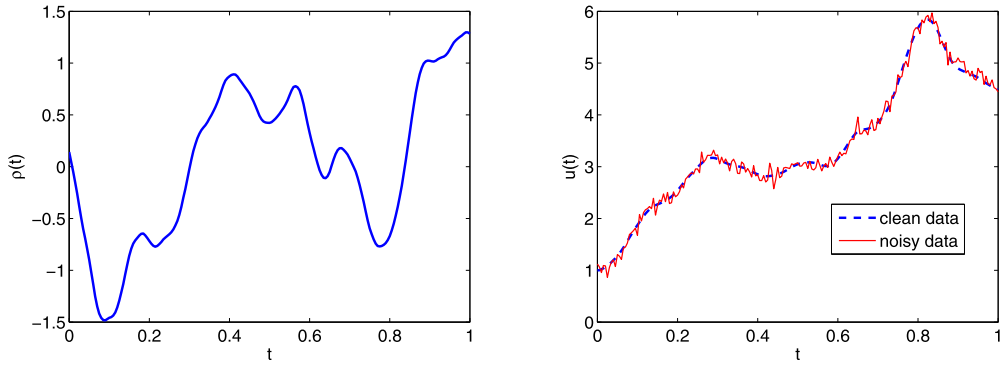

Fig. 6. (For the Robin example.) The truth (left) and the data simulated with it (right).

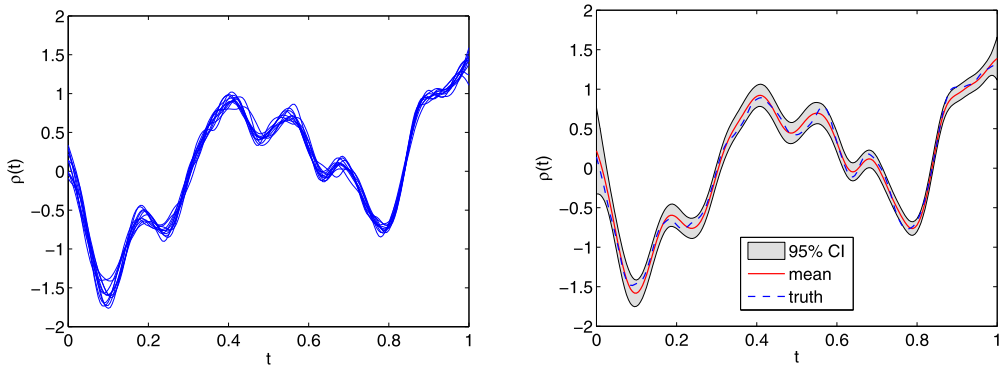

Fig. 7. (For the Robin example.) Left: 10 randomly drawn samples from the posterior. Right: the posterior mean and the 95% confidence interval.

$$\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t), \tag{3.1a}$$

$$u(x,0) = g(x), \tag{3.1b}$$

with the following Robin boundary conditions:

$$-\frac{\partial u}{\partial x}(0,t) + \rho(t)u(0,t) = h_0(t), \tag{3.1c}$$

$$\frac{\partial u}{\partial x}(L,t) + \rho(t)u(L,t) = h_1(t). \tag{3.1d}$$

Suppose the functions $g(x)$, $h_0(x)$ and $h_1(x)$ are all known, and we want to estimate the unknown Robin coefficient $\rho(t)$ from certain measurements of the temperature $u(x,t)$. This example is studied in [25,24]. Here we choose $L = 1$, $T = 1$ and the functions to be

$$g(x) = x^2 + 1, \quad h_0 = t(2t+1), \quad h_1 = 2 + t(2t+2).$$

We assume that a temperature sensor is placed at the end $x = 0$. The temperature is measured every $T/200$ time unit from 0 to T and the error in each measurement is assumed to be an independent Gaussian $N(0, 0.1^2)$. In the computation, 501 equally spaced grid points are used to represent the unknown. Moreover, the prior is the same as that used in the ODE example.

The data is generated the same as the first example, with the true Robin coefficient randomly drawn from the prior distribution. Both the truth and the simulated data are shown in Fig. 6. We implement the adaptive pCN algorithm, where we choose $\beta = 1/5$, and $\rho = 0.99$ resulting in $J = 14$. With the algorithm, we draw 5.5×10^5 samples from the posterior, including 5×10^4 pCN samples in the pre-run, and the average acceptance probability is around 20%. In the left plot of Fig. 7, we show 10 randomly chosen MCMC samples from the posterior, and in the right plot, we show the posterior mean and the 95% confidence interval, both computed with the MCMC samples. Once again, we sample the posterior with standard pCN algorithm for comparison, and in particular we run pCN with two different stepsizes: first we use $\beta = 1/5$ which is the same as that used the ApCN algorithm; we then use $\beta = 1/300$, yielding higher acceptance probability. In each case, we draw 5.5×10^5 samples, and the average acceptance probability for $\beta = 1/5$ is around 0.3%, and that for $\beta = 1/300$ is around 20%, which matches that of the ApCN algorithm. In Fig. 8, we compare the ACF of the samples drawn by the two

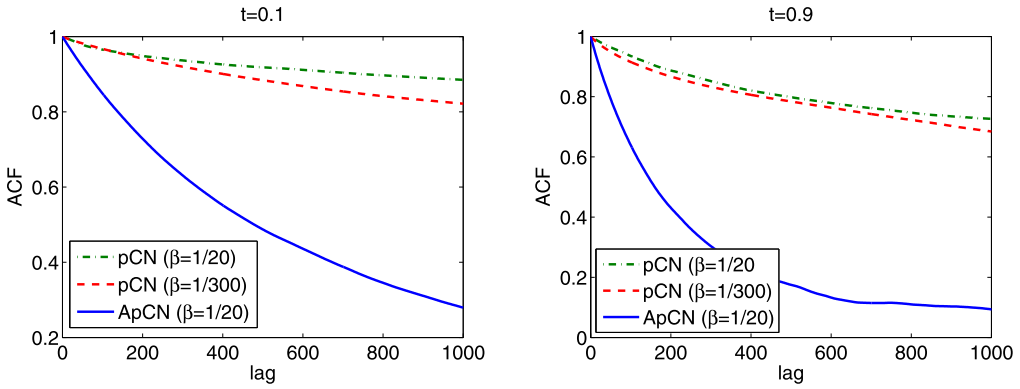

Fig. 8. (For the Robin example.) ACF for the pCN and the ApCN methods at $t = 0.1$ and $t = 0.9$.

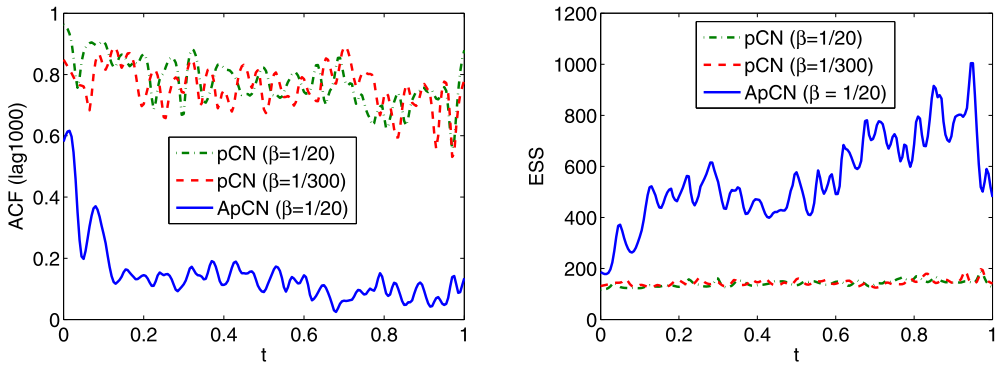

Fig. 9. (For the Robin example.) Left: ACF (lag 1000) at each grid point. Right: the ESS at each grid point.

methods at $t = 0.1$ (left) and $t = 0.9$ (right). One can see from the figures that, the ACF of the chain generated by the pCN with $\beta = 1/300$ decreases slightly faster than that with $\beta = 1/20$, thanks to the higher acceptance probability, while the result of the ApCN is significantly better than both of them. We then compute the ACF of lag 1000 as well as the ESS at all the grid points, and show the results in Fig. 9. Once again, both the ACF and the ESS results suggest that the sampling efficiency of the ApCN is significantly higher than that of the standard pCN algorithm.

4. Conclusions

In summary, we consider MCMC simulations for Bayesian inferences in function spaces. In particular, we develop an adaptive variant of the pCN algorithm to improve the sampling efficiency. The implementation of the ApCN algorithm is rather simple, without requiring any information of the underlying models, and during the iteration the proposal can be efficiently updated with explicit formulas. We also show that the adaptive pCN algorithm satisfies certain ergodicity condition. Finally we demonstrate the effectiveness and efficiency of the ApCN algorithm with several numerical examples. We expect the algorithm can be of use in many practical problems, especially in those involving blackbox models.

The most serious limitation of the present algorithm is that, it determines the “effective subspace” (i.e. the space that we perform adaptation in) from the prior distributions. As a result, the algorithm may become less efficient for problems where the eigenvalues of the priors decay slowly and consequently the dimensionality of effective subspace becomes large. Thus, for such problems, simply determining the effective subspace from the prior may not be a good choice, and one may have to identify the subspace using alternative approaches. We plan to address such problems by making improvements on the present algorithm in a future work.

Acknowledgements

The work was partially supported by the National Natural Science Foundation of China under grant number 11301337. ZH and ZY contribute equally to the work.

Appendix A. Proof of Theorem 1

We provide a proof of Theorem 1 in this appendix, which largely follows the proof for the finite dimensional adaptive Metropolis algorithm given in [10]. We start with the following inequality:

$$
\begin{aligned}
&|Q_{n,\zeta_{n-2}}(u,A) - Q_{n+1,\zeta_{n-1}}(u,A)| \\
&= |\int_A a(u,v) q_{n,\zeta_{n-2}}(u,dv) + \delta_A(u)(1 - \int_X q_{n,\zeta_{n-2}}(u,dz) a(u,z)) \\
&\quad - \int_A a(u,v) q_{n+1,\zeta_{n-1}}(u,dv) + \delta_A(u)(1 - \int_X q_{n+1,\zeta_{n-1}}(u,dz) a(u,z))| \\
&\le 2\int_X a(u,v) |\tfrac{dq_{n,\zeta_{n-2}}(u,\cdot)}{d\mu_0}(v) - \tfrac{dq_{n+1,\zeta_{n-1}}(u,\cdot)}{d\mu_0}(v)| \mu_0(dv) \\
&\le 2\int_X |\tfrac{dq_{n,\zeta_{n-2}}(u,\cdot)}{d\mu_0}(v) - \tfrac{dq_{n+1,\zeta_{n-1}}(u,\cdot)}{d\mu_0}(v)| \mu_0(dv) \\
&\le 2\int_X |\tfrac{dq_{n,\zeta_{n-2}}(u,\cdot)}{d\mu_0}(v) - \tfrac{d\widetilde{q}}{d\mu_0}(v)| \mu_0(dv) + 2\int_X |\tfrac{d\widetilde{q}}{d\mu_0}(v) - \tfrac{dq_{n+1,\zeta_{n-1}}(u,\cdot)}{d\mu_0}(v)| \mu_0(dv),
\end{aligned}
$$

where $\widetilde{q}$ is the Gaussian measure that has the same mean with $q_{n,\zeta_{n-2}}(u,\cdot)$ and has the same covariance operator with $q_{n+1,\zeta_{n-1}}(u,\cdot)$. It should be clear that $\widetilde{q}$ is equivalent to μ_0. Now let

$$
I_1 = \int_X |\frac{dq_{n,\zeta_{n-2}}(u,\cdot)}{d\mu_0}(v) - \frac{d\widetilde{q}}{d\mu_0}(v)| \mu_0(dv) \tag{A.1}
$$

and

$$
I_2 = \int_X |\frac{d\widetilde{q}}{d\mu_0}(v) - \frac{dq_{n+1,\zeta_{n-1}}(u,\cdot)}{d\mu_0}(v)| \mu_0(dv).
$$

First we consider I_1. Since $q_{n,\zeta_{n-2}}(u,\cdot)$ and $\widetilde{q}$ are both Gaussian measures with same mean, and their covariance operators have the same eigenfunctions and at most J different eigenvalues, we can show that,

$$
I_1 = \int_{\mathbb{R}^J} |\prod_{i=1}^J \frac{1}{\sqrt{2\pi\beta^2\lambda_{n,i}}} \exp(-\frac{x_i^2}{2\beta^2\lambda_{n,i}}) - \prod_{i=1}^J \frac{1}{\sqrt{2\pi\beta^2\lambda_{n+1,i}}} \exp(-\frac{x_i^2}{2\beta^2\lambda_{n+1,i}})| dx_1 \cdots dx_J. \tag{A.2}
$$

Thanks to the modified likelihood function (2.18), it is easy to see that there exist constants C_1, $C_2 > 0$ such that

$$
|\lambda_{n,i} - \lambda_{n+1,i}| \le \frac{C_1}{n}, \quad \text{and} \quad \lambda_{n+1,i} \ge C_2, \tag{A.3}
$$

for $i = 1 \ldots J$. Using these results, and by some elementary calculus, one can derive that $I_1 < C_3/n$ for some constant $C_3 > 0$.

We now consider I_2. Let

$$
\Delta m = (\mathcal{I} - \beta^2 \mathcal{B}_{n,\zeta_{n-2}}(u)\mathcal{L})^{\frac{1}{2}} u - (\mathcal{I} - \beta^2 \mathcal{B}_{n+1,\zeta_{n-1}}(u)\mathcal{L})^{\frac{1}{2}} u,
$$

and it can be seen that $\langle \Delta m, e_i \rangle = 0$ for $\forall i > J$. We re-write I_2 as

$$
I_2 = \int_X |1 - \frac{dq_{n+1,\zeta_{n-1}}(u,\cdot)}{d\widetilde{q}}(v)| \widetilde{q}(dv),
$$

where

$$
\frac{dq_{n+1,\zeta_{n-1}}(u,\cdot)}{d\widetilde{q}}(v) = \exp(-\frac{1}{2}\|(\beta^2 \mathcal{B}_{n,\zeta_{n-2}}(u))^{-\frac{1}{2}} \Delta m\|^2 + \langle v, (\beta^2 \mathcal{B}_{n,\zeta_{n-2}}(u))^{-1} \Delta m \rangle).
$$

Similar to I_1, we can also write I_2 as a finite dimensional integral:

$$
\begin{aligned}
I_2 = \int_{\mathbb{R}^J} |\prod_{i=1}^J \frac{1}{\sqrt{2\pi\beta^2\lambda_{n+1,i}}} \exp(-\frac{(x_i - \langle \Delta m, e_i \rangle)^2}{2\beta^2\lambda_{n+1,i}} \\
- \prod_{i=1}^J \frac{1}{\sqrt{2\pi\beta^2\lambda_{n+1,i}}} \exp(-\frac{x_i^2}{2\beta^2\lambda_{n+1,i}})| dx_1 \cdots dx_J.
\end{aligned}
$$

It then follows from Eqs. (2.19) and (A.3) that there exist constants C_4, $C_5 > 0$ such that

$$
|\sigma_{n,i} - \sigma_{n+1,i}| \le \frac{C_4}{n}, \quad \text{and} \quad \sigma_{n,i}, \sigma_{n+1,i} \ge C_5, \tag{A.4}
$$

for $i = 1 \ldots J$. We thus have,

$$|\langle \Delta m, e_i \rangle| = |\sigma_{n,i} - \sigma_{n+1,i}| \cdot |\langle u, e_i \rangle| \leq \frac{C_6}{n},$$

for some constant $C_6 > 0$. Once again, by some elementary calculus, we can obtain $I_2 \leq \frac{C_7}{n}$ for some constant $C_7 > 0$, which completes the proof.

References

[1] Christophe Andrieu, Johannes Thoms, A tutorial on adaptive MCMC, Stat. Comput. 18 (4) (2008) 343–373.
[2] Yves Atchade, Gersende Fort, Eric Moulines, Pierre Priouret, Adaptive Markov chain Monte Carlo: theory and methods, preprint, 2009.
[3] Alexandros Beskos, Gareth Roberts, Andrew Stuart, et al., Optimal scalings for local Metropolis–Hastings chains on nonproduct targets in high dimensions, Ann. Appl. Probab. 19 (3) (2009) 863–898.
[4] Tan Bui-Thanh, Mark Girolami, Solving large-scale PDE-constrained Bayesian inverse problems with Riemann manifold Hamiltonian Monte Carlo, Inverse Probl. 30 (11) (2014) 114014.
[5] Yuxin Chen, David Keyes, Kody J.H. Law, Hatem Ltaief, Accelerated dimension-independent adaptive Metropolis, arXiv preprint, arXiv:1506.05741, 2015.
[6] Simon L. Cotter, Gareth O. Roberts, A.M. Stuart, David White, et al., MCMC methods for functions: modifying old algorithms to make them faster, Stat. Sci. 28 (3) (2013) 424–446.
[7] Tiangang Cui, Kody J.H. Law, Youssef M. Marzouk, Dimension-independent likelihood-informed MCMC, J. Comput. Phys. 304 (2016) 109–137.
[8] Giuseppe Da Prato, An Introduction to Infinite-Dimensional Analysis, Springer, 2006.
[9] Zhe Feng, Jinglai Li, An adaptive independence sampler MCMC algorithm for infinite dimensional Bayesian inferences, arXiv preprint, arXiv:1508.03283, 2015.
[10] Heikki Haario, Eero Saksman, Johanna Tamminen, An adaptive Metropolis algorithm, Bernoulli (2001) 223–242.
[11] Jari Kaipio, Erkki Somersalo, Statistical and Computational Inverse Problems, vol. 160, Springer, 2005.
[12] Robert E. Kass, Bradley P. Carlin, Andrew Gelman, Radford M. Neal, Markov Chain Monte Carlo in practice: a roundtable discussion, Am. Stat. 52 (2) (1998) 93–100.
[13] Kody J.H. Law, Proposals which speed up function-space MCMC, J. Comput. Appl. Math. 262 (2014) 127–138.
[14] James Martin, Lucas C. Wilcox, Carsten Burstedde, Omar Ghattas, A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion, SIAM J. Sci. Comput. 34 (3) (2012) A1460–A1487.
[15] Jonathan C. Mattingly, Natesh S. Pillai, Andrew M. Stuart, et al., Diffusion limits of the random walk Metropolis algorithm in high dimensions, Ann. Appl. Probab. 22 (3) (2012) 881–930.
[16] Noemi Petra, James Martin, Georg Stadler, Omar Ghattas, A computational framework for infinite-dimensional Bayesian inverse problems, part II: stochastic Newton MCMC with application to ice sheet flow inverse problems, SIAM J. Sci. Comput. 36 (4) (2014) A1525–A1555.
[17] Frank J. Pinski, Gideon Simpson, Andrew M. Stuart, Hendrik Weber, Algorithms for Kullback–Leibler approximation of probability measures in infinite dimensions, SIAM J. Sci. Comput. 37 (6) (2015) A2733–A2757.
[18] Carl Edward Rasmussen, Gaussian Processes for Machine Learning, MIT Press, 2006.
[19] Gareth O. Roberts, Andrew Gelman, Walter R. Gilks, et al., Weak convergence and optimal scaling of random walk Metropolis algorithms, Ann. Appl. Probab. 7 (1) (1997) 110–120.
[20] Gareth O. Roberts, Jeffrey S. Rosenthal, Examples of adaptive MCMC, J. Comput. Graph. Stat. 18 (2) (2009) 349–367.
[21] Gareth O. Roberts, Jeffrey S. Rosenthal, et al., Optimal scaling for various Metropolis–Hastings algorithms, Stat. Sci. 16 (4) (2001) 351–367.
[22] Daniel Rudolf, Björn Sprungk, On a generalization of the preconditioned Crank–Nicolson Metropolis algorithm, arXiv preprint, arXiv:1504.03461, 2015.
[23] A.M. Stuart, Inverse problems: a Bayesian perspective, Acta Numer. 19 (2010) 451–559.
[24] Liang Yan, Fenglian Yang, Chuli Fu, A Bayesian inference approach to identify a Robin coefficient in one-dimensional parabolic problems, J. Comput. Appl. Math. 231 (2) (2009) 840–850.
[25] Zhewei Yao, Zixi Hu, Jinglai Li, A TV-Gaussian prior for infinite-dimensional Bayesian inverse problems and its numerical implementations, Inverse Probl. 32 (7) (2016) 075006.