

## A Hybrid Adaptive MCMC Algorithm in Function Spaces\*

Qingping Zhou<sup>†</sup>, Zixi Hu<sup>‡</sup>, Zhewei Yao<sup>‡</sup>, and Jinglai Li<sup>§</sup>

**Abstract.** The preconditioned Crank–Nicolson (pCN) method is a Markov chain Monte Carlo (MCMC) scheme, specifically designed to perform Bayesian inferences in function spaces. Unlike many standard MCMC algorithms, the pCN method can preserve the sampling efficiency under the mesh refinement, a property referred to as being dimension independent. In this work we consider an adaptive strategy to further improve the efficiency of pCN. In particular we develop a hybrid adaptive MCMC method: the algorithm performs an adaptive Metropolis scheme in a chosen finite dimensional subspace and a standard pCN algorithm in the complement space of the chosen subspace. We show that the proposed algorithm satisfies certain important ergodicity conditions. Finally with numerical examples we demonstrate that the proposed method has competitive performance with existing adaptive algorithms.

**Key words.** adaptive Metropolis, Bayesian inference, function space, inverse problems, Markov chain Monte Carlo

**AMS subject classifications.** 65C05, 62F15

**DOI.** 10.1137/16M1082950

**1. Introduction.** Many real-world inverse problems require us to estimate unknowns that are functions of space or time. Solving such problems with the Bayesian approach [12, 23] has become increasingly popular, largely due to its ability to quantify the uncertainty in the estimation results. To implement the Bayesian inference in those problems, it is often required to perform Markov chain Monte Carlo (MCMC) simulations in function spaces. Usually, the unknown is represented with a finite dimensional parametrization, and then MCMC is applied to the resulting finite dimensional problems. Many standard MCMC algorithms, such as the popular random walk Metropolis–Hastings (RWMH), are known to be dimension dependent, as they can become arbitrarily slow as the discretization dimensionality increases [20, 16]. 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 [5] by constructing a Crank–Nicolson (CN) discretiza-

\*Received by the editors July 6, 2016; accepted for publication (in revised form) May 16, 2017; published electronically July 12, 2017.

<http://www.siam.org/journals/juq/5/M108295.html>

**Funding:** This work was supported by the NSFC under grant 11301337.

<sup>†</sup>School of Mathematical Sciences and Institute of Natural Sciences, Shanghai Jiao Tong University, 800 Dongchuan Rd., Shanghai 200240, China ([zhou2015@sjtu.edu.cn](mailto:zhou2015@sjtu.edu.cn)).

<sup>‡</sup>School of Mathematical Sciences, Zhiyuan College, Shanghai Jiao Tong University, 800 Dongchuan Rd., Shanghai 200240, China ([yzw@sjtu.edu.cn](mailto:yzw@sjtu.edu.cn), [hzx@sjtu.edu.cn](mailto:hzx@sjtu.edu.cn)).

<sup>§</sup>Corresponding author. Institute of Natural Sciences, School of Mathematical Sciences, and the MOE Key Laboratory of Scientific and Engineering Computing, Shanghai Jiao Tong University, 800 Dongchuan Rd., Shanghai 200240, China ([jinglaili@sjtu.edu.cn](mailto:jinglaili@sjtu.edu.cn)).

tion of a stochastic partial differential equation (SPDE) that preserves the reference measure. Several variants of the pCN algorithms have been developed to further improve the sampling efficiency. For example, a class of algorithms accelerate the pCN scheme using the gradient information of the likelihood function; such algorithms include the pCN Langevin method [5], the operator-weighted proposal method [14], the dimension-independent likelihood-informed MCMC [6], and the generalized pCN algorithm [22], just to name a few.

In this work, we focus on an alternative way to improve the sampling efficiency, 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, 19] and the references therein) as the iterations proceed. A major advantage of the general 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 [11], we developed an adaptive pCN (ApCN) algorithm based on the idea of adapting the proposal covariance to approximate that of the posterior. The ApCN algorithm requires the proposal covariance to be diagonal, assuming the unknown is represented with the Karhunen–Loève expansion [11], as the implementation involves computing the square root of a large matrix, which is very intensive if the covariance is not diagonal [11]. In this work, we present an improved adaptive MCMC algorithm for functions, particularly addressing this limitation of the ApCN algorithm. The proposed algorithm is essentially a hybrid scheme: it performs an adaptive Metropolis (AM) scheme in a chosen finite dimensional subspace of the state space and a pCN in the complement of it. In the present algorithm, the proposal covariance directly approximates that of the posterior without assuming a diagonal structure. With numerical examples, we show that the present algorithm can outperform the ApCN scheme, thanks to the relaxation of the diagonal structure. Another important improvement of the present method is the convergence property of the algorithm. Recall that, to show the ergodicity property of the ApCN method, we need to impose an artificial modification of the likelihood function to ensure that the support of the posterior is bounded [11]; however, for the present hybrid algorithm, we can show that it satisfies the same ergodicity conditions without modifying the likelihood function.

We note that other MCMC algorithms for the infinite dimensional problems are available, such as the dimension independent adaptive Metropolis [4] and the infinite dimension independence sampler [9]. Comparison of these methods with the pCN based algorithms can be found in [11].

The remainder of the paper is organized as follows. In section 2 we present our hybrid adaptive MCMC algorithm as well as some theoretical results regarding its ergodicity. 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 hybrid adaptive MCMC method.

**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  $\langle \cdot, \cdot \rangle$ . 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  $L(x, y)$ . In the Bayesian inference we assume that the prior  $\mu_0$  of  $u$  is a Gaussian measure defined on  $X$  with mean  $m_0$  covari-

ance operator  $C_0$ , i.e.,  $\mu_0 = N(m_0, C_0)$ . Note that the mean of the Gaussian measure  $m_0$  can be taken to be zero without loss of generality, and  $C_0$  is symmetric positive and of trace class. In this setting, the posterior measure  $\mu^y$  of  $u$  conditional on data  $y$  is provided by the Radon–Nikodym derivative:

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

with  $Z$  being a normalization constant, which can be interpreted as the Bayes rule in the infinite dimensional setting. A typical example is the so-called Bayesian inverse problems [12, 23], which assume that the unknown  $u$  is mapped to the data  $y$  via a forward model  $y = G(u) + \zeta$ , where  $G : X \rightarrow R^d$  and  $\zeta$  is a  $d$ -dimensional Gaussian noise with mean zero and covariance  $C_\zeta$ . In this case,

$$(2) \quad \Phi^y(u) = \frac{1}{2} |C_\zeta^{-\frac{1}{2}}(G(u) - y)|_2^2.$$

For the inference problem to be well-posed, one typically requires the functional  $\Phi^y$  to satisfy [23, Assumptions (2.6)]. Finally we quote the following lemma [8, Chapter 1], which will be useful later.

**Lemma 2.1.** *There exists a complete orthonormal basis  $\{e_j\}_{j \in \mathbb{N}}$  on  $X$  and a sequence of nonnegative numbers  $\{\alpha_j\}_{j \in \mathbb{N}}$  such that  $C_0 e_j = \alpha_j e_j$  and  $\sum_{j=1}^\infty \alpha_j < \infty$ , i.e.,  $\{e_j\}_{j \in \mathbb{N}}$  and  $\{\alpha_j\}_{j \in \mathbb{N}}$  being the eigenfunctions and eigenvalues of  $C_0$ , respectively.*

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  $\mathcal{N}(0, C_0)$ .

**2.2. The preconditioned Crank–Nicolson algorithm.** We now briefly review the basic preconditioned CN algorithm for infinite dimensional Bayesian inference, following the presentation of [5]. Simply speaking the algorithms are based on the SPDE

$$(3) \quad \frac{du}{ds} = -\mathcal{K}\mathcal{L}u + \sqrt{2\mathcal{K}} \frac{db}{ds},$$

where  $\mathcal{L} = C_0^{-1}$  is the precision operator for  $\mu_0$ ,  $\mathcal{K}$  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 (3), yielding

$$(4) \quad v = u - \frac{1}{2} \delta \mathcal{K}\mathcal{L}(u + v) + \sqrt{2\mathcal{K}} \delta \xi_0$$

for a white noise  $\xi_0$  and  $\delta \in (0, 2)$ . In [5], two choices of  $\mathcal{K}$  are proposed, resulting in two different algorithms. First, one can choose  $\mathcal{K} = I$ , the identity, obtaining

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

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

$$(5) \quad v = (1 - \beta^2)^{\frac{1}{2}}u + \beta w,$$

where

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

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

$$(6) \quad a(u, v) = \min\{1, \exp[\Phi^y(u) - \Phi^y(v)]\}.$$

**2.3. The hybrid algorithm.** We start with a nonadaptive version of the proposed hybrid algorithm. For a prescribed integer  $J > 0$  (the interpretation of  $J$  and how to determine it will be discussed later), we let  $X^+ = \text{span}\{e_1, \dots, e_J\}$  and  $X^- = (X^+)^\perp$ , i.e., the orthogonal complement of  $X^+$ . For any  $u \in X$ , we can write  $u = u^+ + u^-$ , where  $u^+ \in X^+$  and  $u^- \in X^-$ . Our algorithm is proposed according to

$$(7a) \quad v = v^+ + v^- \quad \text{with} \quad v^+ = u^+ + \beta w^+, \quad \text{and} \quad v^- = (1 - \beta^2)^{\frac{1}{2}}u^- + \beta w^-,$$

where

$$(7b) \quad w^+ = \sum_{i=1}^J w_i e_i \quad \text{with} \quad (w_1, \dots, w_J)^T \sim N(0, \Sigma)$$

with  $\Sigma$  being a  $J \times J$  covariance matrix (and thus it must be symmetric and positive definite), and

$$(7c) \quad w^- = \sqrt{\mathcal{B}}\xi_0,$$

in which

$$\mathcal{B}\cdot = \sum_{j=J+1}^{\infty} \alpha_j \langle e_j, \cdot \rangle e_j,$$

and  $\xi_0$  is a white Gaussian noise. It is easy to see that the proposal defined by (7) is a Gaussian measure  $\mathcal{N}(m(u), \beta^2 \mathcal{C})$  with mean  $m = u^+ + (1 - \beta^2)^{\frac{1}{2}}u^-$  and covariance  $\mathcal{C}$  such that

$$(8) \quad \mathcal{C} = \sum_{i,j=1}^J \sigma_{i,j} \langle e_i, \cdot \rangle e_j + \mathcal{B}\cdot$$

The key in the algorithm is to choose an appropriate covariance matrix  $\Sigma$ . Before discussing how to choose  $\Sigma$ , we first show that under mild conditions, the proposal (7) results in well-defined acceptance probability in a function space.

**Proposition 2.2.** Let  $q(u, \cdot)$  be the proposal distribution associated to (7). Define measures  $\eta(du, dv) = q(u, dv)\mu^y(du)$  and  $\eta^\perp(du, dv) = q(v, du)\mu^y(dv)$  on  $X \times X$ . If  $\Sigma$  is symmetric and positive definite,  $\eta^\perp$  is absolutely continuous with respect to  $\eta$ , and

$$(9) \quad \frac{d\eta^\perp}{d\eta}(u, v) = \exp \left[ \Phi^y(u) - \Phi^y(v) + \frac{1}{2} \sum_{i=1}^J \frac{(\langle u, e_i \rangle^2 - \langle v, e_i \rangle^2)}{\alpha_i} \right].$$

*Proof.* Define  $\eta_0(du, dv) = q(u, dv)\mu_0(du)$  and  $\eta_0^\perp(du, dv) = q(v, du)\mu_0(dv)$ . Both  $\eta_0$  and  $\eta_0^\perp$  are Gaussian. First, we have

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

and  $\mu^y, \mu_0$  are equivalent. It follows that  $\eta$  and  $\eta_0$  are equivalent and

$$(10) \quad \frac{d\eta}{d\eta_0}(u, v) = \frac{d\mu^y}{d\mu_0}(u) = \frac{1}{Z} \exp(-\Phi^y(u)).$$

Obviously following the same argument, we also have that  $\eta^\perp$  and  $\eta_0^\perp$  are equivalent and

$$(11) \quad \frac{d\eta^\perp}{d\eta_0^\perp}(u, v) = \frac{1}{Z} \exp(-\Phi^y(v)).$$

Now we let  $q'(u, dv)$  be the proposal distribution of the standard pCN algorithm and define  $\eta_{\text{pcn}}(du, dv) = q'(u, dv)\mu_0(du)$  and  $\eta_{\text{pcn}}^\perp(du, dv) = q'(v, du)\mu_0(dv)$ . It is easy to see that

$$(12) \quad \frac{d\eta_0^\perp}{d\eta_0}(u, v) = \frac{d\eta_0^\perp}{d\eta_{\text{pcn}}^\perp}(u, v) \frac{d\eta_{\text{pcn}}^\perp}{d\eta_{\text{pcn}}}(u, v) \frac{d\eta_{\text{pcn}}}{d\eta_0}(u, v),$$

and as is proved in [5],  $\frac{d\eta_{\text{pcn}}^\perp}{d\eta_{\text{pcn}}}(u, v) = 1$ . Moreover, we can see that

$$(13) \quad \frac{d\eta_0^\perp}{d\eta_{\text{pcn}}^\perp}(u, v) = \frac{dq(v, \cdot)}{dq'(v, \cdot)}(u),$$

and by design,

$$(14) \quad \frac{dq(v, \cdot)}{dq'(v, \cdot)}(u) = \exp \left[ -\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \Sigma^{-1}(\mathbf{x} - \mathbf{x}') + \frac{1}{2}(\mathbf{x} - \sqrt{1 - \beta^2} \mathbf{x}')^T C_J^{-1}(\mathbf{x} - \sqrt{1 - \beta^2} \mathbf{x}') \right],$$

where  $\mathbf{x} = (\langle u, e_1 \rangle, \dots, \langle u, e_J \rangle)^T$ ,  $\mathbf{x}' = (\langle v, e_1 \rangle, \dots, \langle v, e_J \rangle)^T$ , and  $C_J$  is a  $J \times J$  diagonal matrix:  $C_J = \text{diag}[\beta^2 \alpha_1, \dots, \beta^2 \alpha_J]$ . Similarly we can show that

$$(15) \quad \begin{aligned} \frac{d\eta_{\text{pcn}}}{d\eta_0}(u, v) &= \frac{dq'(u, \cdot)}{dq(u, \cdot)}(v) \\ &= \exp \left[ \frac{1}{2}(\mathbf{x}' - \mathbf{x})^T \Sigma^{-1}(\mathbf{x}' - \mathbf{x}) - \frac{1}{2}(\mathbf{x}' - \sqrt{1 - \beta^2} \mathbf{x})^T C_J^{-1}(\mathbf{x}' - \sqrt{1 - \beta^2} \mathbf{x}) \right]. \end{aligned}$$

Substituting (14) and (15) into (12) yields

$$(16) \quad \frac{d\eta^\perp}{d\eta_0^\perp}(u, v) = \exp \left[ \frac{1}{2} \sum_{i=1}^J \frac{(\langle u, e_i \rangle^2 - \langle v, e_i \rangle^2)}{\alpha_i} \right].$$

It follows immediately from (10)–(16) that  $\eta$  and  $\eta^\perp$  are equivalent and (9) holds.  $\blacksquare$

From the detailed balance condition (see, for example, [18, Definition 6.45]), one can derive that the acceptance probability of proposal (7) is

$$(17) \quad a(u, v) = \min \left\{ 1, \frac{d\eta^\perp}{d\eta}(u, v) \right\},$$

where  $\frac{d\eta^\perp}{d\eta}(u, v)$  is given by (9).

We now consider how to determine  $\Sigma$ . A rule of thumb in designing efficient MCMC algorithms is that the proposal covariance should be close to the covariance operator of the posterior [20, 10]. Now suppose the posterior covariance is  $\mathcal{C}^y$ , and one can determine the proposal covariance  $\mathcal{C}$  given by (8) by solving

$$(18) \quad \min_{\Sigma} \|\mathcal{C} - \mathcal{C}^y\|_{HS} := \sum_{j=1}^{\infty} \|(\mathcal{C} - \mathcal{C}^y)e_j\|_X^2,$$

where  $\|\cdot\|_{HS}$  is the Hilbert–Schmidt operator norm and  $\|\cdot\|_X$  is the norm defined on the space  $X$ . Since  $\Sigma$  only affects the first  $J$  eigenmodes, (18) is equivalent to

$$(19) \quad \min_{\Sigma} \sum_{j=1}^J \|(\mathcal{C} - \mathcal{C}^y)e_j\|_X^2,$$

which can be written as

$$(20) \quad \sum_{j=1}^J \|(\mathcal{C} - \mathcal{C}^y)e_j\|_X^2 = \sum_{j=1}^J \left\| \sum_{i=1}^J \sigma_{i,j} e_i - \mathcal{C}^y e_j \right\|_X^2 \\ = \sum_{j=1}^J \left[ \left( \sum_{i=1}^J \sigma_{i,j} \right)^2 - 2 \sum_{i=1}^J \sigma_{i,j} \langle \mathcal{C}^y e_j, e_i \rangle + \|\mathcal{C}^y e_j\|_X^2 \right],$$

where  $\sigma_{i,j}$  for  $i, j = 1 \dots J$  are the entries of  $\Sigma$ . Now since the right-hand side of (20) is a quadratic function of  $\{\sigma_{i,j}\}_{i,j=1}^J$ , taking the partial derivative of it with respect to each  $\sigma_{i,j}$  yields the optimal solution of (18):

$$\sigma_{i,j} = \langle \mathcal{C}^y e_j, e_i \rangle \quad \text{for } i, j = 1 \dots J.$$

Since  $\mathcal{C}^y$  is the posterior covariance for any  $v$  and  $v' \in X$ , we have [8]

$$(21) \quad \langle \mathcal{C}^y v, v' \rangle = \int \langle v, u - m^y \rangle \langle v', u - m^y \rangle \mu^y(du),$$

where  $m^y$  is the mean of  $\mu^y$ . Using (21), we can derive that

$$(22) \quad \sigma_{i,j} = \int \langle u - m^y, e_i \rangle \langle u - m^y, e_j \rangle \mu^y(du)$$

for  $i, j = 1 \dots J$ .

Since (22) cannot be computed directly, we estimate the covariance matrix  $\Sigma$  by the adaptive Metropolis method. Simply speaking, the AM algorithm starts with an initial guess of  $\Sigma$  and then adaptively updates the  $\Sigma$  based on the sample history. Namely, suppose we have a set of samples  $\{u_1, \dots, u_n\}$  and let  $x_i$  be the projection of  $u_i$  onto the basis  $(e_1, \dots, e_J)$ :

$$x_i = (\langle u_i, e_1 \rangle, \dots, \langle u_i, e_J \rangle).$$

We estimate  $\Sigma$  with

$$(23a) \quad \hat{x} = \frac{1}{n} \sum_{i=1}^n x_i,$$

$$(23b) \quad \hat{\Sigma} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \hat{x})(x_i - \hat{x})^T + \delta I,$$

where  $\delta$  is a small positive constant and  $I$  is the identity matrix. Note that the term  $\delta I$  in (23b) is introduced to stabilize the iteration, as is used in [10]. For efficiency's sake, (23) can be recast in a recursive form [10, equation (7)]. It should be noted that it is not robust to estimate the parameter values with a very small number of samples, and to this end we employ a prerun, drawing a certain number of samples with a standard pCN algorithm, before starting the adaptation. Finally, we note that, in principle, a sample  $x$  with very large norm can distort the estimate of the covariance matrix  $\Sigma$ , and to prevent this from happening, we introduce a norm threshold  $R \gg 0$ , and if a samples norm exceeds this threshold, it is not used to update the covariance. This step is essential for our convergence results. We describe the complete hybrid adaptive algorithm in Algorithm 1.

The basic idea behind the proposed method may become more clear if we look at the projections of the proposal onto each eigenmode:

$$(24) \quad \langle v, e_i \rangle = \begin{cases} \langle u, e_i \rangle + \beta w_i & \text{for } i \leq J, \\ (1 - \beta^2)^{\frac{1}{2}} \langle u, e_i \rangle + \beta w_i & \text{for } i > J, \end{cases}$$

where  $v$  follows the proposal distribution  $q(u, \cdot)$ ,  $(w_1, \dots, w_J)^T \sim N(0, \Sigma)$ , and  $w_i \sim N(0, \alpha_i)$  for  $i > J$ . Equation (24) shows the *hybrid* nature of the algorithm: it performs an AM algorithm in a finite dimensional space spanned by  $\{e_1, \dots, e_J\}$  with the proposal covariance adapted to approximate that of the posterior and a standard pCN sampler 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, in the finite dimensional subspace spanned by the modes that are significantly informed by the

---

**Algorithm 1** The hybrid adaptive algorithm.

---

```

1: Initialize  $u_1 \in X$ ;
2: Draw  $N'$  samples with a standard pCN algorithm, denoted as  $\{u_i\}_{i=1}^{N'}$ ;
3: Let  $S = \{u_i, i = 1 \dots N' \mid \|u_i\|_X < R\}$ 
4: Compute  $\Sigma$  using (23) and samples in  $S$ ;
5: for  $n = N'$  to  $N' - 1$  do
6:   Propose  $v$  using (7);
7:   Draw  $\theta \sim U[0, 1]$ 
8:   Compute  $a(u, v)$  with (17);
9:   if  $\theta \leq a$  then
10:     $u_{n+1} \leftarrow v$ ;
11:   else
12:     $u_{n+1} \leftarrow u^n$ ;
13:   end if
14:   if  $\|u^{n+1}\|_X < R$  then
15:     $S \leftarrow S \cup \{u^{n+1}\}$ ;
16:    Update  $\Sigma$  using (23) and samples in  $S$ ;
17:   end if
18: end for

```

---

data, we shall perform an AM algorithm to improve the sampling efficiency; in its complement space, we just use the standard pCN method to preserve the dimension independence of the MCMC scheme.

Finally an important issue in the implementation is to determine the number of adapted eigenvalues  $J$ . Following [11], we let

$$J := \min \left\{ j \in \mathbb{N} : \frac{\sum_{i=1}^j \alpha_i}{\sum_{i=1}^{\infty} \alpha_i} > \rho \right\},$$

where  $0 < \rho < 1$  is a prescribed number (e.g.,  $\rho = 0.9$ ). In section 3, with numerical examples, we demonstrate how the choice of  $J$  affects the sampling efficiency of the algorithm.

**2.4. The convergence property.** It is well known that the chain constructed with an adaptive MCMC algorithm may not converge to the target distribution, i.e., losing its ergodicity. Thus, for a new adaptive algorithm, it is important to study whether it can correctly converge to the target distribution. It has been proved by Roberts and Rosenthal [21] that an adaptive MCMC algorithm has the correct asymptotic convergence, provided that it satisfies the diminishing adaptation (DA) and the containment conditions. The DA condition, loosely speaking, requires the transition probabilities to converge as the iteration proceeds, i.e., as in the following.

**Definition 1.** *The DA condition is*

$$\lim_{n \rightarrow \infty} \sup_{u \in X} \|Q_{n+1}(u, \cdot) - Q_n(u, \cdot)\|_{\text{TV}} = 0, \quad \text{in probability,}$$

where  $Q_n$  is the transition kernel at step  $n$ , and  $\|\cdot\|_{\text{TV}}$  is the total variation norm of distributions.



The containment condition [3] is stated as follows.

**Definition 2.** *The containment condition is as follows: for any  $\epsilon > 0$ , the sequence  $\{M_\epsilon(u_n, \Sigma_n)\}_{n=1}^\infty$  is bounded in probability where*

$$M_\epsilon(u, \Sigma) = \inf_n \{n \geq 1 : \|Q_\Sigma^n(u, \cdot) - \pi(\cdot)\|_{TV} < \epsilon\},$$

$Q_\Sigma$  is a transition kernel with the subspace covariance  $\Sigma$ , and  $\pi$  is the target measure.

It has also been suggested by the authors that the containment condition is often merely a technical condition which is satisfied for virtually all reasonable adaptive schemes [19], and thus here we show that the proposed hybrid algorithm satisfies the DA condition. Recall that to show the ApCN algorithm satisfies the DA condition, the likelihood function is modified to be

$$\frac{d\mu^y}{d\mu_0}(u) \propto \begin{cases} \exp(-\Phi^y(u)), & \|u\|_X \leq R_{\max}, \\ 0, & \|u\|_X > R_{\max}, \end{cases}$$

where  $R_{\max}$  is a prescribed positive constant. Removal of this artificial modification is certainly desirable, and in what follows we shall show that the present hybrid algorithm satisfies the DA condition, without making such a modification.

Suppose at iteration  $n$ , we have samples  $\{u_0, u_1, \dots, u_{n-2}, u\}$  and for simplicity we define the notation:  $\zeta_{n-2} = (u_0, u_1, \dots, u_{n-2})$ . Let  $\Sigma_n$  be the subspace covariance matrix estimated with  $\{u_0, u_1, \dots, u_{n-2}, u\}$  using (23), and  $C_{n, \zeta_{n-2}}(u)$  be the corresponding proposal covariance operator. We define  $q_{n, \zeta_{n-2}}(u, dv) = \mathcal{N}(m(u), \beta^2 C_{n, \zeta_{n-2}}(u))$ , i.e., the proposal distribution at iteration  $n$ , and

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

where  $a(\cdot, \cdot)$  is given by (17). We then have the following theorem (the DA condition).

**Theorem 2.3.** *There is a fixed positive constant  $\gamma$  such that*

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

for any  $\zeta_{n-1}$  and  $\zeta_{n-2}$  such that  $\zeta_{n-1}$  is a direct continuation of  $\zeta_{n-2}$ .

*Proof.* First it is easy to see that  $q_{n, \zeta_{n-2}}(u, \cdot)$  and  $q_{n+1, \zeta_{n-1}}(u, \cdot)$  are both Gaussian measures with same mean, and we can write them as

$$(25) \quad q_{n, \zeta_{n-2}}(u, \cdot) = q_{n, \zeta_{n-2}}^J \times \tilde{q}_{n, \zeta_{n-2}}^J, \text{ and } q_{n+1, \zeta_{n-1}}(u, \cdot) = q_{n+1, \zeta_{n-1}}^J \times \tilde{q}_{n+1, \zeta_{n-1}}^J,$$

where  $q_{n, \zeta_{n-2}}^J$  and  $q_{n+1, \zeta_{n-1}}^J$  are Gaussian distributions defined on  $X^+$ , and  $\tilde{q}_{n, \zeta_{n-2}}^J$  and  $\tilde{q}_{n+1, \zeta_{n-1}}^J$  are Gaussian distributions defined on  $X^-$ . It should be clear that by design we have  $\tilde{q}_{n, \zeta_{n-2}}^J = \tilde{q}_{n+1, \zeta_{n-1}}^J$ , and the covariances of  $q_{n, \zeta_{n-2}}^J$  and  $q_{n+1, \zeta_{n-1}}^J$  are  $\Sigma_n$  and  $\Sigma_{n+1}$ , respectively. Thus some simple calculations yield

$$(26) \quad \frac{dq_{n, \zeta_{n-2}}(u, \cdot)}{dq_{n+1, \zeta_{n-1}}(u, \cdot)}(v) = \sqrt{\frac{|\Sigma_{n+1}|}{|\Sigma_n|}} \exp \left[ \frac{1}{2} \Delta x^T (\Sigma_{n+1}^{-1} - \Sigma_n^{-1}) \Delta x \right],$$

where  $\Delta x = (\langle v - u, e_1 \rangle, \dots, \langle v - u, e_J \rangle)^T$ . Let  $A$  be any member of the Borel  $\sigma$ -field of  $X$ , and we compute

$$\begin{aligned}
& |Q_{n, \zeta_{n-2}}(u, A) - Q_{n+1, \zeta_{n-1}}(u, A)| \\
&= \left| \int_A a(u, v) q_{n, \zeta_{n-2}}(u, dv) + \delta_A(u) \left(1 - \int_X a(u, v') q_{n, \zeta_{n-2}}(u, dv')\right) \right. \\
&\quad \left. - \int_A a(u, v) q_{n+1, \zeta_{n-1}}(u, dv) + \delta_A(u) \left(1 - \int_X a(u, v') q_{n+1, \zeta_{n-1}}(u, dv')\right) \right| \\
&\leq 2 \int_X a(u, v) \left| \frac{dq_{n, \zeta_{n-2}}(u, \cdot)}{dq_{n+1, \zeta_{n-1}}(u, \cdot)}(v) - 1 \right| q_{n+1, \zeta_{n-1}}(u, dv) \\
&\leq 2 \int_X \left| \frac{dq_{n, \zeta_{n-2}}(u, \cdot)}{dq_{n+1, \zeta_{n-1}}(u, \cdot)}(v) - 1 \right| q_{n+1, \zeta_{n-1}}(u, dv), \\
&= 2 \int_X \left| \sqrt{\frac{|\Sigma_{n+1}|}{|\Sigma_n|}} \exp \left[ \frac{1}{2} \Delta x^T (\Sigma_{n+1}^{-1} - \Sigma_n^{-1}) \Delta x \right] - 1 \right| q_{n+1, \zeta_{n-1}}(u, dv), \\
&= \frac{2}{(2\pi)^{\frac{J}{2}}} \int_{\mathbb{R}^J} \left| \frac{1}{\sqrt{|\Sigma_n|}} \exp \left( -\frac{1}{2} \Delta x^T \Sigma_n^{-1} \Delta x \right) - \frac{1}{\sqrt{|\Sigma_{n+1}|}} \exp \left( -\frac{1}{2} \Delta x^T \Sigma_{n+1}^{-1} \Delta x \right) \right| d\Delta x \\
&\leq c_1 \|\Sigma_n - \Sigma_{n+1}\|_{HS}
\end{aligned}$$

for some constant  $c_1 > 0$ . Note that the very last inequality in the equation above follows directly from [10, equation (16)]. If  $\|u_n\|_X > R$ ,  $\|\Sigma_n - \Sigma_{n-1}\|_{HS} = 0$ ; otherwise, following the same argument of the proof of [10, Theorem 2], we can show that  $\|\Sigma_n - \Sigma_{n-1}\|_{HS} \leq c_2/n$  for some constant  $c_2 > 0$ . It follows directly that the theorem holds. ■

### 3. Numerical examples.

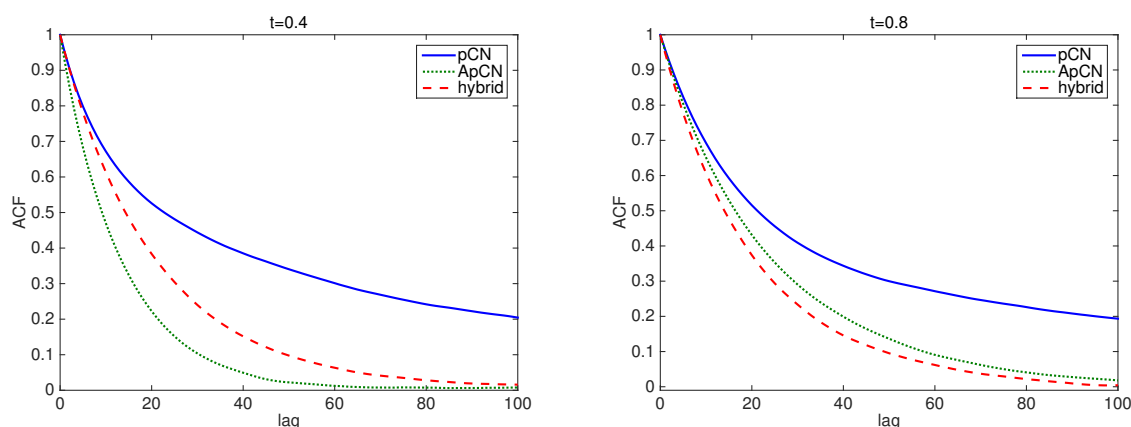
**3.1. A Gaussian example.** Intuitively, we expect that the proposed hybrid method should be advantageous over ApCN in problems where the eigenmodes are strongly correlated. To test this property, we construct a simple mathematical example. We assume the unknown is a function defined on the interval  $[0, 1]$ , and the prior is taken to be a zero mean Gaussian with Matérn covariance [17]:

$$(27) \quad K(t_1, t_2) = \sigma^2 \frac{2^{1-\nu}}{\text{Gam}(\nu)} \left( \sqrt{2\nu} \frac{d}{l} \right)^\nu B_\nu \left( \sqrt{2\nu} \frac{d}{l} \right),$$

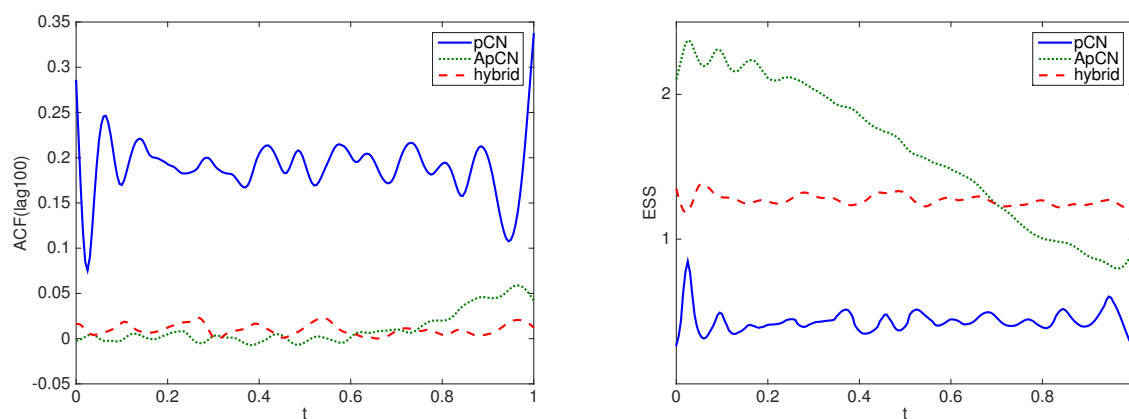
where  $d = |t_1 - t_2|$ ,  $\text{Gam}(\cdot)$  is the Gamma function, and  $B_\nu(\cdot)$  is the modified Bessel function. A random function with the Matérn covariance is  $[\nu - 1]$  mean square (MS) differentiable, and here we choose  $\nu = 5/2$  for all numerical examples implying second order MS differentiability. Moreover, we set  $\sigma = 1$  and  $l = 1$  in this example. We take the function  $\Phi^y(u)$  to be

$$\Phi^y(u) = \frac{1}{2} x^T \Gamma x,$$

where  $x = (\langle u, e_1 \rangle, \dots, \langle u, e_K \rangle)^T$  for a positive integer  $K$  and  $\Gamma[i, j] = \exp(-(i - j)^2/\Delta)$  for  $i, j = 1 \dots K$  and a constant  $\Delta > 0$ . In this example we choose  $K = 14$ . It is easy to see that the posterior distribution is also Gaussian, and by choosing a different value of  $\Delta$  we can



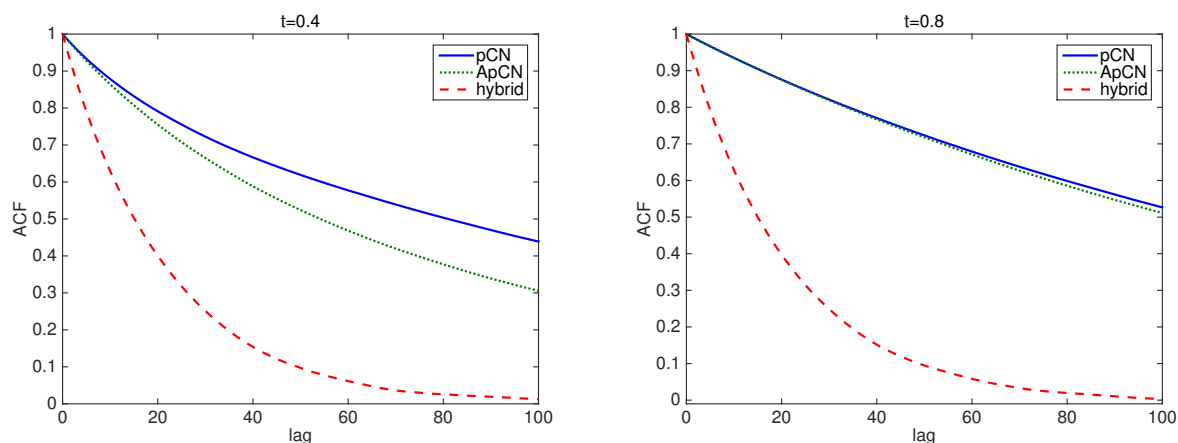
**Figure 1.** (For the Gaussian example:  $\Delta = 1$ ) ACF for the chains drawn by the pCN, the ApCN, and the hybrid methods at  $t = 0.4$  and  $t = 0.8$ .



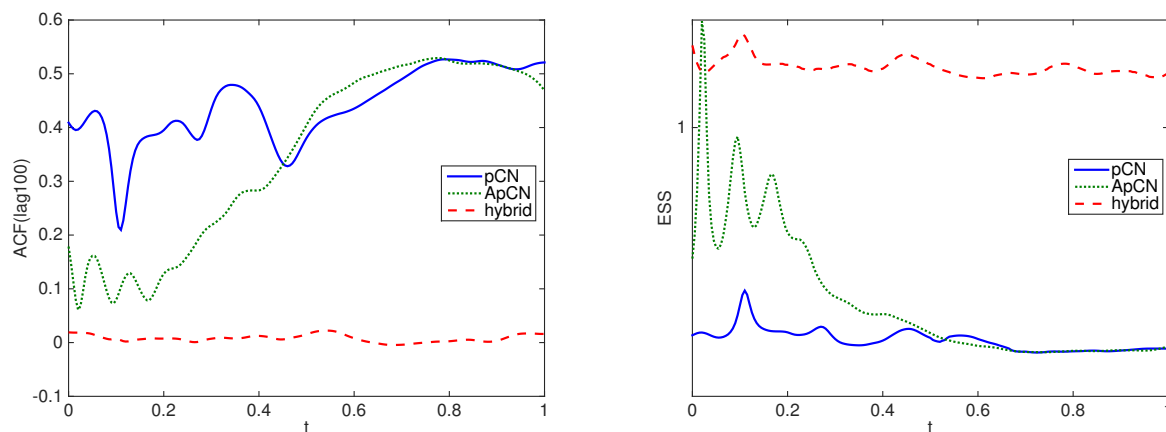
**Figure 2.** (For the Gaussian example:  $\Delta = 1$ ) Left: ACF (lag 100) at each grid point. Right: ESS per 100 samples at each grid point.

control the posterior correlation between the eigenmodes. In particular, we perform numerical tests for the two cases:  $\Delta = 1$  (weak correlation) and  $\Delta = 14$  (strong correlation). In each case, we sample the posterior distribution with three methods: the standard pCN, ApCN, and the hybrid method. For the ApCN and the hybrid methods, we draw  $5 \times 10^5$  samples with another  $0.5 \times 10^5$  pCN samples used in the prurun, and for the pCN method, we directly draw  $5.5 \times 10^5$  samples. Moreover, we set  $J = 14$  in both the ApCN and the hybrid methods. We note that, in all the numerical tests performed in this work, unless stated otherwise, the unknown is represented with  $N = 201$  grid points and the stepsize  $\beta$  has been chosen in a way that the resulting acceptance probability is around 25%. Moreover, in all the numerical examples, we set  $R = 3N\alpha_1$  for the hybrid algorithm.

We first show the results for  $\Delta = 1$ . In Figure 1, we plot the autocorrelation function (ACF) of the samples drawn by each method against the lag at  $t = 0.4$  and  $t = 0.8$ . We then compute the ACF of lag 100 at all the grid points and show the results in Figure 2

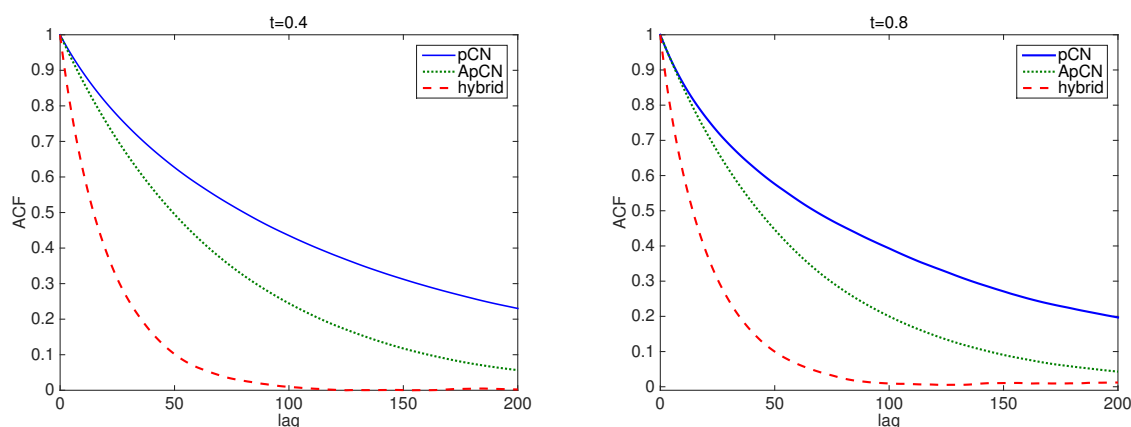


**Figure 3.** (For the Gaussian example:  $\Delta = 14$ ) ACF for the chains drawn by the pCN, the ApCN, and the hybrid methods at  $t = 0.4$  and  $t = 0.8$ .



**Figure 4.** (For the Gaussian example:  $\Delta = 14$ ) Left: ACF (lag 100) at each grid point. Right: ESS per 100 samples at each grid point.

(left). The effective sample size (ESS) is another popular measure of the sampling efficiency of MCMC, which gives an estimate of the number of effectively independent draws in the chain. Specifically the ESS is computed as  $N/(1 + 2\tau)$ , where  $N$  is the number of sample size and  $\tau$  is the integrated autocorrelation time [13]. We compute the ESS per 100 samples of the unknown  $u$  at each grid point and show the results in Figure 2 (right). We then show the same plots for  $\Delta = 14$  in Figures 3 and 4. We can see from these plots that, in the weakly correlated case  $\Delta = 1$ , the hybrid method is not clearly advantageous over the ApCN algorithm; in the strongly correlated case  $\Delta = 14$ , the hybrid method performs much better than the ApCN algorithm, suggesting that taking the covariances between eigenmodes into account can significantly improve the sampling efficiency in this case. These results agree well with our expectations.



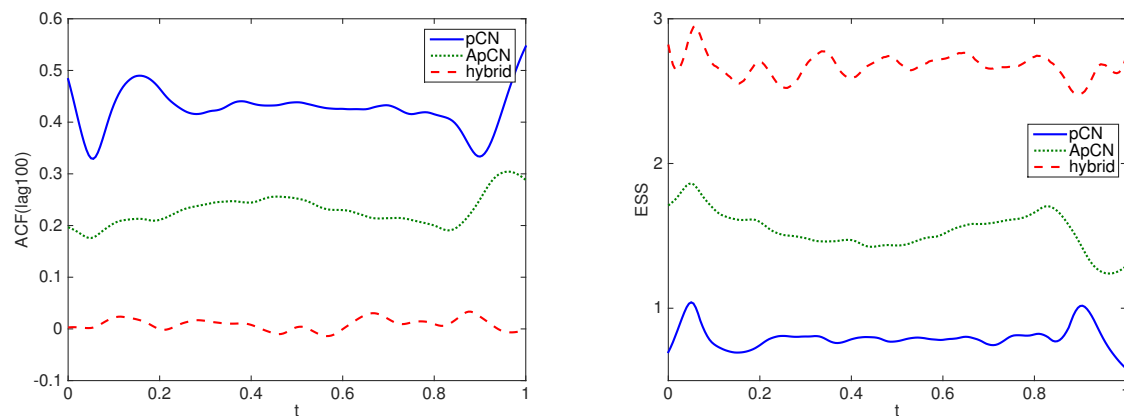
**Figure 5.** (For the ODE example: test 1) ACF for the chains drawn by the pCN, the ApCN, and the hybrid methods at  $t = 0.4$  and  $t = 0.8$ .

**3.2. An ODE example.** Our second example is an inverse problem where the forward model is governed by an ordinary differential equation (ODE):

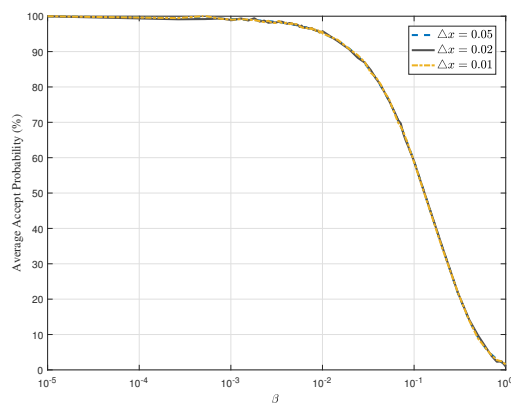
$$\frac{\partial x(t)}{\partial t} = -u(t)x(t)$$

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  $T = 1$ . The equation is solved with a fourth order Runge–Kutta scheme. Now suppose that the solution is measured every  $T/50$  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 covariance specified by (27).

First we want to compare the performance of the hybrid method with that of the ApCN method introduced in [11], and so we use the same problem setup as in [11]: we choose  $l = 1$ ,  $\sigma = 1$ . We also use the same true coefficient  $u(t)$  and synthetic data  $x(t)$  as those in [11]. We draw samples from the posterior with three methods: pCN, ApCN, and the hybrid algorithm. In both ApCN and the hybrid algorithm, we use  $5 \times 10^5$  samples with additional  $0.5 \times 10^5$  pCN samples used in the prurun, and in the standard pCN we directly draw  $5.5 \times 10^5$  samples. In both the ApCN and the hybrid methods, we follow [11] and choose  $J = 14$ , i.e., 14 eigenvalues being adapted. Since the inference results have been reported in [11], we omit them here and only compare the performance of the three methods. In Figure 5, we plot the ACF of the samples drawn by each method against the lag. The results indicate that the ACF of both adaptive algorithms decay faster than the standard pCN, while the ACF of the hybrid algorithm decays faster than that of the ApCN. We then compute the ACF of lag 100 at all the grid points and show the results in Figure 6 (left), and we can see that the ACF of the chain generated by the hybrid method is clearly lower than that of the standard pCN and the ApCN at all the grid points. We compute the ESS per 100 samples of the unknown  $u$  at each grid point and show the results in Figure 6 (right). The results show that the hybrid algorithm



**Figure 6.** (For the ODE example: test 1) Left: ACF (lag 100) at each grid point. Right: ESS per 100 samples at each grid point.

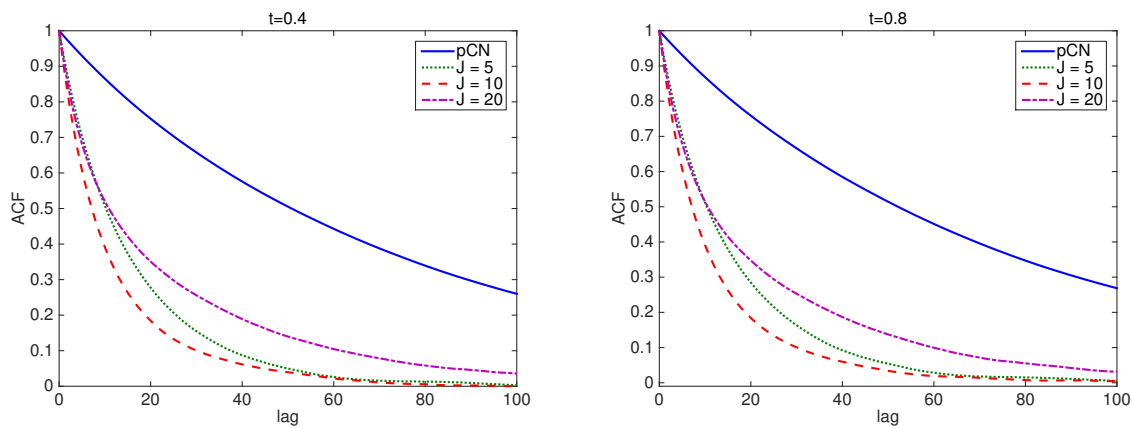


**Figure 7.** (For the ODE example: test 1) Acceptance probabilities with different resolutions are plotted against the stepsize parameter  $\beta$ .

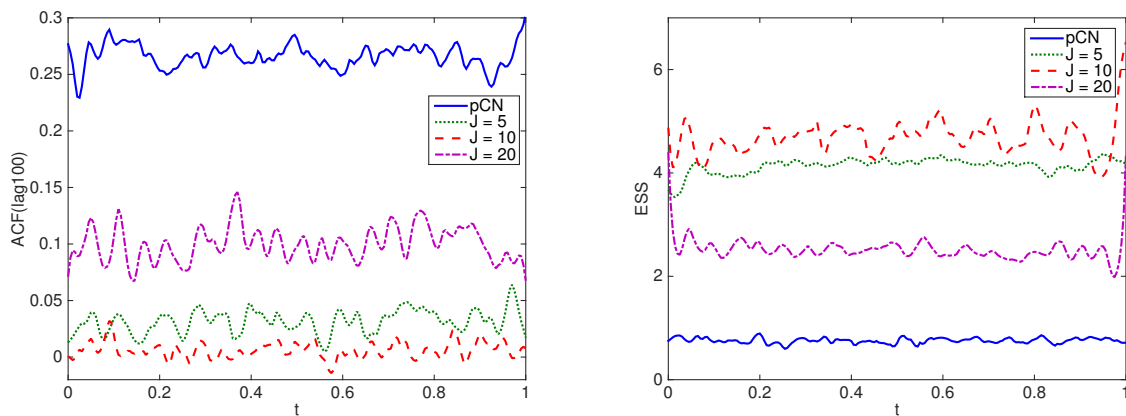
produces much more effectively independent samples than pCN and ApCN. In summary, with this example, we show that the proposed hybrid adaptive method performs better than both the standard pCN and the ApCN methods.

We note that another interesting issue is to show that the proposed algorithm is independent of discretization dimensionality. To show this, we plot in Figure 7 the average acceptance probability as a function of the stepsize parameter  $\beta$  for three different numbers of grid points: 101, 201, and 501. The figure shows that the three plots agree very well with each other, indicating that the proposed hybrid method is indeed dimension-independent.

Next we use the example to test how the value of  $J$  affects the sampling efficiency. We choose  $l = 0.2$ ,  $\sigma = 1$ . A “truth”  $u(t)$  is randomly generated from the prior distribution, and the synthetic data  $x(t)$  is generated by applying the forward model to the generated coefficient  $u$  and then adding noise to the result. We perform the hybrid algorithm with three different values of  $J$ :  $J = 5$ ,  $J = 10$ , and  $J = 20$ , each with  $5 \times 10^5$  plus  $0.5 \times 10^5$  pCN (prerun)



**Figure 8.** (For the ODE example: test 2) ACF for the chains drawn by the hybrid method with  $J = 5, 10, 20$  at  $t = 0.4$  and  $t = 0.8$ .



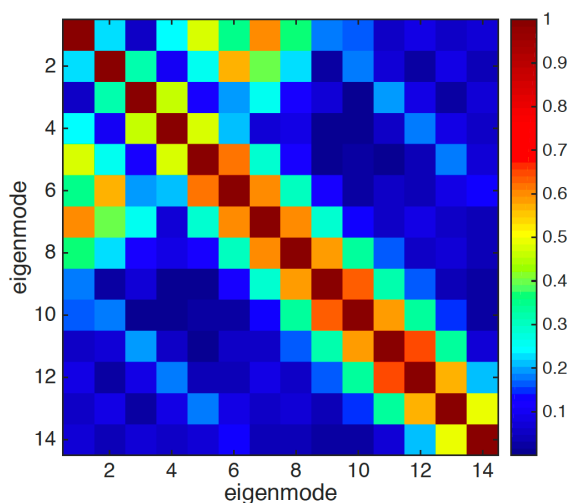
**Figure 9.** (For the ODE example: test 2) Left: ACF (lag 100) at each grid point for  $J = 5, 10, 20$ . Right: ESS per 100 samples at each grid point for  $J = 5, 10, 20$ .

samples. As a comparison, we also perform a standard pCN with  $5.5 \times 10^4$  samples. We plot the ACF as a function of lag at  $t = 0.4$  and  $t = 0.8$  for all the results in Figure 8. In Figures 9 we plot the ACF of lag 100 as well as the ESS at all the grid points. One can see from the plots that the algorithm with  $J = 10$  yields the best results, suggesting that  $J = 10$  may be sufficient for this problem and  $J = 20$  may be too large for the given number of samples. Nevertheless, in all the cases, the hybrid method performs better than the standard pCN.

**3.3. Estimating the Robin coefficient.** In the last example, we consider a one-dimensional heat conduction equation in the region  $x \in [0, L]$ ,

$$(28a) \quad \frac{\partial u}{\partial t}(x, t) = \frac{\partial^2 u}{\partial x^2}(x, t),$$

$$(28b) \quad u(x, 0) = g(x),$$



**Figure 10.** (For the Robin example) The correlation coefficients between any two of the first 14 eigenmodes.

with the following Robin boundary conditions:

$$(28c) \quad -\frac{\partial u}{\partial x}(0, t) + \rho(t)u(0, t) = h_0(t),$$

$$(28d) \quad \frac{\partial u}{\partial x}(L, t) + \rho(t)u(L, t) = h_1(t).$$

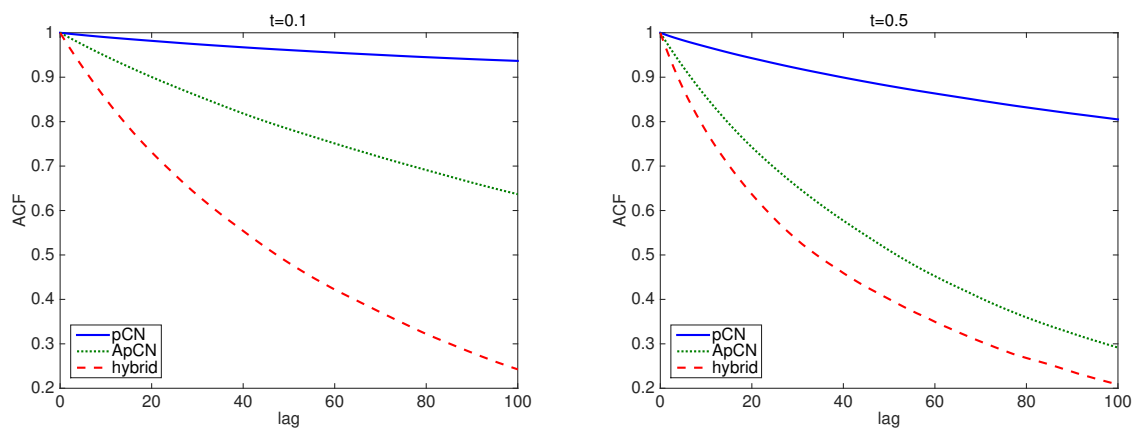
Suppose the functions  $g(x)$ ,  $h_0(x)$ , and  $h_1(x)$  are all known, and we want to estimate the unknown Robin coefficient  $\rho(t)$  from certain measurements of the temperature  $u(x, t)$ . This example is studied in [26, 11, 25], and here we solve the heat equation using the finite difference method as in [25]. Here we choose  $L = 1$ ,  $T = 1$  and the functions to be

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

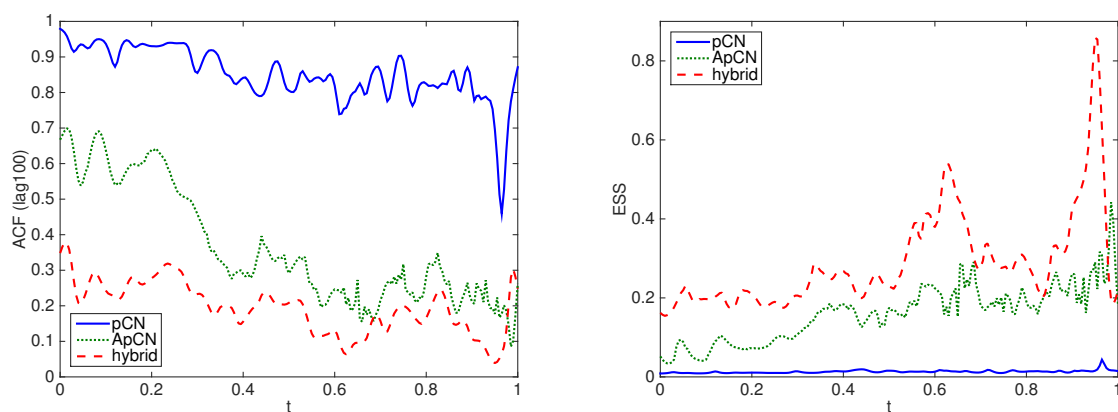
A temperature sensor is placed at the end  $x = 0$ . The solution is measured every  $T/200$  time unit from 0 to  $T$ , and the error in each measurement is assumed to be an independent Gaussian  $N(0, 0.1^2)$ . Moreover, the prior is the same as that used in the first test of the ODE example.

Just like the ODE example, a “true” Robin coefficient  $\rho(t)$  is randomly drawn from the prior distribution and the synthetic data  $u(x, t)$  is generated by applying the forward model to the drawn Robin coefficient  $\rho$  and then adding noise to the result. We sample the posterior distribution with the three methods: pCN, ApCN, and the hybrid algorithm. For the ApCN and the hybrid method, once again we choose  $J = 14$  and draw  $5 \times 10^5$  (adaptive) +  $0.5 \times 10^5$  (prerun) samples. For the pCN method, we draw  $5.5 \times 10^5$  samples using standard pCN directly. Moreover, we plot the correlation coefficients of any two of the first 14 eigenmodes computed from the posterior samples in Figure 10, and the figure shows that many eigenmodes are indeed strongly correlated. In fact, we have found that about one quarter of the correlation coefficients in Figure 10 are larger than 0.3. We now compare the performance of the three





**Figure 11.** (For the Robin example) ACF for the pCN and the ApCN methods at  $t = 0.1$  (left) and  $t = 0.5$  (right).



**Figure 12.** (For the Robin example) Left: ACF (lag 100) at each grid point. Right: the ESS at each grid point.

methods. First we plot the ACF of the samples obtained by the methods at  $t = 0.1$  and  $t = 0.5$  in Figure 11, and then we plot the ACF at lag 100 and the ESS at all the grid points in Figure 12. In all these figures, we can see that, while both adaptive algorithms yield better results than the standard pCN, the hybrid algorithm clearly outperforms the ApCN method, which again indicates that the new algorithm can significantly improve the sampling efficiency over the ApCN approach, by taking the correlations between eigenmodes into account.

**4. Conclusions.** In summary, we consider MCMC simulations for Bayesian inference in function spaces. We develop a hybrid algorithm, which combines the adaptive Metropolis and the pCN algorithm, particularly addressing some limitations of our previously developed ApCN method. The implementation of the proposed algorithm is rather simple, without requiring any information of the underlying models. We also show that the hybrid adaptive algorithm satisfies an important ergodicity condition without making any modifications of the

likelihood function. Finally we demonstrate the efficiency of the hybrid adaptive algorithm with several numerical examples, in which we see that the hybrid algorithm can evidently outperform the ApCN method, thanks to its ability to take into account the correlations between eigenfunctions. Note here that, in problems where the correlations between eigenfunctions are weak, the hybrid may not improve the efficiency much over the ApCN method. Nevertheless, as is illustrated by our numerical examples, in that case, the hybrid algorithm's performance is at least comparable to that of the ApCN. We expect the hybrid adaptive algorithm can be useful in many applied problems, especially in those with underlying models whose gradient information is difficult to obtain.

Some improvements of the hybrid algorithm are possible. First, in the present formulation of the hybrid algorithm, we choose to adapt in the subspace spanned by the eigenfunctions corresponding to the leading eigenvalues. This strategy can be improved by allowing the algorithm to automatically identify this "data-informed subspace" during the iterations [6]. Moreover, reduced models or surrogates (see, e.g., [7, 15, 24]) of the forward operator may be constructed using only the intrinsic dimensions to accelerate the simulation. Another possible improvement is that in the present algorithm the parameter  $R$  is fixed, and one can probably further improve the performance of the algorithm by adapting the parameter  $R$  as well. Another issue is that in this work we only show that the hybrid algorithm satisfies the DA condition, and a more comprehensive study of the ergodicity property of the algorithm is certainly needed. In particular, it is certainly desirable to prove the containment condition for the algorithm as well. We plan to address these issues in future studies.

#### REFERENCES

- [1] C. ANDRIEU AND J. THOMS, *A tutorial on adaptive MCMC*, Stat. Comput., 18 (2008), pp. 343–373.
- [2] Y. ATCHADE, G. FORT, E. MOULINES, AND P. PRIOURET, *Adaptive Markov chain Monte Carlo: Theory and methods*, in Bayesian Time Series Models, D. Barber, A. T. Cemgil, and S. Chiappa, eds., Cambridge University Press, Cambridge, 2011, pp. 32–51.
- [3] Y. BAI, G. O. ROBERTS, AND J. S. ROSENTHAL, *On the Containment Condition for Adaptive Markov Chain Monte Carlo Algorithms*, <http://probability.ca/jeff/ftpdir/yanbai1.pdf> (2009).
- [4] Y. CHEN, D. KEYES, K. J. LAW, AND H. LTAIEF, *Accelerated dimension-independent adaptive Metropolis*, SIAM J. Sci. Comput., 38 (2016), pp. S539–S565.
- [5] S. L. COTTER, G. O. ROBERTS, A. STUART, AND D. WHITE, *MCMC methods for functions: modifying old algorithms to make them faster*, Statist. Sci., 28 (2013), pp. 424–446.
- [6] T. CUI, K. J. LAW, AND Y. M. MARZOUK, *Dimension-independent likelihood-informed MCMC*, J. Comput. Phys., 304 (2016), pp. 109–137.
- [7] T. CUI, Y. M. MARZOUK, AND K. E. WILLCOX, *Data-driven model reduction for the Bayesian solution of inverse problems*, Internat. J. Numer. Methods Engrg., 102 (2015), pp. 966–990.
- [8] G. DA PRATO, *An introduction to Infinite-Dimensional Analysis*, Springer, New York, 2006.
- [9] Z. FENG AND J. LI, *An Adaptive Independence Sampler MCMC Algorithm for Infinite Dimensional Bayesian Inferences*, preprint 1508.03283, 2015.
- [10] H. HAARIO, E. SAKSMAN, AND J. TAMMINEN, *An adaptive Metropolis algorithm*, Bernoulli, 7 (2001), pp. 223–242.
- [11] Z. HU, Z. YAO, AND J. LI, *On an adaptive preconditioned Crank–Nicolson algorithm for infinite dimensional Bayesian inferences*, J. Comput. Phys., 332 (2017), pp. 492–503.
- [12] J. KAIPIO AND E. SOMERSALO, *Statistical and Computational Inverse Problems*, Vol. 160, Springer, New York, 2006.
- [13] R. E. KASS, B. P. CARLIN, A. GELMAN, AND R. M. NEAL, *Markov Chain Monte Carlo in Practice: A Roundtable Discussion*, Amer. Statist., 52 (1998), pp. 93–100.

- [14] K. J. LAW, *Proposals which speed up function-space MCMC*, J. Comput. Appl. Math., 262 (2014), pp. 127–138.
- [15] J. LI AND Y. M. MARZOUK, *Adaptive construction of surrogates for the Bayesian solution of inverse problems*, SIAM J. Sci. Comput., 36 (2014), pp. A1163–A1186.
- [16] J. C. MATTINGLY, N. S. PILLAI, AND A. M. STUART, *Diffusion limits of the random walk Metropolis algorithm in high dimensions*, Ann. Appl. Probab., 22 (2012), pp. 881–930.
- [17] C. E. RASMUSSEN, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006.
- [18] C. ROBERT AND G. CASELLA, *Monte Carlo Statistical Methods*, 2nd ed., Springer, New York, 2004.
- [19] G. O. ROBERTS AND J. S. ROSENTHAL, *Examples of adaptive MCMC*, J. Comput. Graph. Statist., 18 (2009), pp. 349–367.
- [20] G. O. ROBERTS AND J. S. ROSENTHAL, *Optimal scaling for various Metropolis–Hastings algorithms*, Statist. Sci., 16 (2001), pp. 351–367.
- [21] G. ROBERTS AND J. ROSENTHAL, *Coupling and ergodicity of adaptive MCMC*, J. Appl. Probab., 44 (2007), pp. 458–475.
- [22] D. RUDOLF AND B. SPRUNGK, *On a generalization of the preconditioned Crank–Nicolson Metropolis algorithm*, Found. Comput. Math., (2015), pp. 1–35.
- [23] A. M. STUART, *Inverse Problems: A Bayesian Perspective*, Acta Numer., 19 (2010), pp. 451–559.
- [24] L. YAN AND L. GUO, *Stochastic collocation algorithms using  $l_1$ -minimization for Bayesian solution of inverse problems*, SIAM J. Sci. Comput., 37 (2015), pp. A1410–A1435.
- [25] F. YANG, L. YAN, AND T. WEI, *The identification of a Robin coefficient by a conjugate gradient method*, Internat. J. Numer. Methods Engrg., 78 (2009), pp. 800–816.
- [26] Z. YAO, Z. HU, AND J. LI, *A TV-Gaussian prior for infinite-dimensional Bayesian inverse problems and its numerical implementations*, Inverse Problems, 32 (2016), 075006.