COMMUN. MATH. SCI.
Vol. 22, No. 3, pp. 655–684

APPROXIMATE PRIMAL-DUAL FIXED-POINT BASED LANGEVIN ALGORITHMS FOR NON-SMOOTH CONVEX POTENTIALS*

ZIRUO CAI†, JINGLAI LI‡, AND XIAOQUN ZHANG§

Abstract. The Langevin algorithms are frequently used to sample the posterior distributions in Bayesian inference. In many practical problems, however, the posterior distributions often consist of non-differentiable components, posing challenges for the standard Langevin algorithms, as they require to evaluate the gradient of the energy function in each iteration. To this end, a popular remedy is to utilize the proximity operator, and as a result one needs to solve a proximity subproblem in each iteration. The conventional practice is to solve the subproblems accurately, which can be exceedingly expensive, as the subproblem needs to be solved in each iteration. We propose an approximate primal-dual fixed-point algorithm for solving the subproblem, which only seeks an approximate solution of the subproblem and therefore reduces the computational cost considerably. We provide theoretical analysis of the proposed method and also demonstrate its performance with numerical examples.

Keywords. Bayesian inference; Langevin alorithms; non-smooth convex potentials; proximity operators.

AMS subject classifications. 62F15; 65C05; 68U10.

1. Introduction

The Bayesian inference approach has become increasingly popular as a tool for solving inverse problems [19, 36], largely due to its ability to quantify the uncertainty in the results. Simply put the Bayesian approach casts the sought parameter as a random variable and computes its a posterior probability distribution, conditional on the data observed. The ability to accurately and efficiently compute the posterior distribution is crucial for the implementation of the Bayesian framework in real-world problems. A common practice to compute the posterior distributions is to generate samples from them, via some sampling schemes, such as the Markov Chain Monte Carlo (MCMC) methods [18]. To this end, the Langevin algorithm based Monte Carlo (LMC) methods [15, 26, 27, 34] attract significant attention, mainly due to its ability to efficiently explore the state space. Loosely speaking, the Langevin algorithm consists of the following steps: it first constructs a Langevin system with the target distribution as its invariant measure, numerically solves the resulting Langevin system with random initial conditions for sufficiently long time, and regards the final states as samples drawn from the target distribution. In particular the Langevin systems are usually solved with the Euler-Maruyama discretization, yielding a sampling scheme analogous to the gradient descent method for optimization. The algorithm can be incorporated into a MCMC framework by adding a Metropolis-Hastings accept-reject step, resulting in the so-called Metropolis-adjusted Langevin algorithm (MALA) [14, 31–34, 37]; as a contrast, the Langevin algorithms without the Metropolis adjustment are usually referred to as the unadjusted Langevin algorithms (ULA). We consider both types of Langevin algorithms in this work.

*Received: November 06, 2022; Accepted (in revised form): August 16, 2023. Communicated by Wotao Yin.

†School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai, 200240, China (sjtu_caiziruo@sjtu.edu.cn).

‡School of Mathematics, University of Birmingham, Birmingham, UK (j.li.10@bham.ac.uk).

§School of Mathematical Sciences, MOE-LSC and Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai, 200240, China (xqzhang@sjtu.edu.cn).

The standard Langevin algorithms require the evaluation of the gradient of the energy function associated to the target distribution in each iteration. In many Bayesian inverse problems, however, non-differentiable prior distributions are often used – a notable example is the Total Variation (TV) prior used in image reconstruction problems. In such problems the posteriors are also not differentiable, which renders the standard LMC algorithms infeasible. Considerable efforts have been devoted to developing Langevin algorithms for non-differentiable distributions [6, 13, 21–23, 25, 29, 35]

Among the existing methods, a very popular class of methods borrows ideas from the non-smooth optimization research, constructing an approximation of the actual target distribution, and as a result a convex proximity subproblem is solved in each iteration [13, 23, 25, 29]. This idea has been used in both Metropolis adjusted and the unadjusted algorithms. The computational cost of these methods is typically much higher than the standard Langevin algorithms, as they require to solve a proximity subproblem in each iteration. In this regard, it is of critical importance to improve the efficiency in solving the subproblem. This work is devoted to addressing the issue and our approach has the following two main ingredients. First, we adopt the primal-dual fixed point (PDFP) method developed in [7] for non-smooth convex composite potentials $U(x)=f(x)+g(Bx)$ to solve the proximity subproblem. Simply speaking, PDFP solves a non-smooth optimization problem using the primal-dual formulation, and it has been shown that the method has many desired theoretical and computational properties in [7, 38]. More importantly, we propose that, it may not be necessary to solve the subproblem accurately as is usually done in the existing methods; rather an approximate solution obtained by conducting a small number of optimization iterations may suffice for the sampling accuracy while reducing the computational cost considerably. We study the strategy via both theoretical analysis and numerical experiments. Theoretically we provide analysis of the sampling error due to the finite-step subproblem optimization. Via numerical experiments, we demonstrate that the approximate PDFP (i.e., that with a small number of optimization iterations) based Langevin algorithms, especially the Metropolis-adjusted version, have very competitive performance in terms of sampling efficiency.

The rest of the paper is organized as follows: Section 2 reviews the standard Langevin algorithms for smooth distributions. Section 3 considers sampling non-smooth distributions and introduces the proximal MALA (PMALA) approach in particular. We present the approximate PDFP based Langevin algorithm in Section 4 and provide its nonasymptotic error analysis in Section 5. Two numerical examples are provided in Section 6 to demonstrate the performance of the proposed methods and finally Section 7 concludes the paper.

2. The standard Langevin algorithms

We start with a brief introduction to the standard Langevin algorithms for differentiable energy functions. Our goal here is to draw samples from a probability density in the form of

$$\pi(\theta) \propto \exp\left(-U(\theta)\right), \quad \theta \in \mathbb{R}^d, \tag{2.1}$$

where $U(\theta)$ is the energy function. Throughout this work we assume that the energy function $U(\theta)$ is convex and lower semi-continuous, which is an essential presumption for many theoretical studies. Note here that the normalizing constant of π in Equation (2.1) is usually not available in practice, and as such the sampling methods should not require its knowledge.

Assuming $U(\theta)$ is differentiable, we can write down the following Langevin system:

$$\begin{aligned} \mathrm{d}L_t &= \nabla \log \pi(L_t)\mathrm{d}t + \sqrt{2}\mathrm{d}W_t \\ &= -\nabla U(L_t)\mathrm{d}t + \sqrt{2}\mathrm{d}W_t, \end{aligned} \tag{2.2}$$

where W_t is a standard Wiener process. It should be clear that π is the invariant distribution of process L_t. Apply the Euler-Maruyama discretization to Equation (2.2), and we obtain the ULA update:

Algorithm 1: ULA

Result: $\{\theta_n\}_{n=1}^N$.
L_U : Lipschitz constant of ∇U.
Set $\delta \in (0, 1/L_U]$, $\theta_0 \in \mathbb{R}^d$.
for $n = 0$ *to* $N-1$ **do**
  $\theta_{n+1} = \theta_n - \delta \nabla U(\theta_n) + \sqrt{2\delta}\xi_n, \quad \xi_n \sim \mathcal{N}(0, I)$.

The choice of δ is given by [9] that an upper bound of δ related to the Lipschitz continuity of ∇U should imply the convergence of ULA, and an upper bound of $\mathbb{E}U(\theta_n)$.

To remove the bias of ULA, a popular adjustment is to add a Metropolis step to this ULA [31–33], resulting in the following procedure:

- Propose a new state by ULA: $Y_{n+1} = \theta_n - \delta \nabla U(\theta_n) + \sqrt{2\delta}\xi_n, \quad \xi_n \sim \mathcal{N}(0, I)$.
- Compute acceptance rate: $A(Y_{n+1}, \theta_n) = \min\left(1, \dfrac{\pi(Y_{n+1})}{\pi(\theta_n)} \cdot \dfrac{p(\theta_n | Y_{n+1})}{p(Y_{n+1} | \theta_n)}\right)$

$$= \min\left(1, \frac{\pi(Y_{n+1})}{\pi(\theta_n)} \cdot \frac{\exp\left(-\frac{1}{4\delta}\|\theta_n - Y_{n+1} - \delta \nabla U(Y_{n+1})\|_2^2\right)}{\exp\left(-\frac{1}{4\delta}\|Y_{n+1} - \theta_n - \delta \nabla U(\theta_n)\|_2^2\right)}\right).$$

- Draw $a \sim U[0,1]$.
- If $a < A(Y_{n+1}, \theta_n)$; let $\theta_{n+1} = Y_{n+1}$; otherwise, let $\theta_{n+1} = \theta_n$.

The theoretical properties of the ULA have been extensively studied. Provided that one can have access to the accurate gradient ∇U, the nonasymptotic analysis on convergence and errors is given in [9] for strongly convex U and [12] for convex U. Moreover, [11] studies the problem in the convex optimization perspective, by separately considering the gradient descent step and the random walk step in the ULA iteration. When the accurate evaluations of the gradient ∇U are not available, [10] investigates the case of using inaccurate gradient when U is strongly convex. Many techniques and results provided in [9] will be used here in our theoretical analysis.

3. Langevin algorithms for non-smooth distributions

In many real-world applications the energy function U includes some non-differentiable terms. Obviously the ULA and the MALA algorithms introduced in Section 2 can not be used directly in this case. A straightforward solution is to use the subgradient of $U(\cdot)$ in such problems, but the algorithm becomes significantly inefficient compared to the case of a smooth distribution as is demonstrated in [29]. In this section we will discuss a proximal Langevin algorithm framework [13, 29] for non-differentiable energy functions.

3.1. Definitions and propositions. We first provide some definitions and lemmas that are used in the rest of this work, all of which can be found in [1,3].

DEFINITION 3.1. *The proximity operator* $\operatorname{prox}_f(x):\mathbb{R}^d\to\mathbb{R}^d$ *of function* $f:\mathbb{R}^d\to\mathbb{R}$ *is defined by*

$$\operatorname{prox}_f(x):=\arg\min_{u\in\mathbb{R}^d}\left(\frac{\|u-x\|^2}{2}+f(u)\right). \tag{3.1}$$

DEFINITION 3.2. *An operator* $T:\mathbb{R}^d\to\mathbb{R}^d$ *is firmly nonexpansive if and only if*

$$\|Tx-Ty\|_2^2\leqslant\langle Tx-Ty,x-y\rangle,\quad \forall x,y\in\mathbb{R}^d.$$

DEFINITION 3.3. *Let* $m\in\mathbb{R},m>0$. *A function* f *is* m*-strongly convex if only if*

$$f(y)\geqslant f(x)+\langle\nabla f(x),y-x\rangle+\frac{m}{2}\|y-x\|_2^2,\quad \forall x,y\in\mathbb{R}^d. \tag{3.2}$$

LEMMA 3.1. *Let* $m\in\mathbb{R},m>0$. *If* f *is* m*-strongly convex, then*

$$\langle x-y,\nabla f(x)-\nabla f(y)\rangle\geqslant m\|x-y\|_2^2,\quad \forall x,y\in\mathbb{R}^d. \tag{3.3}$$

LEMMA 3.2. *Let* $m\in\mathbb{R},m>0$. *Function* $h(x)$ *is* m*-strongly convex if and only if* $h(x)-\frac{m}{2}\|x\|_2^2$ *is convex.*

LEMMA 3.3. *For convex function* $f:\mathbb{R}^d\to\mathbb{R}$, prox_f *and* $I-\operatorname{prox}_f$ *are firmly nonexpansive operators.*

DEFINITION 3.4. *Function* f *has* M*-Lipschitz continuous gradient if*

$$\|\nabla f(x)-\nabla f(y)\|_2\leqslant M\|x-y\|_2,\quad \forall x,y\in\mathbb{R}^d. \tag{3.4}$$

LEMMA 3.4. *If* f *has* M*-Lipschitz continuous gradient, then*

$$f(y)\leqslant f(x)+\langle\nabla f(x),y-x\rangle+\frac{M}{2}\|y-x\|_2^2,\quad \forall x,y\in\mathbb{R}^d. \tag{3.5}$$

Moreover, if f *is convex, then*

$$M\langle x-y,\nabla f(x)-\nabla f(y)\rangle\geqslant\|\nabla f(x)-\nabla f(y)\|_2^2,\quad \forall x,y\in\mathbb{R}^d. \tag{3.6}$$

DEFINITION 3.5. *The conjugate function of function* g *is defined by*

$$g^*(v)=\sup_{y\in dom(g)}\left(v^Ty-g(y)\right), \tag{3.7}$$

where $v\in V=dom(g^*)=\{v|g^*(v)<\infty\}$.

3.2. Proximal Langevin algorithms. To tackle this situation when U is convex but non-smooth, [29] and [13, 30] respectively replace the original π with two continuously differentiable distributions which can be arbitrarily close to π. In this work we follow the Moreau approximation settings in [29], for any $\rho>0$, define the $\rho-$Moreau approximation of π as

$$\pi_\rho(\theta)=\frac{1}{K'}\sup_{u\in\mathbb{R}^d}\left(\pi(u)\exp\left(-\frac{\|u-\theta\|^2}{2\rho}\right)\right). \tag{3.8}$$

By simple computation,

$$\pi_\rho(\theta)=\frac{1}{K'}\exp(-U_\rho(\theta)), \tag{3.9}$$

where

$$U_\rho(\theta)=\min_{y\in\mathbb{R}^d}\left(U(y)+\frac{\|y-\theta\|_2^2}{2\rho}\right)=U(\text{prox}_{\rho U}(\theta))+\frac{\|\text{prox}_{\rho U}(\theta)-\theta\|^2}{2\rho}$$

is the Moreau envelope [24] of $U(\theta)$. By [1, 8], π_ρ and U_ρ have several useful properties summarized in Lemma 3.5:

LEMMA 3.5.

(1) *When $\rho\to 0$, $\pi_\rho(\theta)\to\pi(\theta)$ pointwisely and $U_\rho(\theta)\to U(\theta)$ pointwisely.*

(2) *$U_\rho(\theta)$ is convex and has $\frac{1}{\rho}-$Lipschitz continuous gradient.*

(3) *$U(\theta)$ and $U_\rho(\theta)$ have the same minimizers.*

(4) *Even though π and U can be non-differentiable, π_ρ and U_ρ are continuously differentiable and*

$$\nabla U_\rho(\theta)=\frac{\theta-\text{prox}_{\rho U}(\theta)}{\rho}. \tag{3.10}$$

Replace the original π with π_ρ in Langevin diffusion (2.2) and one obtains the SDE

$$\begin{aligned}\mathrm{d}L_t^\rho&=\nabla\log\pi_\rho(L_t^\rho)\mathrm{d}t+\sqrt{2}\mathrm{d}W_t\\&=-\nabla U_\rho(L_t^\rho)\mathrm{d}t+\sqrt{2}\mathrm{d}W_t.\end{aligned} \tag{3.11}$$

Here the solution $L_t^\rho\to\pi_\rho$ in TV norm as $t\to+\infty$ from Lemma 5.8 (see also Lemma 1 in [9]). By Euler-Maruyama discretization and Lemma 3.5 (4) one obtains the proximal ULA [29]:

$$\begin{aligned}\theta_{n+1}&=\theta_n-\delta\nabla U_\rho(\theta_n)+\sqrt{2\delta}\xi_n\\&=\theta_n-\delta\frac{\theta_n-\text{prox}_{\rho U}(\theta_n)}{\rho}+\sqrt{2\delta}\xi_n,\quad \xi_n\sim\mathcal{N}(0,I)\\&=\left(1-\frac{\delta}{\rho}\right)\theta_n+\frac{\delta}{\rho}\text{prox}_{\rho U}(\theta_n)+\sqrt{2\delta}\xi_n,\quad \xi_n\sim\mathcal{N}(0,I).\end{aligned} \tag{3.12}$$

Basically ρ is the parameter of the Moreau approximation and δ is the stepsize of the Euler-Maruyama discretization, therefore δ should be independent of ρ. For the stability of the algorithm δ should be within $(0,\rho]$ (Proposition 1 in [9]), and [29] sets $\delta=\rho$ yielding a more concise algorithm:

$$\Rightarrow\theta_{n+1}=\text{prox}_{\delta h}(\theta_n)+\sqrt{2\delta}\xi_n,\quad \xi_n\sim\mathcal{N}(0,I). \tag{3.13}$$

Algorithm 2: Proximal ULA

Result: $\{\theta_n\}_{n=1}^N$.
Set $\rho > 0$, $\delta \in (0, \rho]$, $\theta_0 \in \mathbb{R}^d$.
for $n = 0$ *to* $N-1$ **do**

$$\theta_{n+1} = \left(1 - \frac{\delta}{\rho}\right)\theta_n + \frac{\delta}{\rho}\operatorname{prox}_{\rho U}(\theta_n) + \sqrt{2\delta}\xi_n, \quad \xi_n \sim \mathcal{N}(0, I)$$

However, for discretization error analysis one should fix ρ and let $\delta \to 0$. In this work, we do not constrain δ to be equal to ρ and in the later section we denote proximal ULA by Algorithm 2.

So far, proximal ULA (Algorithm 2) has introduced two errors to draw samples from π: one is the Moreau approximation error from π to π_ρ, another is the discretization error from Langevin diffusion (3.11) to Algorithm 2. One can eliminate these errors by adding a Metropolis-Hastings accept-reject step [31–34] and turns proximal ULA into proximal Metropolis-Adjusted Langevin Algorithm (MALA) [29]:

Algorithm 3: Proximal MALA

Result: $\{\theta_n\}_{n=1}^N$.
Set $\rho > 0$, $\delta \in (0, \rho]$, $\theta_0 \in \mathbb{R}^d$.
for $n = 0$ *to* $N-1$ **do**
  if $n > 0$ **then**
    Compute $\operatorname{prox}_{\rho U}(\theta_n)$ according to the previous accept-reject step:
    $\operatorname{prox}_{\rho U}(\theta_n) = \operatorname{prox}_{\rho U}(\theta_{n-1})$, or $\operatorname{prox}_{\rho U}(\theta_n) = \operatorname{prox}_{\rho U}(Y_n)$
  Propose a new state by proximal ULA:

$$Y_{n+1} = \left(1 - \frac{\delta}{\rho}\right)\theta_n + \frac{\delta}{\rho}\operatorname{prox}_{\rho U}(\theta_n) + \sqrt{2\delta}\xi_n, \quad \xi_n \sim \mathcal{N}(0, I)$$

  Compute acceptance rate: $A(Y_{n+1}, \theta_n) = \min\left(1, \frac{\pi(Y_{n+1})}{\pi(\theta_n)} \cdot \frac{p(\theta_n | Y_{n+1})}{p(Y_{n+1} | \theta_n)}\right)$

$$= \min\left(1, \frac{\pi(Y_{n+1})}{\pi(\theta_n)} \cdot \frac{\exp\left(-\frac{1}{4\delta}\left\|\theta_n - \left(1 - \frac{\delta}{\rho}\right)Y_{n+1} - \frac{\delta}{\rho}\operatorname{prox}_{\rho U}(Y_{n+1})\right\|_2^2\right)}{\exp\left(-\frac{1}{4\delta}\left\|Y_{n+1} - \left(1 - \frac{\delta}{\rho}\right)\theta_n - \frac{\delta}{\rho}\operatorname{prox}_{\rho U}(\theta_n)\right\|_2^2\right)}\right)$$

  Sample a from uniform distribution: $a \sim U[0, 1]$.
  if $a < A(Y_{n+1}, \theta_n)$ **then**
    Accept Y_{n+1}: $\theta_{n+1} = Y_{n+1}$
  else
    Reject Y_{n+1}: $\theta_{n+1} = \theta_n$

From above, $\{\theta_n\}_{n=1}^N$ in Algorithm 3 is actually a Metropolis-Hastings Markov chain proposed by proximal ULA. Noted that the information of the Moreau approximation π_ρ is included in the proposal step but in the accept-reject step π is evaluated instead. For efficient computation, in the n-th iteration we need to know both $\operatorname{prox}_{\rho U}(\theta_n)$ and $\operatorname{prox}_{\rho U}(Y_{n+1})$, but actually only $\operatorname{prox}_{\rho U}(Y_{n+1})$ need to be computed since $\operatorname{prox}_{\rho U}(\theta_n)$ can be obtained from the $(n-1)$-th iteration: If $\theta_n = Y_n$ then

$\text{prox}_{\rho U}(\theta_n) = \text{prox}_{\rho U}(Y_n)$, which has been computed in the $(n-1)$-th iteration. If $\theta_n = \theta_{n-1}$ then $\text{prox}_{\rho U}(\theta_n) = \text{prox}_{\rho U}(\theta_{n-1})$. The challenge is, each iteration of the sampling scheme involves solving an optimization problem $\text{prox}_{\rho U}(x)$. In both PMALA [29] and MYULA [30], the algorithm of [4] is used to solve the subproblem, and in next section we will discuss an alternative method for this.

4. The approximate PDFP based Langevin Algorithms

4.1. The primal-dual fixed point algorithm. Before considering drawing samples from the given distribution $\pi(\theta)$, in this section we introduce the Primal-Dual Fixed Point (PDFP) algorithm developed in [7] and some of its theoretical results. Here we assume that the energy function U can be decomposed into two parts

$$U(x) = f(x) + g(Bx), \tag{4.1}$$

where

- f is convex and continuously differentiable with a M_2-Lipschitz gradient.
- B is a linear operator.
- g is convex and perhaps non-differentiable but having a proximity operator $\text{prox}_g(x)$ which is generally easy to compute.

Note here that Equation (4.1) is a very generic form of posterior distributions in Bayesian inference.

Recall the following convex minimization problem which can be understood as computing a point estimator by maximizing the posterior distribution:

$$\min_{x\in\mathbb{R}^d} \; f(x) + g(Bx). \tag{4.2}$$

Alternatively, Equation (4.2) can be reformulated as a min-max problem

$$\min_{x\in\mathbb{R}^d} \max_{v\in V} \; f(x) + \langle Bx, v\rangle - g^*(v). \tag{4.3}$$

Both problem (4.2) and its min-max reformulation (4.3) have been well studied in the last decades, e.g., [5, 16]. The PDFP method (detailed in Algorithm 4) is a fixed point iteration based algorithm to solve the min-max problem (4.3) and consequently it solves problem (4.2) as well.

Algorithm 4: Primal-Dual Fixed Point method for problem (4.2)

Result: $\{x_k\}_{k=1}^K$.

Set $0 < \lambda \leqslant \dfrac{1}{\lambda_{\max}(BB^T)}$, $0 < \gamma < \dfrac{2}{M_2}$, $x_0 \in \mathbb{R}^d$, $v_0 \in V$.

for $k = 0$ *to* $K-1$ **do**

$$y_{k+1} = x_k - \gamma\nabla f(x_k) - \gamma B^T v_k$$

$$v_{k+1} = \text{prox}_{\frac{\lambda}{\gamma}g^*}\left(\frac{\lambda}{\gamma}By_{k+1} + v_k\right)$$

$$x_{k+1} = x_k - \gamma\nabla f(x_k) - \gamma B^T v_{k+1}$$

As one can see, Algorithm 4 generates two sequences, the primal variable sequence $\{x_k\}_{k=1}^K$ and the dual variable sequence $\{v_k\}_{k=1}^K$. For the min-max problem (4.3), x_k and v_k will converge to the optimal primal point x^* and the optimal dual point v^*

respectively. Note that the convergence of PDFP (Algorithm 4) does not require the strong convexity of $U(x)$, but from Theorem 3.7 in [7] one has the linear convergence rate when $f(x)$ is strongly convex and $\rho_{\min}(BB^T)>0$.

To simplify the notation, in the k-th iteration of Algorithm 4 one denotes $T_1(v_k,x_k)$ and $T_2(v_k,x_k)$ by

$$\begin{cases} v_{k+1}=\operatorname{prox}_{\frac{\lambda}{\gamma}g^*}\left(\frac{\lambda}{\gamma}B\left(x_k-\gamma\nabla f(x_k)-\gamma B^T v_k\right)+v_k\right) & =: T_1(v_k,x_k) \\ x_{k+1}=x_k-\gamma\nabla f(x_k)-\gamma B^T T_1(v_k,x_k) & =: T_2(v_k,x_k). \end{cases} \tag{4.4}$$

Define the operator $T(v,x)$ by

$$T(v_k,x_k):=(v_{k+1},x_{k+1})=(T_1(v_k,x_k),T_2(v_k,x_k)), \tag{4.5}$$

then one can deduce the fixed point property of PDFP proved in [7]:

LEMMA 4.1. *(v^*,x^*) is a fixed point of T:*

$$(v^*,x^*)=T(v^*,x^*), \tag{4.6}$$

which is

$$\begin{cases} v^*=\operatorname{prox}_{\frac{\lambda}{\gamma}g^*}\left(\frac{\lambda}{\gamma}B(x^*-\gamma\nabla f(x^*))+(I-\lambda BB^T)v^*\right) \\ x^*=x^*-\gamma\nabla f(x^*)-\gamma B^T v^*. \end{cases} \tag{4.7}$$

Different from Theorem 3.7 in [7], here we give another version of the linear convergence lemma of PDFP. This lemma shows that $x_k\to x^*$ and $v_k\to v^*$ simultaneously, but the linear convergence rate is for (v,x) with the norm defined by $\|(v,x)\|_{\frac{\gamma^2}{\lambda}}:=\sqrt{\|x\|_2^2+\frac{\gamma^2}{\lambda}\|v\|_2^2}$, which means x_k alone does not necessarily converge at a linear rate to x^* ignoring v_k. For the simplification of notation we define $\phi(x):=x-\gamma\nabla f(x)$, $M:=I-\lambda BB^T$.

LEMMA 4.2. *Assume that x^* and v^* are the optimal solutions of problem (4.3). Assume that $\{x_k\}_k$ and $\{v_k\}_k$ are the two sequences generated by Algorithm 4. Assume that γ,λ are the parameters in Algorithm 4. If $\rho_{\min}(BB^T)>0$ and $\exists\eta_1\in[0,1)$ such that $\|\phi(x)-\phi(y)\|_2\leqslant\eta_1\|x-y\|_2$, $\forall x,y\in\mathbb{R}^d$, then $\forall k\in\mathbb{N}$,*

$$\|x_k-x^*\|_2^2+\frac{\gamma^2}{\lambda}\|v_k-v^*\|_2^2\leqslant\eta^k\left(\|x_0-x^*\|_2^2+\frac{\gamma^2}{\lambda}\|v_0-v^*\|_2^2\right),\quad 0\leqslant\eta<1, \tag{4.8}$$

where $\eta=\max\left(\eta_1^2,1-\lambda\rho_{\min}(BB^T)\right)$.

Proof. See Appendix A.1. □

REMARK 4.1. If f is m_f-strongly convex, then the condition that $\|\phi(x)-\phi(y)\|_2\leqslant\eta_1\|x-y\|_2$, $\eta_1<1$ is easily satisfied:

$$\begin{aligned} &\|\phi(x)-\phi(y)\|_2^2=\|x-y-\gamma(\nabla f(x)-\nabla f(y))\|_2^2 \\ &=\|x-y\|_2^2+\gamma^2\|\nabla f(x)-\nabla f(y)\|_2^2-2\gamma\langle x-y,\nabla f(x)-\nabla f(y)\rangle \\ &\leqslant\|x-y\|_2^2-\left(\frac{2\gamma}{M_2}-\gamma^2\right)\|\nabla f(x)-\nabla f(y)\|_2^2. \end{aligned} \tag{4.9}$$

The inequality follows from the fact that f has M_2-Lipschitz gradient and Lemma 3.4.

Since $0<\gamma<\dfrac{2}{M_2}$, we have $\dfrac{2\gamma}{M_2}-\gamma^2>0$. From the assumption that f is m_f-strongly convex and Lemma 3.1, we have

$$\begin{aligned}&m_f\|x-y\|_2^2\leqslant\langle x-y,\nabla f(x)-\nabla f(y)\rangle\leqslant\|x-y\|_2\|\nabla f(x)-\nabla f(y)\|_2,\quad\forall x,y\in\mathbb{R}^d\\&\Rightarrow m_f\|x-y\|_2\leqslant\|\nabla f(x)-\nabla f(y)\|_2,\quad\forall x,y\in\mathbb{R}^d.\end{aligned}\tag{4.10}$$

Then from (4.9),

$$\|\phi(x)-\phi(y)\|_2^2\leqslant\left(1-m_f^2\left(\frac{2\gamma}{M_2}-\gamma^2\right)\right)\|x-y\|_2^2.\tag{4.11}$$

Therefore $\eta_1=\sqrt{1-m_f^2\left(\dfrac{2\gamma}{M_2}-\gamma^2\right)}$ and $\eta_1\in[0,1)$ since $m_f\leqslant M_2$.

4.2. K-step PDFP-based Langevin Algorithms. This subsection discusses how to implement the PDFP based ULA and MALA to sample the distribution density (3.9). The two algorithms are based on Algorithm 2 and Algorithm 3 respectively. Recall that in Algorithms 2 and 3, an optimization subproblem

$$\operatorname{prox}_{\rho U}(\theta_n)=\arg\min_{x\in\mathbb{R}^d}\left(\frac{\|x-\theta_n\|^2}{2\rho}+f(x)+g(Bx)\right)\tag{4.12}$$

needs to be solved. The object function in Eq. (4.12) changes with respect to different θ_n. We then apply the PDFP algorithm to Equation (4.12), yielding the following iteration:

$$\begin{cases}y_{n,k+1}=x_{n,k}-\gamma\left(\nabla f(x_{n,k})+\dfrac{1}{\rho}(x_{n,k}-\theta_n)\right)-\gamma B^Tv_{n,k}\\v_{n,k+1}=\operatorname{prox}_{\frac{\lambda}{\gamma}g^*}\left(\dfrac{\lambda}{\gamma}By_{n,k+1}+v_{n,k}\right)\\x_{n,k+1}=x_{n,k}-\gamma\left(\nabla f(x_{n,k})+\dfrac{1}{\rho}(x_{n,k}-\theta_n)\right)-\gamma B^Tv_{n,k+1}.\end{cases}\tag{4.13}$$

Inserting the PDFP iteration in Eqs. (4.13) into Algorithms 2 and 3, yields Algorithms 5 (ULA-PDFP) and 6 (MALA-PDFP) respectively.

It is natural to ask why we solve Equation (4.12) by PDFP, instead of other algorithms such as FISTA [2] and Chambolle-Pock (CP) [5]. Firstly, FISTA cannot directly solve Equation (4.2) when B is not an identity matrix and solving Equation (4.12) by FISTA requires a two-layer subproblem. Secondly, solving Equation (4.12) by CP requires an additional conjugate-gradient algorithm even for $K=1$, which is inefficient when function f includes a non-trivial forward operator. When f is zero and the Moreau envelope is applied merely on g, this is what actually MYULA [13] is doing and therefore CP can solve Equation (4.12) with the conjugate-gradient algorithm analytically solved. See more details of the experiments between ULA-PDFP and MYULA-CP in Section 6.

Note here that an important feature of the proposed algorithms are that they only conduct a fixed number (i.e., K) of PDFP iterations, a key difference from the existing algorithms that requires to solve the proximal subproblem $\operatorname{prox}_{\rho U}(\theta_n)$ accurately.

Algorithm 5: ULA-PDFP

Result: $\{\theta_n\}_{n=1}^N$.

Set $\rho>0$, $\delta\in(0,\rho]$, $0<\lambda\leqslant\dfrac{1}{\lambda_{\max}(BB^T)}$, $0<\gamma<\dfrac{2}{M_2+1/\rho}$, $\theta_0\in\mathbb{R}^d$. **for** $n=0$ *to* $N-1$ **do**

Initialization: $x_{n,0}=\theta_n$, $v_{n,0}=0$. **for** $k=0$ *to* $K-1$ **do**

$$y_{n,k+1}=x_{n,k}-\gamma\left(\nabla f(x_{n,k})+\frac{1}{\rho}(x_{n,k}-\theta_n)\right)-\gamma B^T v_{n,k}$$

$$v_{n,k+1}=\operatorname{prox}_{\frac{\lambda}{\gamma}g^*}\left(\frac{\lambda}{\gamma}By_{n,k+1}+v_{n,k}\right)$$

$$x_{n,k+1}=x_{n,k}-\gamma\left(\nabla f(x_{n,k})+\frac{1}{\rho}(x_{n,k}-\theta_n)\right)-\gamma B^T v_{n,k+1}$$

$$\theta_{n+1}=\left(1-\tfrac{\delta}{\rho}\right)\theta_n+\tfrac{\delta}{\rho}x_{n,K}+\sqrt{2\delta}\xi_n,\quad \xi_n\sim\mathcal{N}(0,I)$$

Consequently $x_{n,K}$ is only an approximation of $\operatorname{prox}_{\rho U}(\theta_n)$ and Algorithm 5 is actually an ULA with inaccurate gradient. The motivation for doing this is to reduce the computational cost – as one can see each iteration needs to evaluate $\nabla f(x)$, and so the computational cost for computing $\operatorname{prox}_{\rho U}(\theta_n)$ may be exceedingly high, especially when evaluating $\nabla f(x)$ itself is time-consuming. In this case, using a small number of iterations (i.e. small value of K) may effectively reduce the computational cost. Since the approximation is used, the resulting sampling error in Algorithm 5 must be analyzed (note that the approximation does not introduce sampling error in Algorithm 6 thanks to the Metropolis step).

It should be noted that, in the iteration in Algorithms 5 and 6 we initialize the dual variable $v_{n,0}=0$ instead of $v_{n,0}=v_{n-1,K}$, different from the optimization algorithm. The reason is that the Langevin algorithms are expected to generate a Markov chain $\{\theta_n\}$, which means that the $(n+1)$-th state θ_{n+1} only depends on the n-th state θ_n and transition probability $P(\theta_{n+1}|\theta_n)$. Once the dual variable $v_{n,0}$ is initialized as $v_{n-1,K}$, it actually involves the information in the $(n-1)$-th state and the transition probability hence becomes $P(\theta_{n+1}|\theta_n,\theta_{n-1})$, violating the Markov property of sequence $\{\theta_n\}$.

Recall that, if $\operatorname{prox}_{\rho U}(\theta_n)$ is accurately evaluated, then from [9,11,12] one directly has the convergence and the upper bound on the sampling error of Algorithm 2. As has been mentioned, Algorithm 5 is actually an ULA with inaccurate gradient and so its convergence property needs to be studied. [10] considers both deterministic and stochastic approximations of the gradient of the log-density and quantifies the impact of the gradient evaluation inaccuracies. In Algorithm 5 one intuitively has better upper bound on the sampling error for larger K, but at more computational cost. The detailed error analysis is presented in Section 5. We also want to mention that, our numerical experiments illustrate that the PDFP based algorithms with small K can produce sufficiently accurate samples, with more details in Section 6.

5. Convergence results

In this section we present the convergence analysis of ULA with K-step PDFP (Algorithm 5). Most of our proofs follow from [9]. To start with, we first give a lemma which specifies the strong convexity of the Moreau envelope of a given strongly convex function.

Algorithm 6: MALA-PDFP

Result: $\{\theta_n\}_{n=1}^N$.

Set $\rho>0$, $\delta\in(0,\rho]$, $0<\lambda\leqslant\dfrac{1}{\lambda_{\max}(BB^T)}$, $0<\gamma<\dfrac{2}{M_2+1/\rho}$, $P_0=\theta_0\in\mathbb{R}^d$. **for** $n=0$ *to* $N-1$ **do**

Propose a new state: $Y_{n+1}=\left(1-\dfrac{\delta}{\rho}\right)\theta_n+\dfrac{\delta}{\rho}P_n+\sqrt{2\delta}\xi_n,\quad \xi_n\sim\mathcal{N}(0,I)$

Initialization: $x_{n,0}=Y_{n+1}$, $v_{n,0}=0$ **for** $k=0$ *to* $K-1$ **do**

$$y_{n,k+1}=x_{n,k}-\gamma\left(\nabla f(x_{n,k})+\frac{1}{\rho}(x_{n,k}-Y_{n+1})\right)-\gamma B^T v_{n,k}$$

$$v_{n,k+1}=\operatorname{prox}_{\frac{\lambda}{\gamma}g^*}\left(\frac{\lambda}{\gamma}By_{n,k+1}+v_{n,k}\right)$$

$$x_{n,k+1}=x_{n,k}-\gamma\left(\nabla f(x_{n,k})+\frac{1}{\rho}(x_{n,k}-Y_{n+1})\right)-\gamma B^T v_{n,k+1}$$

$P_{\text{tmp}}=x_{n,K}$

Compute acceptance rate: $A(Y_{n+1},\theta_n)=\min\left(1,\dfrac{\pi(Y_{n+1})}{\pi(\theta_n)}\cdot\dfrac{p(\theta_n|Y_{n+1})}{p(Y_{n+1}|\theta_n)}\right)$

$$=\min\left(1,\frac{\pi(Y_{n+1})}{\pi(\theta_n)}\cdot\frac{\exp\left(-\frac{1}{4\delta}\left\|\theta_n-\left(1-\frac{\delta}{\rho}\right)Y_{n+1}-\frac{\delta}{\rho}P_{\text{tmp}}\right\|_2^2\right)}{\exp\left(-\frac{1}{4\delta}\left\|Y_{n+1}-\left(1-\frac{\delta}{\rho}\right)\theta_n-\frac{\delta}{\rho}P_n\right\|_2^2\right)}\right)$$

Sample a from uniform distribution: $a\sim U[0,1]$.

if $a<A(Y_{n+1},\theta_n)$ **then**

Accept Y_{n+1}: $\theta_{n+1}=Y_{n+1}$, $P_{n+1}=P_{\text{tmp}}$

else

Reject Y_{n+1}: $\theta_{n+1}=\theta_n$, $P_{n+1}=P_n$

LEMMA 5.1. *Let $m,\rho\in\mathbb{R},m>0,\rho>0$. If function $h(x)$ is m-strongly convex, then the ρ-Moreau envelope of $h(x)$,*

$$h_\rho(x)=\min_y\left(h(y)+\frac{\|y-x\|_2^2}{2\rho}\right),\tag{5.1}$$

is $\dfrac{m}{1+\rho m}$-strongly convex.

Proof. Define $p(x):=h(x)-\dfrac{m}{2}\|x\|_2^2$. Then from Lemma 3.2, $p(\cdot)$ is convex. By the definition,

$$\begin{aligned}h_\rho(x)&=\min_y\left(h(y)+\frac{\|y-x\|_2^2}{2\rho}\right)\\&=\min_y\left(h(y)-\frac{m}{2}\|y\|_2^2+\frac{m}{2}\|y\|_2^2+\frac{\|y-x\|_2^2}{2\rho}\right)\\&=\min_y\left(h(y)-\frac{m}{2}\|y\|_2^2+\frac{(1+\rho m)}{2\rho}\left\|y-\frac{x}{1+\rho m}\right\|_2^2+\frac{m}{2(1+\rho m)}\|x\|_2^2\right)\end{aligned}$$

$$= \min_y \left(p(y) + \frac{(1+\rho m)}{2\rho} \left\| y - \frac{x}{1+\rho m} \right\|_2^2 \right) + \frac{m}{2(1+\rho m)} \|x\|_2^2. \tag{5.2}$$

Define $q(z) := p\left(\frac{z}{1+\rho m}\right)$. Then function $q(\cdot)$ is convex.

From the exchange of variables $y = \frac{z}{1+\rho m}$,

$$\begin{aligned} h_\rho(x) &= \min_z \left(q(z) + \frac{1}{2\rho(1+\rho m)} \|z-x\|_2^2 \right) + \frac{m}{2(1+\rho m)} \|x\|_2^2 \\ &= q_{\rho(1+\rho m)}(x) + \frac{m}{2(1+\rho m)} \|x\|_2^2. \end{aligned} \tag{5.3}$$

$h_\rho(x) - \frac{m}{2(1+\rho m)} \|x\|_2^2 = q_{\rho(1+\rho m)}(x)$ is the Moreau envelope of q, hence convex. From Lemma 3.2, $h_\rho(x)$ is $\frac{m}{1+\rho m}$-strongly convex. ∎

From Lemma 3.5, one can see that when $\rho \to 0$, $h_\rho(x) \to h(x)$ pointwisely and this result is consistent with $\frac{m}{1+\rho m} \to m$. When $\rho \to +\infty$, $h_\rho(x)$ tends to a constant function and $\frac{m}{1+\rho m} \to 0$. In the later convergence analysis of Algorithm 5 when we require the strong convexity of $U_\rho(x)$, the strong convexity of $U(x)$ is sufficient.

For the study of ULA with inaccurate gradient of log-density, [10] gives an upper bound of the sampling error when the inaccuracies of the gradients have bounded expectations and variances, with the assumption that U is strongly convex. Actually the convergence of PDFP (Algorithm 4) and convergence of ULA with accurate gradients do not require the strong convexity of U. To prove the boundness of the samples generated by Algorithm 5, we need the same assumption that U is strongly convex. In this case, we assume that f is m-strongly convex and therefore U_ρ is $\frac{m}{1+\rho m}$-strongly convex from Lemma 5.1.

Another assumption we make is the boundedness of $\left\| \mathrm{prox}_{\frac{\lambda}{\gamma} g^*}(v) \right\|_2$. This is true when g is the L^1 norm and g^* is an indicator function of a bounded convex set.

Since the PDFP iteration and the optimal primal and dual solution of problem (4.12) change with different θ_n, we simplify the notation by denoting the PDFP iteration of problem (4.12) in Algorithm 5 as

$$\begin{aligned} & v_{n,k+1} = T_{n,1}(v_{n,k}, x_{n,k}), \quad x_{n,k+1} = T_{n,2}(v_{n,k}, x_{n,k}) \\ & \Rightarrow (v_{n,k+1}, x_{n,k+1}) = T_n(v_{n,k}, v_{n,k}) := (T_{n,1}(v_{n,k}, x_{n,k}), T_{n,2}(v_{n,k}, x_{n,k})). \end{aligned} \tag{5.4}$$

From this notation, the iteration (4.13) and Algorithm 5 turns into

$$\begin{cases} x_{n,0} = \theta_n, \quad v_{n,0} = 0 \\ v_{n,K} = T_{n,1} T_n^{K-1}(v_{n,0}, x_{n,0}) \\ x_{n,K} = T_{n,2} T_n^{K-1}(v_{n,0}, x_{n,0}) \\ \theta_{n+1} = \theta_n - \frac{\delta}{\rho}(\theta_n - x_{n,K}) + \sqrt{2\delta}\xi_n, \quad \xi_n \sim \mathcal{N}(0, I). \end{cases} \tag{5.5}$$

With a K-step PDFP iteration, Algorithm 5 and (5.5) evaluate the gradient $\nabla U_\rho(\theta_n)$ by the approximation $\frac{\theta_n - T_{n,2} T_n^{K-1}(0, \theta_n)}{\rho}$, leading to the error

$$\nabla U_\rho(\theta_n) - \frac{\theta_n - T_{n,2} T_n^{K-1}(0, \theta_n)}{\rho} = \frac{T_{n,2} T_n^{K-1}(0, \theta_n) - \mathrm{prox}_{\rho U}(\theta_n)}{\rho}. \tag{5.6}$$

Since the function $\frac{\|x-\theta_n\|^2}{2\rho}+f(x)$ is always strongly convex even if f is not strongly convex, we then give a lemma which quantifies the error of K-step PDFP in Algorithm 5:

LEMMA 5.2. *Assume that $\{\theta_n\}_n$ is the sequence generated by Algorithm 5. Assume that ρ,K,λ,γ are the parameters in Algorithm 5. Let $m\geqslant 0$, $m\in\mathbb{R}$. Assume that f is m-strongly convex and $\rho_{\min}(BB^T)>0$. If g is a function such that, $\forall v\in V$, $\left\|\mathrm{prox}_{\frac{\lambda}{\gamma}g^*}(v)\right\|_2\leqslant C$, then $\forall n\in\mathbb{N}$,*

$$\left\|\frac{T_{n,2}T_n^{K-1}(0,\theta_n)-\mathrm{prox}_{\rho U}(\theta_n)}{\rho}\right\|_2^2\leqslant\eta^K\left(\|\nabla U_\rho(\theta_n)\|_2^2+\frac{\gamma^2C^2}{\lambda\rho^2}\right), \tag{5.7}$$

where

$$\eta=\max\left(1-\left(m+\frac{1}{\rho}\right)^2\left(\frac{2\gamma}{M_2+\frac{1}{\rho}}-\gamma^2\right),1-\lambda\rho_{\min}(BB^T)\right). \tag{5.8}$$

Proof. See Appendix A.2. □

To obtain the convergence analysis of Algorithm 5, we use the same proof technique as in [9,11] to first obtain some upper bound of $\mathbb{E}(U_\rho(\theta_{n+1})-U_\rho(\theta_n))$. To be more specific, we respectively give the bound of $\mathbb{E}\left(U_\rho\left(\theta_n-\frac{\delta}{\rho}(\theta_n-x_{n,K})+\sqrt{2\delta}\xi_n\right)-U_\rho\left(\theta_n-\frac{\delta}{\rho}(\theta_n-x_{n,K})\right)\right)$ and $\mathbb{E}\left(U_\rho(\theta_n-\frac{\delta}{\rho}(\theta_n-x_{n,K})-U_\rho(\theta_n)\right)$, which both simply make use of the Lipschitz gradient of U_ρ. Those are explained by Lemma 5.3 and Lemma 5.4.

LEMMA 5.3. *$\forall x\in\mathbb{R}^d$, if $\xi\sim\mathcal{N}(0,I)$ is independent of x, then*

$$\mathbb{E}\left(U_\rho(x+\sqrt{2\delta}\xi)-U_\rho(x)\right)\leqslant\frac{\delta d}{\rho}. \tag{5.9}$$

Proof. From Lemma 3.5 (2), U_ρ has $\frac{1}{\rho}$-Lipschitz gradient, then by Lemma 3.4,

$$U_\rho(x+\sqrt{2\delta}\xi)-U_\rho(x)\leqslant\left\langle\nabla U_\rho(x),\sqrt{2\delta}\xi\right\rangle+\frac{1}{2\rho}\left\|\sqrt{2\delta}\xi\right\|_2^2. \tag{5.10}$$

From the assumption that $\xi\sim\mathcal{N}(0,I)$ is independent of x, we have $\mathbb{E}\left\langle\nabla U_\rho(x),\sqrt{2\delta}\xi\right\rangle=0$.

Then

$$\mathbb{E}\left(U_\rho(x+\sqrt{2\delta}\xi)-U_\rho(x)\right)\leqslant\frac{1}{2\rho}\mathbb{E}\left\|\sqrt{2\delta}\xi\right\|_2^2=\frac{\delta d}{\rho}. \tag{5.11}$$

□

LEMMA 5.4. *$\forall x\in\mathbb{R}^d$, $\forall v\in V$,*

$$\begin{aligned}U_\rho\left(x-\frac{\delta}{\rho}\left(x-T_{n,2}T_n^{K-1}(v,x)\right)\right)-U_\rho(x)&\leqslant-\delta\left(1-\frac{\delta}{2\rho}\right)\|\nabla U_\rho(x)\|_2^2\\&+\frac{\delta^2}{2\rho}\left\|\frac{T_{n,2}T_n^{K-1}(v,x)-\mathrm{prox}_{\rho U}(x)}{\rho}\right\|_2^2+\delta\left(1-\frac{\delta}{\rho}\right)\left\langle\nabla U_\rho(x),\frac{T_{n,2}T_n^{K-1}(v,x)-\mathrm{prox}_{\rho U}(x)}{\rho}\right\rangle.\end{aligned} \tag{5.12}$$

Proof. From Lemma 3.5 (2), U_ρ has $\frac{1}{\rho}$-Lipschitz gradient, then by Lemma 3.4,

$$\begin{aligned}
&U_\rho\left(x-\frac{\delta}{\rho}\left(x-T_{n,2}T_n^{K-1}(v,x)\right)\right)-U_\rho(x)\\
&\leqslant -\delta\left\langle \nabla U_\rho(x),\frac{x-T_{n,2}T_n^{K-1}(v,x)}{\rho}\right\rangle+\frac{\delta^2}{2\rho}\left\|\frac{x-T_{n,2}T_n^{K-1}(v,x)}{\rho}\right\|_2^2\\
&=-\delta\left\langle \nabla U_\rho(x),\nabla U_\rho(x)-\frac{T_{n,2}T_n^{K-1}(v,x)-\mathrm{prox}_{\rho U}(x)}{\rho}\right\rangle\\
&\quad+\frac{\delta^2}{2\rho}\left\|\nabla U_\rho(x)-\frac{T_{n,2}T_n^{K-1}(v,x)-\mathrm{prox}_{\rho U}(x)}{\rho}\right\|_2^2\\
&=-\delta\left(1-\frac{\delta}{2\rho}\right)\|\nabla U_\rho(x)\|_2^2+\frac{\delta^2}{2\rho}\left\|\frac{T_{n,2}T_n^{K-1}(v,x)-\mathrm{prox}_{\rho U}(x)}{\rho}\right\|_2^2\\
&\quad+\delta\left(1-\frac{\delta}{\rho}\right)\left\langle \nabla U_\rho(x),\frac{T_{n,2}T_n^{K-1}(v,x)-\mathrm{prox}_{\rho U}(x)}{\rho}\right\rangle. \qquad (5.13)
\end{aligned}$$

∎

From Lemma 5.3 and Lemma 5.4 we can deduce the following lemma showing that the upper bound of $\mathbb{E}(U_\rho(\theta_{n+1})-U_\rho(\theta_n))$ can be controlled by $\mathbb{E}\|\nabla U_\rho(\theta_n)\|_2^2$.

LEMMA 5.5. *Assume that $\{\theta_n\}_n$ is the sequence generated by Algorithm 5. Assume that x^* is the optimal solution of problem (4.2). Assume that $\delta,\rho,K,\lambda,\gamma$ are the parameters in Algorithm 5. Let $m\geqslant 0$, $m\in\mathbb{R}$. Assume that f is m-strongly convex and $\rho_{\min}(BB^T)>0$. If g is a function such that, $\forall v\in V$, $\left\|\mathrm{prox}_{\frac{\lambda}{\gamma}g^*}(v)\right\|_2\leqslant C$, then $\forall n\in\mathbb{N}$,*

$$\mathbb{E}(U_\rho(\theta_{n+1})-U_\rho(\theta_n))\leqslant -\frac{\delta}{2}(1-\eta^K)\mathbb{E}\|\nabla U_\rho(\theta_n)\|_2^2+\frac{2\delta d\lambda\rho+\delta\gamma^2C^2\eta^K}{2\lambda\rho^2}, \qquad (5.14)$$

where

$$\eta=\max\left(1-\left(m+\frac{1}{\rho}\right)^2\left(\frac{2\gamma}{M_2+\frac{1}{\rho}}-\gamma^2\right),1-\lambda\rho_{\min}(BB^T)\right). \qquad (5.15)$$

Proof. See Appendix A.3. ∎

In the above lemma, whether $m=0$ or $m>0$ simply makes a difference in η. If $m>0$ we can further deduce the boundness of $\mathbb{E}U_\rho(\theta_n)$ and $\mathbb{E}\|\theta_n-x^*\|_2^2$, by the following theorem:

THEOREM 5.1. *Under the conditions in Lemma 5.5, if $m>0$, then $\forall n\in\mathbb{N}$,*

$$\mathbb{E}(U_\rho(\theta_n)-U_\rho(x^*))\leqslant\left(1-m_\rho\delta(1-\eta^K)\right)^n\mathbb{E}(U_\rho(\theta_0)-U_\rho(x^*))+\frac{2d\lambda\rho+\gamma^2C^2\eta^K}{2\lambda\rho^2m_\rho(1-\eta^K)}, \qquad (5.16)$$

where

$$m_\rho=\frac{m}{1+\rho m},\quad \eta=\max\left(1-\left(m+\frac{1}{\rho}\right)^2\left(\frac{2\gamma}{M_2+\frac{1}{\rho}}-\gamma^2\right),1-\lambda\rho_{\min}(BB^T)\right). \qquad (5.17)$$

Proof. See Appendix A.4. ∎

By simple computation we have that $\eta \in [0,1)$. Since K is the number of iterations in subproblems and is independent of η, when $K \to +\infty$, we have that $\eta^K \to 0$.

Thus the gradients $\nabla U_\rho(\theta_n)$ are almost accurate and the inequality (5.16) is reduced to

$$\mathbb{E}(U_\rho(\theta_n) - U_\rho(x^*)) \leqslant (1 - m_\rho\delta)^n \mathbb{E}(U_\rho(\theta_0) - U_\rho(x^*)) + \frac{d}{\rho m_\rho}, \tag{5.18}$$

which matches Proposition 1 in [9]. This lemma implies that the upper bound of $\mathbb{E}U_\rho(\theta_n)$ includes a term not depending on the discretization parameter δ and another term approaching to zero as $n \to +\infty$. Moreover, we can also obtain the upper bound of $\mathbb{E}\|\theta_n - x^*\|_2^2$ by the m_ρ-strongly convexity of U_ρ and $\frac{m_\rho}{2}\|x - x^*\|_2^2 \leqslant U_\rho(x) - U_\rho(x^*)$. Both the boundness of $\mathbb{E}U_\rho(\theta_n)$ and $\mathbb{E}\left(\|\theta_n - x^*\|_2^2\right)$ essentially require the strong convexity of U_ρ.

Theorem 5.1 shows that for any $K \in \mathbb{N}$, Algorithm 5 will not blow up in the sense of expectation. The remaining portion of this section will complete the nonasymptotic error analysis of the sampling. We now present a lemma quantifying the accummulated gradients of the log-density and the accummulated errors:

LEMMA 5.6. *Under the conditions in Lemma 5.5, we have*

$$\begin{aligned}
&\delta \sum_{n=0}^{N-1} \mathbb{E}\|\nabla U_\rho(\theta_n)\|_2^2 \leqslant \frac{2}{1-\eta^K}\mathbb{E}(U_\rho(\theta_0) - U_\rho(x^*)) + \frac{N\delta\left(2d\lambda\rho + \gamma^2 C^2 \eta^K\right)}{\lambda\rho^2(1-\eta^K)}\\
&\delta \sum_{n=0}^{N-1} \mathbb{E}\left\|\frac{T_{n,2}T_n^{K-1}(0,\theta_n) - \operatorname{prox}_{\rho U}(\theta_n)}{\rho}\right\|_2^2 \leqslant \frac{2\eta^K}{1-\eta^K}\mathbb{E}(U_\rho(\theta_0) - U_\rho(x^*)) + \frac{N\delta\eta^K\left(2d\lambda\rho + \gamma^2C^2\right)}{\lambda\rho^2(1-\eta^K)}\\
&\delta \sum_{n=0}^{N-1} \mathbb{E}\left\|\frac{\theta_n - T_{n,2}T_n^{K-1}(0,\theta_n)}{\rho}\right\|_2^2 \leqslant \frac{4(1+\eta^K)}{1-\eta^K}\mathbb{E}(U_\rho(\theta_0) - U_\rho(x^*)) + \frac{4N\delta\left(d\lambda\rho(1+\eta^K) + \gamma^2C^2\eta^K\right)}{\lambda\rho^2(1-\eta^K)},
\end{aligned} \tag{5.19}$$

where

$$\eta = \max\left(1 - \left(m + \frac{1}{\rho}\right)^2\left(\frac{2\gamma}{M_2 + \frac{1}{\rho}} - \gamma^2\right), 1 - \lambda\rho_{\min}(BB^T)\right). \tag{5.20}$$

Proof. See Appendix A.5. ◻

Assume that $\{\mathbf{L}_t^\rho, t \geqslant 0\}$ is the solution of Langevin diffusion (3.11). For a fixed time interval $[0,l]$ where $l = N\delta$, Lemma 5.6 shows an upper bound of $\delta\sum_{n=0}^{N-1}\mathbb{E}\|\nabla U_\rho(\theta_n)\|_2^2$ when $N \to +\infty$. For the sampling error analysis we aim to prove that the solution $L_l^\rho \to \pi_\rho$ as $l \to +\infty$, and then with fixed l the distribution of the N-th sample θ_N can be arbitrarily close to L_l^ρ as $N \to +\infty$ and $K \to +\infty$.

For the samples $\{\theta_n\}_{n=0}^N$ generated by Algorithm 5, we introduce a continuous time Markov process $\{\mathbf{D}_t : t \geqslant 0\}$ such that the distributions of $(\theta_0, \theta_1, \dots, \theta_N)$ and $(\mathbf{D}_0, \mathbf{D}_\delta, \dots, \mathbf{D}_{N\delta})$ coincide. The process $\{\mathbf{D}_t : t \geqslant 0\}$ is defined as the solution of the stochastic differential equation

$$\mathrm{d}\mathbf{D}_t = \mathbf{b}_t(\mathbf{D_t})\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{W}_t, \quad t \geqslant 0, \mathbf{D}_0 = \theta_0, \tag{5.21}$$

$$\mathbf{b}_t(\mathbf{D}_t) = \sum_{n=0}^{\infty} \frac{T_{n,2}T_n^{K-1}(0, \mathbf{D}_{n\delta}) - \mathbf{D}_{n\delta}}{\rho} \mathbb{1}_{[n\delta,(n+1)\delta]}(t), \tag{5.22}$$

where $T_n(v,x)$ and $T_{n,2}(v,x)$ are defined by (5.4).

THEOREM 5.2. *If the continuous time Markov process* $\{\mathbf{D}_t : t \geqslant 0\}$ *is defined by (5.21, 5.22), then* $(\theta_0, \theta_1, \ldots, \theta_N)$ *has the same distribution as* $(\mathbf{D}_0, \mathbf{D}_\delta, \ldots, \mathbf{D}_{N\delta})$.

Proof. We prove this by induction.

For $n \in \mathbb{Z}$, assume that $\mathbf{D}_{n\delta}$ and θ_n have the same distribution. By (5.21, 5.22) we have

$$\begin{aligned}
\mathbf{D}_{(n+1)\delta} &= \mathbf{D}_{n\delta} + \int_{n\delta}^{(n+1)\delta} b_\tau(\mathbf{D}_\tau)\mathrm{d}\tau + \int_{n\delta}^{(n+1)\delta} \sqrt{2}\mathrm{d}\mathbf{W}_\tau \\
&= \mathbf{D}_{n\delta} + \int_{n\delta}^{(n+1)\delta} \sum_{k=0}^{\infty} \frac{T_{n,2}T_n^{K-1}(0,\mathbf{D}_{k\delta}) - \mathbf{D}_{k\delta}}{\rho} \mathbb{1}_{[k\delta,(k+1)\delta]}(\tau)\mathrm{d}\tau + \sqrt{2\delta}\xi_n \\
&= \mathbf{D}_{n\delta} + \frac{\delta}{\rho}\left(T_{n,2}T_n^{K-1}(0,\mathbf{D}_{n\delta}) - \mathbf{D}_{n\delta}\right) + \sqrt{2\delta}\xi_n \\
&= \left(1 - \frac{\delta}{\rho}\right)\mathbf{D}_{n\delta} + \frac{\delta}{\rho}T_{n,2}T_n^{K-1}(0,\mathbf{D}_{n\delta}) + \sqrt{2\delta}\xi_n.
\end{aligned} \tag{5.23}$$

Compare (5.23) with (5.5) and we can deduce that $\mathbf{D}_{(n+1)\delta}$ and θ_{n+1} have the same distribution. By induction we complete the proof. ∎

Now we have a continuous time Markov process $\{\mathbf{D}_t : t \geqslant 0\}$. To obtain the KL distance between the distributions of the processes $\{\mathbf{L}^\rho : t \in [0, N\delta]\}$ and $\{\mathbf{D} : t \in [0, N\delta]\}$ we use a lemma from [9] based on the Girsanov formula:

LEMMA 5.7. *If for some* $B > 0$ *the non-anticipative drift function* $\mathbf{b} : C(\mathbb{R}_+, \mathbb{R}^d) \times \mathbb{R}_+ \to \mathbb{R}^d$ *satisfies the inequality* $\|\mathbf{b}(\mathbf{D}, t)\|_2 \leqslant B(1 + \|\mathbf{D}\|_\infty)$ *for every* $t \in [0, N\delta]$ *and every* $\mathbf{D} \in C(\mathbb{R}_+, \mathbb{R}^d)$, *then the Kullback-Leibler divergence between* $\mathbb{P}_{\mathbf{L}^\rho}^{\mathbf{x},N\delta}$ *and* $\mathbb{P}_{\mathbf{D}}^{\mathbf{x},N\delta}$, *the distributions of the processes* $\{\mathbf{L}^\rho : t \in [0, N\delta]\}$ *and* $\{\mathbf{D} : t \in [0, N\delta]\}$ *with the initial value* $\mathbf{L}_0^\rho = \mathbf{D}_0 = \mathbf{x}$, *is given by*

$$\mathrm{KL}\left(\mathbb{P}_{\mathbf{L}^\rho}^{\mathbf{x},N\delta} \| \mathbb{P}_{\mathbf{D}}^{\mathbf{x},N\delta}\right) \leqslant \frac{1}{4}\int_0^{N\delta} \mathbb{E}\left[\|\nabla U_\rho(\mathbf{D}_t) + \mathbf{b}_t(\mathbf{D}_t)\|_2^2\right]\mathrm{d}t. \tag{5.24}$$

Using Lemma 5.7 we can prove the following theorem which gives an upper bound of the KL divergence:

THEOREM 5.3. *Let* $l = N\delta$ *be fixed. Assume that* $\mathbf{D}$ *is defined by (5.21, 5.22). Suppose that all the conditions of Lemma 5.5 and Lemma 5.7 are satisfied, then*

$$\begin{aligned}
\mathrm{KL}\left(\mathbb{P}_{\mathbf{L}^\rho}^{\mathbf{x},l} \| \mathbb{P}_{\mathbf{D}}^{\mathbf{x},l}\right) &\leqslant \frac{2\delta^2(1+\eta^K) + 3\rho^2\eta^K}{3\rho^2(1-\eta^K)}\mathbb{E}(U_\rho(x) - U_\rho(x^*)) \\
&+ \frac{ld\lambda\rho\left(4\delta^2(1+\eta^K) + 3\delta\rho(1-\eta^K) + 6\rho^2\eta^K\right) + l\gamma^2C^2\eta^K(4\delta^2 + 3\rho^2)}{6\lambda\rho^4(1-\eta^K)},
\end{aligned} \tag{5.25}$$

where

$$\eta = \max\left(1 - \left(m + \frac{1}{\rho}\right)^2\left(\frac{2\gamma}{M_2 + \frac{1}{\rho}} - \gamma^2\right), 1 - \lambda\rho_{\min}(BB^T)\right). \tag{5.26}$$

Proof. See Appendix A.6. ∎

Given fixed ρ, this upper bound of $\mathrm{KL}\left(\mathbb{P}_{\mathbf{L}^\rho}^{\mathbf{x},l} \| \mathbb{P}_{\mathbf{D}}^{\mathbf{x},l}\right)$ tends to 0 as $\delta \to 0$ and $K \to +\infty$. Meanwhile, this upper bound also partly depends on the initial sample $\mathbf{D}_0 = \theta_0 = x$. Up to now, we have no detailed assumption on θ_0. If θ_0 is drawn from the initial distribution ν, from Lemma 3.5 and lemma 1 in [9] one can deduce the following lemma:

LEMMA 5.8. *$m\in\mathbb{R}, m>0$. If U is m-strongly convex and $m_\rho=\frac{m}{1+\rho m}$, then for any initial probability density ν we have*

$$\left\|\nu\mathbf{P}^t_{\mathbf{L}^\rho}-\pi_\rho\right\|_{\mathrm{TV}}\leqslant\frac{1}{2}\chi^2(\nu\|\pi_\rho)^{1/2}\exp\left(\frac{-tm_\rho}{2}\right),\quad \forall t\geqslant 0.$$

Proof. See lemma 1 in [9]. ∎

We can prove the following lemma when the initial distribution ν is a Gaussian distribution $\mathcal{N}_d(x^*,\rho\mathbf{I}_d)$ with mean x^*.

LEMMA 5.9. *$m\in\mathbb{R}, m>0$. Assume that x^* is the optimal solution of problem (4.2). If U is m-strongly convex and $m_\rho=\frac{m}{1+\rho m}$, if ν is the density of the Gaussian distribution $\mathcal{N}_d(x^*,\rho\mathbf{I}_d)$, then we have*

$$\int_{\mathbb{R}^d}\frac{\nu^2(x)}{\pi_\rho(x)}\mathrm{d}x\leqslant\frac{1}{(\rho m_\rho)^{\frac{d}{2}}}.$$

Proof. The proof follows the same pattern of lemma 5 in [9].
From (3.9), Lemma 3.5 and Lemma 3.4,

$$\begin{aligned}
&\pi_\rho(x)^{-1}=\exp\{U_\rho(x)\}\int_{\mathbb{R}^d}\exp\{-U_\rho(\overline{x})\}\mathrm{d}\overline{x}=\exp\{U_\rho(x)-U_\rho(x^*)\}\int_{\mathbb{R}^d}\exp\{-U_\rho(\overline{x})+U_\rho(x^*)\}\mathrm{d}\overline{x}\\
&\leqslant\exp\left\{\nabla U_\rho(x^*)^{\mathrm{T}}(x-x^*)+\frac{1}{2\rho}\|x-x^*\|_2^2\right\}\int_{\mathbb{R}^d}\exp\left\{-\nabla U_\rho(x^*)^{\mathrm{T}}(\overline{x}-x^*)-\frac{m_\rho}{2}\|\overline{x}-x^*\|_2^2\right\}\mathrm{d}\overline{x}\\
&=\left(\frac{2\pi}{m_\rho}\right)^{d/2}\exp\left(\frac{1}{2\rho}\|x-x^*\|_2^2\right),
\end{aligned}\tag{5.27}$$

then we have

$$\begin{aligned}
\int_{\mathbb{R}^d}\frac{\nu^2(x)}{\pi_\rho(x)}\mathrm{d}x&=(2\pi\rho)^{-d}\int_{\mathbb{R}^d}\exp\left\{-\frac{1}{\rho}\|x-x^*\|_2^2\right\}\pi_\rho(x)^{-1}\mathrm{d}x\\
&\leqslant(2\pi\rho)^{-d}\left(\frac{2\pi}{m_\rho}\right)^{d/2}\int_{\mathbb{R}^d}\exp\left\{-\frac{\|x-x^*\|_2^2}{2\rho}\right\}\mathrm{d}x\\
&=\frac{1}{(\rho m_\rho)^{\frac{d}{2}}}.
\end{aligned}\tag{5.28}$$

∎

In the next theorem we finally give the error analysis of the Total-Variation norm between the distribution of the N-th sample θ_n and π_ρ.

THEOREM 5.4. *Let $l=N\delta$. Assume that $\mathbf{D}$ is defined by (5.21, 5.22). Suppose that all the conditions of Lemma 5.5 and Lemma 5.7 are satisfied. Assume that ν is the Gaussian distribution $\mathcal{N}_d(x^*,\rho\mathbf{I}_d)$. If $m>0$ and $m_\rho=\frac{m}{1+\rho m}$, then the TV-norm between the distribution of the N-th sample θ_N and the distribution π_ρ satisfies*

$$\begin{aligned}
&\|\nu\mathbf{P}_{\theta_N}-\mathbf{P}_{\pi_\rho}\|_{\mathrm{TV}}\leqslant\frac{1}{2}\exp\left(-\frac{d}{4}\log(\rho m_\rho)-\frac{lm_\rho}{2}\right)\\
&\quad+\sqrt{\frac{\lambda d\left(2\delta^2\rho^2+4l\delta^2\rho+3l\delta\rho^2\right)+\eta^K\left[\lambda d\left(2\delta^2\rho^2+3\rho^4+4l\delta^2\rho-3l\delta\rho^2+6l\rho^3\right)+l\gamma^2C^2\left(4\delta^2+3\rho^2\right)\right]}{12\lambda\rho^4(1-\eta^K)}},
\end{aligned}\tag{5.29}$$

where

$$\eta = \max\left(1 - \left(m + \frac{1}{\rho}\right)^2 \left(\frac{2\gamma}{M_2 + \frac{1}{\rho}} - \gamma^2\right), 1 - \lambda\rho_{\min}(BB^T)\right). \tag{5.30}$$

Therefore for any fixed ρ, $\forall \epsilon > 0$, $\exists l > 0, \delta \in (0, \rho]$ *and* $K \in \mathbb{N}$, *such that* $\left\|\nu \mathbf{P}_{\theta_N} - \mathbf{P}_{\pi_\rho}\right\|_{\mathrm{TV}} < \epsilon$.

Proof. See Appendix A.7. ∎

This upper bound demonstrates that, in order to make the error small one first needs a long burn-in time l. While l is large enough and remains fixed, small discretization step-size δ and more iterations K will lead to a satisfactory error. This also matches Theorem 2 in [9].

Fig. 5.1: *Left: the ground truth. Middle: the blurred and noisy image. Right: the posterior mean.*

6. Numerical experiments

To demonstrate the performance of the proposed algorithms, we provide two practical examples – an image motion deblurring problem and a computerized tomography (CT) reconstruction problem. We formulate both problems in the Bayesian framework and therefore sampling their posterior distributions is the primary goal here.

Choices of K and δ: The stepsize δ should satisfy the upper bound studied in Section 5. Both δ and the number of iterations K control a tradeoff between asymptotic accuracy and convergence speed. For ULA-PDFP, using large δ and small K then the Markov chain will move quickly to its stationary regime, ignoring a larger bias. We recommend using $K=1$, since in later results $K=1$ leads to a satisfactory bias. However in the situations where a small bias is expected, one should choose a small δ and a large K, though more computation is required during the burn-in time of the Markov chains.

6.1. Image motion deblurring. In the image motion deblurring problem, suppose that we use the TV prior, and the resulting posterior distribution

$$\pi(\theta) \propto \exp\left(-\frac{\|y-A\theta\|_2^2}{2\sigma^2} - \lambda\|\nabla\theta\|_1\right),$$

where $y(t)$ is the blurred image, $\theta(t)$ is the target image that we want to reconstruct, σ^2 is the observation noise variance (assuming zero-mean Gaussian noise), λ is the regularization coefficient, and A is a linear motion blur operator in the form of

$$(A\theta)(t) = \int \theta(\tau)K(t-\tau)\mathrm{d}\tau.$$

We use three commonly used tested images: Peppers, Cameraman and Barbara (left column in Figure 5.1). In all three experiments, we choose $\sigma=0.01$ and operator A formed by the kernel K of size 10×10. The dimensionality of the unknown images and the associated regularization parameter λ values are given in Table 6.1. We use synthetic data (Middle column in Figure 5.1) generated from the ground truth images (left column in Figure 5.1). The posterior mean is used as an estimator of the original image. In this experiment we draw 10000 samples from the posterior $\pi(\theta)$. We use the following quantitative measures to assess the performance of the sampling methods. To compare the estimation error we compute the peak signal-to-noise ratio (PSNR) of the posterior sample mean, which is used as an estimator of the true image. For sampling efficiency comparison we respectively calculate the effective sample size (ESS) [20] and the expected square jumping distance (ESJD) [28] of the samples.

	pepper	cameraman	barbara
dimensionality	256×256	256×256	512×512
λ	0.13	0.12	0.08

Table 6.1: *Dimensionality and λ values.*

We first examine the unadjusted algorithms, and we restate that, without the Metropolis step, the samples obtained by this type of methods are subject to bias. Apart from the proposed PDFP based algorithm, we also implement Moreau-Yosida unadjusted Langevin algorithm (MYULA) in [13]. Note that in MYULA, it is proposed to accurately solve the subproblem by Chambolle2004 [4], and to have a more comprehensive comparison, we also implement a slightly modified version of MYULA – replacing Chambolle2004 with a K-step Chambolle-Pock [5].

We summarize the results in Table 6.2, and note that for ULA-PDFP and MYULA-CP we tested three cases $K=1,5$, and $K=100$. In particular in the $K=100$ case the subproblem is considered to be precisely solved, and in fact our numerical experiments suggest that most of the subproblems can meet the stopping criteria $\|x_{n,k+1}-x_{n,k}\| <$

$\rho = 0.01$		peppers			cameraman			barbara		
	K	PSNR	ESJD	time	PSNR	ESJD	time	PSNR	ESJD	time
ULA-PDFP	1	26.48	1311	**55s**	24.13	1311	**59s**	23.20	5243	**246s**
ULA-PDFP	5	26.50	1311	191s	24.18	1311	195s	23.20	5243	842s
ULA-PDFP	100	26.42	1311	242s	24.18	1311	267s	23.22	5243	1047s
MYULA-CP	1	26.44	1311	64s	24.17	1310	66s	23.18	5239	287s
MYULA-CP	5	26.49	1310	146s	24.11	1309	137s	23.21	5237	656s
MYULA-CP	100	26.46	1310	1097s	24.16	1310	980s	23.21	5238	4133s
MYULA	100	26.43	1310	551s	24.17	1310	525s	23.22	5238	2421s

Table 6.2: *Comparison of the unadjusted Langevin algorithms.*

	K	PSNR	ESJD	ESS	parameters	time
pepper						
MALA(subgradient)		25.58	3.9	4.03	$\delta = $ 8e-5	**93s**
PMALA-CP	1	26.05	19.4	4.09	$\rho = \delta = $ 3e-4	103s
PMALA-CP	5	26.68	439.0	4.78	$\rho = \delta = $ 7e-3	239s
PMALA-CP	100	26.70	427.7	4.75	$\rho = \delta = $ 7e-3	1218s
PMALA	100	26.69	420.9	4.76	$\rho = \delta = $ 7e-3	581s
MALA-PDFP	1	26.61	441.0	4.81	$\rho = \delta = $ 7e-3	108s
MALA-PDFP	5	26.66	439.8	4.78	$\rho = \delta = $ 7e-3	257s
MALA-PDFP	100	26.70	437.6	4.76	$\rho = \delta = $ 7e-3	295s
cameraman						
MALA(subgradient)		23.65	3.4	3.97	$\delta = $ 6e-5	**89s**
PMALA-CP	1	24.31	18.2	4.05	$\rho = \delta = $ 4e-4	107s
PMALA-CP	5	24.46	384.6	4.68	$\rho = \delta = $ 6e-3	179s
PMALA-CP	100	24.51	390.9	4.70	$\rho = \delta = $ 6e-3	877s
PMALA	100	24.54	383.3	4.65	$\rho = \delta = $ 6e-3	442s
MALA-PDFP	1	24.51	370.7	4.62	$\rho = \delta = $ 6e-3	91s
MALA-PDFP	5	24.57	384.1	4.67	$\rho = \delta = $ 6e-3	230s
MALA-PDFP	100	24.58	375.1	4.66	$\rho = \delta = $ 6e-3	234s
barbara						
MALA(subgradient)		22.09	11.3	3.99	$\delta = $ 5e-5	**338s**
PMALA-CP	1	23.11	47.4	3.96	$\rho = \delta = $ 2e-4	403s
PMALA-CP	5	23.28	1073.8	4.30	$\rho = \delta = $ 5e-3	790s
PMALA-CP	100	23.23	993.1	4.29	$\rho = \delta = $ 5e-3	3827s
PMALA	100	23.29	947.2	4.27	$\rho = \delta = $ 5e-3	1793s
MALA-PDFP	1	23.30	934.7	4.24	$\rho = \delta = $ 5e-3	387s
MALA-PDFP	5	23.24	1033.2	4.30	$\rho = \delta = $ 5e-3	978s
MALA-PDFP	100	23.28	973.4	4.28	$\rho = \delta = $ 5e-3	865s

Table 6.3: *Comparison of the Metropolis-adjusted Langevin algorithms.*

10^{-5} in less than 30 steps. For MYULA, the subproblem is solved accurately using Chambolle2004 [4]. From the table we observe that the PSNR and ESJD of the sample means calculated by all the methods are approximately the same, suggesting that all the methods can produce similar sampling results. Quite interestingly, the results show that PDFP and CP with $K=1$ can produce results of the same PSNR and ESJD as solving the subproblem accurately. On the other hand, as has been discussed, smaller K leads to less computational burden, which is supported by the time cost shown in the table. Also ULA-PDFP with $K=1$ seems to be the most efficient one in terms of

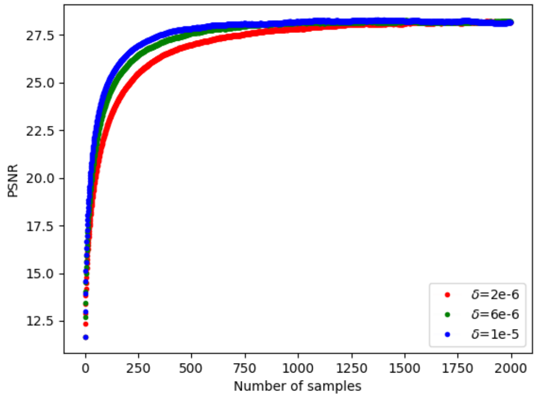

Fig. 6.1: *PSNR of the samples (ULA-PDFP) in the burn-in period.*

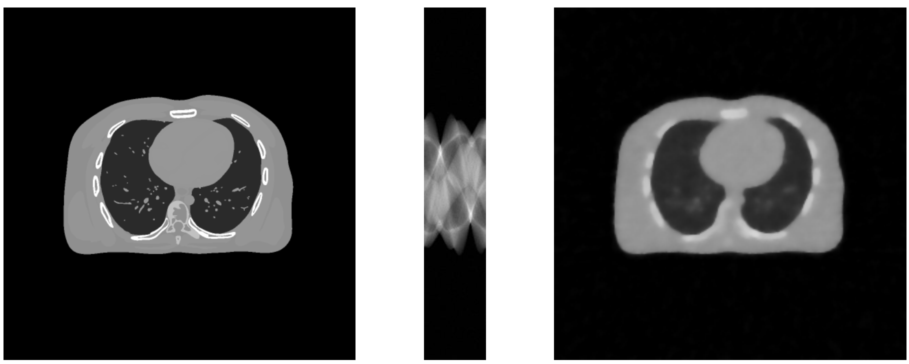

Fig. 6.2: *Left: Original image (unknown). Mid: Observation* (512×90)*. Right: Reconstructed image (posterior mean).*

time cost. In summary, the results suggest that while all the algorithms yield similar sampling performance, those that do not seek to solve the subproblem accurately are significantly more computationally efficient.

Next we test the algorithms with the additional Metropolis (accept-reject) step included. More precisely we implement the following algorithms: MALA with subgradient, the PMALA method in [29], a variant of PMALA with Chambolle2004 replaced by K-step Chambolle-Pock, and the proposed PDFP based algorithm denoted as MALA-PDFP. The results of all the methods are compared in Table 6.3, and we reinstate that thanks to the Metropolis step, the samples are asymptotically unbiased. For the stability of PMALA and MALA-PDFP, step size δ should be no larger than parameter ρ. Following [29] we fix $\delta=\rho$ and their values (that are shown in Table 6.3) are chosen such that the acceptance rates of all the algorithms are around 50% [31, 33] for fair comparison. First we have found that MALA with subgradient clearly has the worst performance among all the methods, a finding agreeing with [29]. Moreover, in both MALA-PDFP and PMALA-CP, we can see that the results of $K=5$ are rather close to those of $K=100$ and PMALA where in both cases the subproblem is solved accurately.

Notably in Table 6.3 the run time of MALA-PDFP for $K=100$ is similar or less than that for $K=5$, this is because in these experiments ρ is much smaller than Table 6.2 and the stopping criteria $\|x_{n,k+1}-x_{n,k}\|<10^{-5}$ is met even if $k<5$. More interestingly, however, PMALA-CP with $K=1$ yields substantially worse results (in terms of ESS and ESJD) than the algorithms that solve the subprobem accurately, while MALA-PDFP with $K=1$ produces results that are comparable to those. While this is an interesting indicator that the 1-step MALA-PDFP may be an effective and efficient sampling algorithm, further investigation and more comprehensive tests of the method are needed.

	PSNR	K	time
ULA-PDFP	29.22	1	**72s**
ULA-PDFP	29.26	5	275s
ULA-PDFP	29.26	100	1513s
MYULA-CP	29.25	1	80s
MYULA-CP	29.26	5	166s
MYULA-CP	29.24	100	720s
MYULA	29.26	100	1615s

Table 6.4: *Comparison of the unadjusted Langevin algorithms.*

	PSNR	ESJD	ESS mean	parameters	K	time
PMALA-CP	28.30	5.6e-4	4.03	$\rho=\delta=$ 1.0e-8	1	**98s**
PMALA-CP	28.51	5.8e-3	4.87	$\rho=\delta=$ 1.0e-7	5	197s
PMALA-CP	28.89	4.6e-2	8.44	$\rho=\delta=$ 8.0e-7	100	579s
PMALA	28.91	4.8e-2	8.63	$\rho=\delta=$ 8.0e-7	100	305s
MALA-PDFP	28.86	4.9e-2	8.75	$\rho=\delta=$ 8.0e-7	1	120s
MALA-PDFP	28.85	4.8e-2	8.67	$\rho=\delta=$ 8.0e-7	5	326s
MALA-PDFP	28.86	4.8e-2	8.66	$\rho=\delta=$ 8.0e-7	100	352s

Table 6.5: *Comparison of the Metropolis-adjusted Langevin algorithms.*

6.2. Computed tomography reconstruction of medical image. In this section we consider the computed tomography (CT) reconstruction problem with the posterior distribution

$$\pi(\theta)\propto\exp\left(-\frac{\|y-A\theta\|_2^2}{2\sigma^2}-\lambda\|\nabla\theta\|_1\right),$$

where $\theta\in\mathbb{R}^{256\times256}$ is the unknown XCAT phantom image and $y\in\mathbb{R}^{512\times90}$ is the projection observed. The range of y is about $[0,5.0]^{512\times90}$. The observation noise is assumed to be additive white Gaussian noise with standard variance $\sigma=0.5$ and λ is taken to be 50. The operator A is the Radon transform which can be efficiently computed by a parallelizable algorithm in [17] using fan-beam geometry, but still very time-consuming that less calls of A will significantly reduce the time cost. In this experiment the number of detectors is 512 and that of the viewers is 90 defining a highly ill-posed problem.

Before the Markov chain reaches its stable regime, the burn-in time takes less than 2000 samples. Smaller stepsize δ leads to longer burn-in period as shown in Figure 6.1.

After the burn-in time, we draw 10000 samples from the posterior π with the same set of unadjusted algorithms in the first example, and show the results in Table 6.4. In all the algorithms we use $\rho = 10^{-5}$. The results in these examples are largely consistent with those reported in the first example: all the methods produce similar results in terms of PSNR while those with small K are more computationally efficient. Next we test the Metropolis-adjusted algorithms – again by drawing 10,000 samples from the posterior, and the results are shown in Table 6.5. Once again the parameter values are chosen so that the acceptance probability is around 50% [31,33]. We observe that in this example the 1-step MALA-PDFP has similar performance as the algorithms that solve the subproblem accurately, while 1-step PMALA-CP is clearly less efficient in terms of both ESS and ESJD, supporting our results in the first example.

7. Conclusion

Langevin algorithms are important tools for sampling posterior distributions in Bayesian inference. Since the gradient information is typically needed in the Langevin algorithms, it is particularly challenging to apply them to non-smooth distributions. In this work we consider the class of methods where one solves a proximity subproblem in each iteration. In particular we propose to solve the proximity subproblem with the PDFP algorithm, and more importantly the method only seeks to find an approximate solution of the subproblem by conducting a (small) fixed number of PDFP iterations. We provide error analysis of the approximate PDFP based algorithms. Our numerical experiments also suggest that the 1-step PDFP based algorithms, especially the Metropolis-adjusted version, yields a good performance, in terms of sampling efficiency and computation time.

Appendix. Proofs.

A.1. Lemma 4.2.

Proof. From the fixed point property by Lemma 4.1 we know

$$\begin{cases} v^* = \text{prox}_{\frac{\lambda}{\gamma}g^*}\left(\frac{\lambda}{\gamma}B(x^* - \gamma\nabla f(x^*)) + (I - \lambda BB^T)v^*\right) = \text{prox}_{\frac{\lambda}{\gamma}g^*}\left(\frac{\lambda}{\gamma}B\phi(x^*) + Mv^*\right) \\ x^* = x^* - \gamma\nabla f(x^*) - \gamma B^T v^*. \end{cases} \tag{A.1}$$

Let $x_k, x_{k+1}, v_k, v_{k+1}$ be the variables in Algorithm 4, then

$$\begin{aligned} \|v_{k+1} - v^*\|_2^2 &= \left\|\text{prox}_{\frac{\lambda}{\gamma}g^*}\left(\frac{\lambda}{\gamma}B\phi(x_k) + Mv_k\right) - \text{prox}_{\frac{\lambda}{\gamma}g^*}\left(\frac{\lambda}{\gamma}B\phi(x^*) + Mv^*\right)\right\|_2^2 \\ &\leqslant \left\langle \text{prox}_{\frac{\lambda}{\gamma}g^*}\left(\frac{\lambda}{\gamma}B\phi(x_k) + Mv_k\right) - \text{prox}_{\frac{\lambda}{\gamma}g^*}\left(\frac{\lambda}{\gamma}B\phi(x^*) + Mv^*\right), \right. \\ &\qquad \left. \frac{\lambda}{\gamma}B\phi(x_k) + Mv_k - \frac{\lambda}{\gamma}B\phi(x^*) - Mv^* \right\rangle \\ &= \frac{\lambda}{\gamma}\langle v_{k+1} - v^*, B(\phi(x_k) - \phi(x^*))\rangle + \langle v_{k+1} - v^*, M(v_k - v^*)\rangle. \end{aligned} \tag{A.2}$$

The inequality follows from the firm nonexpansiveness of $\text{prox}_{\frac{\lambda}{\gamma}g^*}(\cdot)$ (Definition 3.2). By the definition of x_{k+1} in Algorithm 4,

$$\begin{aligned} \|x_{k+1} - x^*\|_2^2 &= \left\|\phi(x_k) - \phi(x^*) - \gamma B^T(v_{k+1} - v^*)\right\|_2^2 \\ &= \|\phi(x_k) - \phi(x^*)\|_2^2 - 2\gamma\left\langle \phi(x_k) - \phi(x^*), B^T(v_{k+1} - v^*)\right\rangle + \frac{\gamma^2}{\lambda^2}\left\|\lambda B^T(v_{k+1} - v^*)\right\|_2^2 \end{aligned}$$

$$=\|\phi(x_k)-\phi(x^*)\|_2^2-2\gamma\left\langle\phi(x_k)-\phi(x^*),B^T(v_{k+1}-v^*)\right\rangle+\frac{\gamma^2}{\lambda}\|v_{k+1}-v^*\|_2^2$$
$$-\frac{\gamma^2}{\lambda}\|v_{k+1}-v^*\|_M^2. \tag{A.3}$$

Here the last equality follows from the definition $M=I-\lambda BB^T$ and $\|z\|_M:=\sqrt{\langle z,Mz\rangle}$.

Combine (A.2) with (A.3),

$$\begin{aligned}
&\|x_{k+1}-x^*\|_2^2+\frac{\gamma^2}{\lambda}\|v_{k+1}-v^*\|_2^2\\
=&\|\phi(x_k)-\phi(x^*)\|_2^2-2\gamma\left\langle\phi(x_k)-\phi(x^*),B^T(v_{k+1}-v^*)\right\rangle+\frac{2\gamma^2}{\lambda}\|v_{k+1}-v^*\|_2^2\\
&-\frac{\gamma^2}{\lambda}\|v_{k+1}-v^*\|_M^2\\
\leqslant&\|\phi(x_k)-\phi(x^*)\|_2^2+\frac{2\gamma^2}{\lambda}\langle v_{k+1}-v^*,M(v_k-v^*)\rangle-\frac{\gamma^2}{\lambda}\|v_{k+1}-v^*\|_M^2\\
&-2\gamma\left\langle\phi(x_k)-\phi(x^*),B^T(v_{k+1}-v^*)\right\rangle+2\gamma\langle v_{k+1}-v^*,B(\phi(x_k)-\phi(x^*))\rangle\\
=&\|\phi(x_k)-\phi(x^*)\|_2^2+\frac{2\gamma^2}{\lambda}\langle v_{k+1}-v^*,M(v_k-v^*)\rangle-\frac{\gamma^2}{\lambda}\|v_{k+1}-v^*\|_M^2\\
=&\|\phi(x_k)-\phi(x^*)\|_2^2+\frac{\gamma^2}{\lambda}\|v_k-v^*\|_M^2-\frac{\gamma^2}{\lambda}\|v_{k+1}-v_k\|_M^2\\
\leqslant&\eta_1^2\|x_k-x^*\|_2^2+\frac{\gamma^2}{\lambda}\|v_k-v^*\|_M^2\\
\leqslant&\eta_1^2\|x_k-x^*\|_2^2+\frac{\gamma^2}{\lambda}(1-\lambda\rho_{\min}(BB^T))\|v_k-v^*\|_2^2.
\end{aligned}\tag{A.4}$$

The first inequality uses (A.2). The second inequality follows from the condition that $\|\phi(x)-\phi(y)\|_2\leqslant\eta_1\|x-y\|_2$, $\forall x,y\in\mathbb{R}^d$. The last inequality uses the fact that $0<\lambda\leqslant\frac{1}{\rho_{\max}(BB^T)}$ and $0\preceq M\preceq(1-\lambda\rho_{\min}(BB^T))I$. From the definition $\eta:=\max\left(\eta_1^2,1-\lambda\rho_{\min}(BB^T)\right)$, obviously $0\leqslant\eta<1$ since $\eta_1^2<1$ and $0\leqslant1-\lambda\rho_{\min}(BB^T)<1$. Then from (A.4),

$$\begin{aligned}
&\|x_{k+1}-x^*\|_2^2+\frac{\gamma^2}{\lambda}\|v_{k+1}-v^*\|_2^2\leqslant\eta\left(\|x_k-x^*\|_2^2+\frac{\gamma^2}{\lambda}\|v_k-v^*\|_2^2\right)\\
\Rightarrow&\|x_k-x^*\|_2^2+\frac{\gamma^2}{\lambda}\|v_k-v^*\|_2^2\leqslant\eta^k\left(\|x_0-x^*\|_2^2+\frac{\gamma^2}{\lambda}\|v_0-v^*\|_2^2\right).
\end{aligned}\tag{A.5}$$

□

A.2. Lemma 5.2.

Proof. Denote the optimal primal and dual solutions of the problem (4.12) by x_n^* and v_n^*, exactly $x_n^*=\operatorname{prox}_{\rho U}(\theta_n)$. Since $\frac{\|x-\theta_n\|^2}{2\rho}+f(x)$ is $\left(m+\frac{1}{\rho}\right)$-strongly convex with a $\left(M_2+\frac{1}{\rho}\right)$-Lipschitz gradient, by Lemma 4.2 we have

$$\|x_{n,k}-x_n^*\|_2^2+\frac{\gamma^2}{\lambda}\|v_{n,k}-v_n^*\|_2^2\leqslant\eta^k\left(\|x_{n,0}-x_n^*\|_2^2+\frac{\gamma^2}{\lambda}\|v_{n,0}-v_n^*\|_2^2\right), \tag{A.6}$$

where

$$\eta=\max\left(1-\left(m+\frac{1}{\rho}\right)^2\left(\frac{2\gamma}{M_2+\frac{1}{\rho}}-\gamma^2\right),1-\lambda\rho_{\min}(BB^T)\right)$$

$$\geqslant \max\left(1-\left(\frac{1+\rho m}{1+\rho M_2}\right)^2, 1-\lambda\rho_{\min}(BB^T)\right). \tag{A.7}$$

Therefore we get

$$\begin{aligned}
&\left\|T_{n,2}T_n^{K-1}(0,\theta_n)-\mathrm{prox}_{\rho U}(\theta_n)\right\|_2^2=\left\|x_{n,K}-\mathrm{prox}_{\rho U}(\theta_n)\right\|_2^2\\
\leqslant&\eta^K\left(\left\|x_{n,0}-\mathrm{prox}_{\rho U}(\theta_n)\right\|_2^2+\frac{\gamma^2}{\lambda}\|v_{n,0}-v_n^*\|_2^2\right)\\
=&\eta^K\left(\left\|\theta_n-\mathrm{prox}_{\rho U}(\theta_n)\right\|_2^2+\frac{\gamma^2}{\lambda}\|v_n^*\|_2^2\right)\\
\leqslant&\eta^K\left(\left\|\theta_n-\mathrm{prox}_{\rho U}(\theta_n)\right\|_2^2+\frac{\gamma^2C^2}{\lambda}\right).
\end{aligned} \tag{A.8}$$

The second inequality follows from the fixed point lemma 4.1 applied on problem (4.12) that

$$v_n^*=\mathrm{prox}_{\frac{\lambda}{\gamma}g^*}\left(\frac{\lambda}{\gamma}B\left(x_n^*-\gamma\left(\nabla f(x_n^*)+\frac{1}{\rho}(x_n^*-\theta_n)\right)-\gamma B^Tv_n^*\right)+v_n^*\right). \tag{A.9}$$

Then from Lemma 3.5 (4),

$$\begin{aligned}
&\left\|\frac{T_{n,2}T_n^{K-1}(0,\theta_n)-\mathrm{prox}_{\rho U}(\theta_n)}{\rho}\right\|_2^2\leqslant\eta^K\left(\left\|\frac{\theta_n-\mathrm{prox}_{\rho U}(\theta_n)}{\rho}\right\|_2^2+\frac{\gamma^2C^2}{\lambda\rho^2}\right)\\
=&\eta^K\left(\|\nabla U_\rho(\theta_n)\|_2^2+\frac{\gamma^2C^2}{\lambda\rho^2}\right).
\end{aligned} \tag{A.10}$$

□

A.3. Lemma 5.5.

Proof. From Lemma 5.4, we have

$$\begin{aligned}
&U_\rho\left(x-\frac{\delta}{\rho}\left(x-T_{n,2}T_n^{K-1}(v,x)\right)\right)-U_\rho(x)\leqslant-\delta\left(1-\frac{\delta}{2\rho}\right)\|\nabla U_\rho(x)\|_2^2\\
&+\frac{\delta^2}{2\rho}\left\|\frac{T_{n,2}T_n^{K-1}(v,x)-\mathrm{prox}_{\rho U}(x)}{\rho}\right\|_2^2+\delta\left(1-\frac{\delta}{\rho}\right)\left\langle\nabla U_\rho(x),\frac{T_{n,2}T_n^{K-1}(v,x)-\mathrm{prox}_{\rho U}(x)}{\rho}\right\rangle\\
\leqslant&-\frac{\delta}{2}\|\nabla U_\rho(x)\|_2^2+\frac{\delta}{2}\left\|\frac{T_{n,2}T_n^{K-1}(v,x)-\mathrm{prox}_{\rho U}(x)}{\rho}\right\|_2^2.
\end{aligned} \tag{A.11}$$

The second inequality follows from Cauchy-Schwarz inequality since $\delta\in(0,\rho]$.

By (5.5), (A.11) and Lemma 5.3,

$$\begin{aligned}
&\mathbb{E}(U_\rho(\theta_{n+1})-U_\rho(\theta_n))=\mathbb{E}\left(U_\rho\left(\theta_n-\frac{\delta}{\rho}\left(\theta_n-T_{n,2}T_n^{K-1}(0,\theta_n)\right)+\sqrt{2\delta}\xi_n\right)-U_\rho(\theta_n)\right)\\
\leqslant&\frac{\delta d}{\rho}-\frac{\delta}{2}\mathbb{E}\|\nabla U_\rho(\theta_n)\|_2^2+\frac{\delta}{2}\mathbb{E}\left\|\frac{T_{n,2}T_n^{K-1}(0,\theta_n)-\mathrm{prox}_{\rho U}(\theta_n)}{\rho}\right\|_2^2.
\end{aligned} \tag{A.12}$$

From Lemma 5.2 and (A.12),

$$\begin{aligned}
\mathbb{E}(U_\rho(\theta_{n+1})-U_\rho(\theta_n))&\leqslant\frac{\delta d}{\rho}-\frac{\delta}{2}(1-\eta^K)\mathbb{E}\|\nabla U_\rho(\theta_n)\|_2^2+\frac{\delta\gamma^2C^2\eta^K}{2\lambda\rho^2}\\
&=-\frac{\delta}{2}(1-\eta^K)\mathbb{E}\|\nabla U_\rho(\theta_n)\|_2^2+\frac{2\delta d\lambda\rho+\delta\gamma^2C^2\eta^K}{2\lambda\rho^2},
\end{aligned} \tag{A.13}$$

where

$$\eta=\max\left(1-\left(m+\frac{1}{\rho}\right)^2\left(\frac{2\gamma}{M_2+\frac{1}{\rho}}-\gamma^2\right),1-\lambda\rho_{\min}(BB^T)\right). \tag{A.14}$$

□

A.4. Theorem 5.1.

Proof. From Lemma 3.5 (3), U_ρ and U have the same minimizer x^*. Therefore $\nabla U_\rho(x^*)=0$.

Let $m_\rho=\dfrac{m}{1+\rho m}$. Since U_ρ is m_ρ-strongly convex by Lemma 5.1, it is well known [3] that,

$$\|\nabla U_\rho(x)\|_2^2\geqslant 2m_\rho(U_\rho(x)-U_\rho(x^*)),\quad \forall x\in\mathbb{R}^d. \tag{A.15}$$

Together with Lemma 5.5, $\forall n\in\mathbb{N}$,

$$\begin{aligned}
&\mathbb{E}(U_\rho(\theta_{n+1})-U_\rho(x^*))\leqslant\left(1-m_\rho\delta(1-\eta^K)\right)\mathbb{E}(U_\rho(\theta_n)-U_\rho(x^*))+\frac{2\delta d\lambda\rho+\delta\gamma^2C^2\eta^K}{2\lambda\rho^2}\\
&\Rightarrow\mathbb{E}(U_\rho(\theta_n)-U_\rho(x^*))\\
&\leqslant\left(1-m_\rho\delta(1-\eta^K)\right)^n\mathbb{E}(U_\rho(\theta_0)-U_\rho(x^*))+\frac{2\delta d\lambda\rho+\delta\gamma^2C^2\eta^K}{2\lambda\rho^2}\frac{1-(1-m_\rho\delta(1-\eta^K))^n}{1-(1-m_\rho\delta(1-\eta^K))}\\
&\leqslant\left(1-m_\rho\delta(1-\eta^K)\right)^n\mathbb{E}(U_\rho(\theta_0)-U_\rho(x^*))+\frac{2d\lambda\rho+\gamma^2C^2\eta^K}{2\lambda\rho^2m_\rho(1-\eta^K)}.
\end{aligned} \tag{A.16}$$

□

A.5. Lemma 5.6.

Proof. From Lemma 5.5,

$$\frac{\delta}{2}(1-\eta^K)\mathbb{E}\|\nabla U_\rho(\theta_n)\|_2^2\leqslant\mathbb{E}(U_\rho(\theta_n)-U_\rho(\theta_{n+1}))+\frac{2\delta d\lambda\rho+\delta\gamma^2C^2\eta^K}{2\lambda\rho^2},\quad n\in\mathbb{N}. \tag{A.17}$$

Summing the inequalities for $n=0,1,\dots,N-1$, we have

$$\begin{aligned}
\frac{\delta}{2}(1-\eta^K)\sum_{n=0}^{N-1}\mathbb{E}\|\nabla U_\rho(\theta_n)\|_2^2&\leqslant\mathbb{E}(U_\rho(\theta_0)-U_\rho(\theta_N))+\frac{N\delta\left(2d\lambda\rho+\gamma^2C^2\eta^K\right)}{2\lambda\rho^2}\\
&\leqslant\mathbb{E}(U_\rho(\theta_0)-U_\rho(x^*))+\frac{N\delta\left(2d\lambda\rho+\gamma^2C^2\eta^K\right)}{2\lambda\rho^2}\\
\Rightarrow\delta\sum_{n=0}^{N-1}\mathbb{E}\|\nabla U_\rho(\theta_n)\|_2^2&\leqslant\frac{2}{1-\eta^K}\mathbb{E}(U_\rho(\theta_0)-U_\rho(x^*))+\frac{N\delta\left(2d\lambda\rho+\gamma^2C^2\eta^K\right)}{\lambda\rho^2(1-\eta^K)}.
\end{aligned} \tag{A.18}$$

Then from Lemma 5.2,

$$\begin{aligned}
&\delta\sum_{n=0}^{N-1}\mathbb{E}\left\|\frac{T_{n,2}T_n^{K-1}(0,\theta_n)-\mathrm{prox}_{\rho U}(\theta_n)}{\rho}\right\|_2^2\leqslant\eta^K\delta\sum_{n=0}^{N-1}\mathbb{E}\|\nabla U_\rho(\theta_n)\|_2^2+\frac{N\delta\gamma^2C^2\eta^K}{\lambda\rho^2}\\
&\leqslant\frac{2\eta^K}{1-\eta^K}\mathbb{E}(U_\rho(\theta_0)-U_\rho(x^*))+\frac{N\delta\eta^K\left(2d\lambda\rho+\gamma^2C^2\eta^K\right)}{\lambda\rho^2(1-\eta^K)}+\frac{N\delta\gamma^2C^2\eta^K}{\lambda\rho^2}
\end{aligned}$$

$$= \frac{2\eta^K}{1-\eta^K}\mathbb{E}(U_\rho(\theta_0)-U_\rho(x^*)) + \frac{N\delta\eta^K\left(2d\lambda\rho+\gamma^2C^2\right)}{\lambda\rho^2(1-\eta^K)}. \tag{A.19}$$

Using Cauchy-Schwarz inequality,

$$\delta\sum_{n=0}^{N-1}\mathbb{E}\left\|\frac{\theta_n - T_{n,2}T_n^{K-1}(0,\theta_n)}{\rho}\right\|_2^2 = \delta\sum_{n=0}^{N-1}\mathbb{E}\left\|\nabla U_\rho(\theta_n) - \frac{T_{n,2}T_n^{K-1}(0,\theta_n)-\mathrm{prox}_{\rho U}(\theta_n)}{\rho}\right\|_2^2$$

$$\leqslant 2\delta\sum_{n=0}^{N-1}\mathbb{E}\|\nabla U_\rho(\theta_n)\|_2^2 + 2\delta\sum_{n=0}^{N-1}\mathbb{E}\left\|\frac{T_{n,2}T_n^{K-1}(0,\theta_n)-\mathrm{prox}_{\rho U}(\theta_n)}{\rho}\right\|_2^2$$

$$\leqslant \frac{4(1+\eta^K)}{1-\eta^K}\mathbb{E}(U_\rho(\theta_0)-U_\rho(x^*)) + \frac{4N\delta\left(d\lambda\rho(1+\eta^K)+\gamma^2C^2\eta^K\right)}{\lambda\rho^2(1-\eta^K)}. \tag{A.20}$$

□

A.6. Theorem 5.3.

Proof. According to Lemma 5.7,

$$\mathrm{KL}\left(\mathbb{P}^{\mathbf{x},l}_{\mathbf{L}^\rho}\|\mathbb{P}^{\mathbf{x},l}_{\mathbf{D}}\right) \leqslant \frac{1}{4}\sum_{n=0}^{N-1}\int_{n\delta}^{(n+1)\delta}\mathbb{E}\left\|\nabla U_\rho(\mathbf{D}_t) + \frac{T_{n,2}T_n^{K-1}(0,\mathbf{D}_{n\delta})-\mathbf{D}_{n\delta}}{\rho}\right\|_2^2 \mathrm{d}t$$

$$= \frac{1}{4}\sum_{n=0}^{N-1}\int_{n\delta}^{(n+1)\delta}\mathbb{E}\left\|\nabla U_\rho(\mathbf{D}_t) - \nabla U_\rho(\mathbf{D}_{n\delta}) + \nabla U_\rho(\mathbf{D}_{n\delta}) + \frac{T_{n,2}T_n^{K-1}(0,\mathbf{D}_{n\delta})-\mathbf{D}_{n\delta}}{\rho}\right\|_2^2 \mathrm{d}t$$

$$\leqslant \frac{1}{2}\sum_{n=0}^{N-1}\int_{n\delta}^{(n+1)\delta}\mathbb{E}\left[\|\nabla U_\rho(\mathbf{D}_t) - \nabla U_\rho(\mathbf{D}_{n\delta})\|_2^2 + \left\|\nabla U_\rho(\mathbf{D}_{n\delta}) + \frac{T_{n,2}T_n^{K-1}(0,\mathbf{D}_{n\delta})-\mathbf{D}_{n\delta}}{\rho}\right\|_2^2\right]\mathrm{d}t$$

$$= \frac{1}{2}\sum_{n=0}^{N-1}\int_{n\delta}^{(n+1)\delta}\mathbb{E}\left[\|\nabla U_\rho(\mathbf{D}_t) - \nabla U_\rho(\mathbf{D}_{n\delta})\|_2^2 + \left\|\frac{T_{n,2}T_n^{K-1}(0,\theta_n)-\mathrm{prox}_{\rho U}(\theta_n)}{\rho}\right\|_2^2\right]\mathrm{d}t. \tag{A.21}$$

The last inequality follows from Cauchy-Schwarz inequality.

From Lemma 3.5 (2), U_ρ has $\frac{1}{\rho}$-Lipschitz gradient:

$$\|\nabla U_\rho(\mathbf{D}_t) - \nabla U_\rho(\mathbf{D}_{n\delta})\|_2 \leqslant \frac{1}{\rho}\|\mathbf{D}_t - \mathbf{D}_{n\delta}\|_2. \tag{A.22}$$

Then

$$\frac{1}{2}\sum_{n=0}^{N-1}\int_{n\delta}^{(n+1)\delta}\mathbb{E}\|\nabla U_\rho(\mathbf{D}_t) - \nabla U_\rho(\mathbf{D}_{n\delta})\|_2^2\mathrm{d}t \leqslant \frac{1}{2\rho^2}\sum_{n=0}^{N-1}\int_{n\delta}^{(n+1)\delta}\mathbb{E}\|\mathbf{D}_t - \mathbf{D}_{n\delta}\|_2^2\mathrm{d}t. \tag{A.23}$$

From the definition of $\mathbf{D}_t$, for $t\in[n\delta,(n+1)\delta]$,

$$\begin{aligned}\mathbf{D}_t - \mathbf{D}_{n\delta} &= \int_{n\delta}^t b_\tau(\mathbf{D}_\tau)\mathrm{d}\tau + \int_{n\delta}^t \sqrt{2}\mathrm{d}\mathbf{W}_\tau \\ &= \frac{T_{n,2}T_n^{K-1}(0,\mathbf{D}_{n\delta})-\mathbf{D}_{n\delta}}{\rho}\int_{n\delta}^t \mathbb{1}_{[n\delta,(n+1)\delta]}(\tau)\mathrm{d}\tau + \sqrt{2}(\mathbf{W}_t - \mathbf{W}_{n\delta}) \\ &= \frac{T_{n,2}T_n^{K-1}(0,\mathbf{D}_{n\delta})-\mathbf{D}_{n\delta}}{\rho}(t-n\delta) + \sqrt{2}(\mathbf{W}_t - \mathbf{W}_{n\delta}).\end{aligned} \tag{A.24}$$

Then

$$\begin{aligned}
&\frac{1}{2\rho^2}\sum_{n=0}^{N-1}\int_{n\delta}^{(n+1)\delta}\mathbb{E}\|\mathbf{D}_t-\mathbf{D}_{n\delta}\|_2^2\mathrm{d}t\\
&=\frac{1}{2\rho^2}\sum_{n=0}^{N-1}\int_{n\delta}^{(n+1)\delta}\mathbb{E}\left(\left\|\frac{T_{n,2}T_n^{K-1}(0,\mathbf{D}_{n\delta})-\mathbf{D}_{n\delta}}{\rho}(t-n\delta)\right\|_2^2+\left\|\sqrt{2}(\mathbf{W}_t-\mathbf{W}_{n\delta})\right\|_2^2\right)\mathrm{d}t\\
&=\frac{1}{2\rho^2}\sum_{n=0}^{N-1}\left(\frac{\delta^3}{3}\mathbb{E}\left\|\frac{T_{n,2}T_n^{K-1}(0,\mathbf{D}_{n\delta})-\mathbf{D}_{n\delta}}{\rho}\right\|_2^2+\delta^2 d\right)\\
&=\frac{\delta^3}{6\rho^2}\sum_{n=0}^{N-1}\mathbb{E}\left\|\frac{T_{n,2}T_n^{K-1}(0,\theta_n)-\theta_n}{\rho}\right\|_2^2+\frac{\delta l d}{2\rho^2}.
\end{aligned}\tag{A.25}$$

Combine (A.21) with (A.23, A.25), we have

$$\begin{aligned}
&\mathrm{KL}\left(\mathbb{P}^{\mathbf{x},l}_{\mathbf{L}^\rho}\|\mathbb{P}^{\mathbf{x},l}_{\mathbf{D}}\right)\leqslant\frac{\delta^3}{6\rho^2}\sum_{n=0}^{N-1}\mathbb{E}\left\|\frac{T_{n,2}T_n^{K-1}(0,\theta_n)-\theta_n}{\rho}\right\|_2^2+\frac{\delta l d}{2\rho^2}\\
&\quad+\frac{1}{2}\sum_{n=0}^{N-1}\int_{n\delta}^{(n+1)\delta}\mathbb{E}\left\|\frac{T_{n,2}T_n^{K-1}(0,\theta_n)-\mathrm{prox}_{\rho U}(\theta_n)}{\rho}\right\|_2^2\mathrm{d}t\\
&=\frac{\delta^3}{6\rho^2}\sum_{n=0}^{N-1}\mathbb{E}\left\|\frac{T_{n,2}T_n^{K-1}(0,\theta_n)-\theta_n}{\rho}\right\|_2^2+\frac{\delta l d}{2\rho^2}+\frac{\delta}{2}\sum_{n=0}^{N-1}\mathbb{E}\left\|\frac{T_{n,2}T_n^{K-1}(0,\theta_n)-\mathrm{prox}_{\rho U}(\theta_n)}{\rho}\right\|_2^2.
\end{aligned}\tag{A.26}$$

Combined with Lemma 5.6, we obtain the inequality

$$\begin{aligned}
\mathrm{KL}\left(\mathbb{P}^{\mathbf{x},l}_{\mathbf{L}^\rho}\|\mathbb{P}^{\mathbf{x},l}_{\mathbf{D}}\right)&\leqslant\frac{2\delta^2(1+\eta^K)+3\rho^2\eta^K}{3\rho^2(1-\eta^K)}\mathbb{E}(U_\rho(x)-U_\rho(x^*))\\
&+\frac{ld\lambda\rho\left(4\delta^2(1+\eta^K)+3\delta\rho(1-\eta^K)+6\rho^2\eta^K\right)+l\gamma^2C^2\eta^K(4\delta^2+3\rho^2)}{6\lambda\rho^4(1-\eta^K)}.
\end{aligned}\tag{A.27}$$

∎

A.7. Theorem 5.4.

Proof. From triangular inequality we have

$$\left\|\nu\mathbf{P}_{\theta_N}-\mathbf{P}_{\pi_\rho}\right\|_{\mathrm{TV}}=\left\|\nu\mathbf{P}_{\mathbf{D}}^{N\delta}-\mathbf{P}_{\pi_\rho}\right\|_{\mathrm{TV}}\leqslant\left\|\nu\mathbf{P}^l_{\mathbf{L}^\rho}-\mathbf{P}_{\pi_\rho}\right\|_{\mathrm{TV}}+\left\|\nu\mathbf{P}^l_{\mathbf{D}}-\nu\mathbf{P}^l_{\mathbf{L}^\rho}\right\|_{\mathrm{TV}}.\tag{A.28}$$

From Lemma 5.8 and Lemma 5.9,

$$\left\|\nu\mathbf{P}^l_{\mathbf{L}^\rho}-\mathbf{P}_{\pi_\rho}\right\|_{\mathrm{TV}}\leqslant\frac{1}{2}\chi^2(\nu\|\pi_\rho)^{1/2}\exp\left(\frac{-lm_\rho}{2}\right)\leqslant\frac{1}{2}\exp\left(-\frac{d}{4}\log(\rho m_\rho)-\frac{lm_\rho}{2}\right).\tag{A.29}$$

By Pinsker inequality,

$$\left\|\nu\mathbf{P}^l_{\mathbf{D}}-\nu\mathbf{P}^l_{\mathbf{L}^\rho}\right\|_{\mathrm{TV}}\leqslant\left\|\nu\mathbb{P}^l_{\mathbf{D}}-\nu\mathbb{P}^l_{\mathbf{L}^\rho}\right\|_{\mathrm{TV}}\leqslant\sqrt{\frac{1}{2}\mathrm{KL}\left(\nu\mathbb{P}^l_{\mathbf{L}^\rho}\|\nu\mathbb{P}^l_{\mathbf{D}}\right)}.\tag{A.30}$$

By Lemma 5.3 and Lemma 3.4,

$$\begin{aligned}
&\mathrm{KL}\left(\nu\mathbb{P}^l_{\mathbf{L}^\rho}\|\nu\mathbb{P}^l_{\mathbf{D}}\right)\\
&\leqslant\frac{2\delta^2(1+\eta^K)+3\rho^2\eta^K}{3\rho^2(1-\eta^K)}\frac{d}{2}+\frac{ld\lambda\rho\left(4\delta^2(1+\eta^K)+3\delta\rho(1-\eta^K)+6\rho^2\eta^K\right)+l\gamma^2C^2\eta^K(4\delta^2+3\rho^2)}{6\lambda\rho^4(1-\eta^K)}\\
&=\frac{\lambda d\left(2\delta^2\rho^2+4l\delta^2\rho+3l\delta\rho^2\right)+\eta^K\left[\lambda d\left(2\delta^2\rho^2+3\rho^4+4l\delta^2\rho-3l\delta\rho^2+6l\rho^3\right)+l\gamma^2C^2\left(4\delta^2+3\rho^2\right)\right]}{6\lambda\rho^4(1-\eta^K)}.
\end{aligned}\tag{A.31}$$

From (A.28, A.29, A.30) and above,

$$\left\|\nu\mathbf{P}_{\theta_N}-\mathbf{P}_{\pi_\rho}\right\|_{\mathrm{TV}}\leqslant\frac{1}{2}\exp\left(-\frac{d}{4}\log(\rho m_\rho)-\frac{lm_\rho}{2}\right)+\sqrt{\frac{\lambda d\left(2\delta^2\rho^2+4l\delta^2\rho+3l\delta\rho^2\right)+\eta^K\left[\lambda d\left(2\delta^2\rho^2+3\rho^4+4l\delta^2\rho-3l\delta\rho^2+6l\rho^3\right)+l\gamma^2C^2\left(4\delta^2+3\rho^2\right)\right]}{12\lambda\rho^4(1-\eta^K)}},\tag{A.32}$$

where

$$\eta=\max\left(1-\left(m+\frac{1}{\rho}\right)^2\left(\frac{2\gamma}{M_2+\frac{1}{\rho}}-\gamma^2\right),1-\lambda\rho_{\min}(BB^T)\right).\tag{A.33}$$

□

REFERENCES

[1] H.H. Bauschke and P.L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, 408, 2011. 3.1, 3.2

[2] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2(1):183–202, 2009. 4.2

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004. 3.1, A.4

[4] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging. Vis., 20(1):89–97, 2004. 3.2, 6.1, 6.1

[5] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging. Vis., 40(1):120–145, 2011. 4.1, 4.2, 6.1

[6] N. Chatterji, J. Diakonikolas, M.I. Jordan, and P. Bartlett, *Langevin Monte Carlo without smoothness*, International Conference on Artificial Intelligence and Statistics, 1716–1726, 2020. 1

[7] P. Chen, J. Huang, and X. Zhang, *A primal–dual fixed point algorithm for convex separable minimization with applications to image restoration*, Inverse Probl., 29(2):025011, 2013. 1, 4.1, 4.1, 4.1, 4.1

[8] P.L. Combettes and V.R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul., 4(4):1168–1200, 2005. 3.2

[9] A.S. Dalalyan, *Theoretical guarantees for approximate sampling from smooth and log-concave densities*, J. R. Stat. Soc. Ser. B Stat. Methodol., 79(3):651–676, 2017. 2, 3.2, 3.2, 4.2, 5, 5, 5, 5, 5, 5, 5, 5

[10] A.S. Dalalyan and A. Karagulyan, *User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient*, Stoch. Process. Appl., 129(12):5278–5311, 2019. 2, 4.2, 5

[11] A. Durmus, S. Majewski, and B. Miasojedow, *Analysis of Langevin Monte Carlo via convex optimization*, J. Mach. Learn. Res., 20(1):2666–2711, 2019. 2, 4.2, 5

[12] A. Durmus and E. Moulines, *Nonasymptotic convergence analysis for the unadjusted Langevin algorithm*, Ann. Appl. Probab., 27(3):1551–1587, 2017. 2, 4.2

[13] A. Durmus, E. Moulines, and M. Pereyra, *Efficient Bayesian computation by proximal Markov chain Monte Carlo: when Langevin meets Moreau*, SIAM J. Imaging Sci., 11(1):473–506, 2018. 1, 3, 3.2, 4.2, 6.1

[14] R. Dwivedi, Y. Chen, M.J. Wainwright, and B. Yu, *Log-concave sampling: Metropolis-hastings algorithms are fast*, J. Mach. Learn. Res., 20:1–42, 2019. 1

[15] D.L. Ermak, *A computer simulation of charged particles in solution. I. Technique and equilibrium properties*, J. Chem. Phys., 62(10):4189–4196, 1975. 1

[16] E. Esser, X. Zhang, and T.F. Chan, *A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science*, SIAM J. Imaging Sci., 3(4):1015–1046, 2010. 4.1

[17] H. Gao, *Fast parallel algorithms for the x-ray transform and its adjoint*, Medical Phys., 39(11):7110–7120, 2012. 6.2

[18] W.R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, CRC Press, 1995. 1

[19] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Springer, 2006. 1

[20] R.E. Kass, B.P. Carlin, A. Gelman, and R.M. Neal, *Markov chain Monte Carlo in practice: a roundtable discussion*, Am. Stat., 52(2):93–100, 1998. 6.1

[21] T.T.-K. Lau and H. Liu, *Bregman proximal Langevin Monte Carlo via Bregman-Moreau envelopes*, Int. Conf. Mach. Learn., 12049–12077, 2022. 1

[22] J. Lehec, *The Langevin Monte Carlo algorithm in the non-smooth log-concave case*, Ann. Appl. Probab., 33(6A):4858–4874, 2023. 1

[23] T.D. Luu, J. Fadili, and C. Chesneau, *Sampling from non-smooth distributions through Langevin diffusion*, Methodol. Comput. Appl. Probab., 23(4):1173–1201, 2021. 1

[24] J.-J. Moreau, *Proximityé and dualité in a Hilbertian space*, Bull. Math. Soc. France, 93:273–299, 1965. 3.2

[25] W. Mou, N. Flammarion, M.J. Wainwright, and P.L. Bartlett, *An efficient sampling algorithm for non-smooth composite potentials*, J. Mach. Learn. Res., 23(233):1–50, 2022. 1

[26] R. Neal, *Bayesian learning via stochastic dynamics*, Adv. Neur. Inf. Process. Syst., 5:475–482, 1992. 1

[27] G. Parisi, *Correlation functions and computer simulations*, Nuclear Phys. B, 180(3):378–384, 1981. 1

[28] C. Pasarica and A. Gelman, *Adaptively scaling the Metropolis algorithm using expected squared jumped distance*, Statist. Sinica, 343–364, 2010. 6.1

[29] M. Pereyra, *Proximal Markov chain Monte Carlo algorithms*, Stat. Comput., 26(4):745–760, 2016. 1, 3, 3.2, 3.2, 3.2, 3.2, 3.2, 6.1

[30] M. Pereyra, L.V. Mieles, and K.C. Zygalakis, *Accelerating proximal Markov chain Monte Carlo by using an explicit stabilized method*, SIAM J. Imaging Sci., 13(2):905–935, 2020. 3.2, 3.2

[31] C.P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, 2, 1999. 1, 2, 3.2, 6.1, 6.2

[32] G.O. Roberts and J.S. Rosenthal, *Optimal scaling of discrete approximations to Langevin diffusions*, J. R. Stat. Soc. Ser. B Stat. Methodol., 60(1):255–268, 1998. 1, 2, 3.2

[33] G.O. Roberts and O. Stramer, *Langevin diffusions and Metropolis-Hastings algorithms*, Methodol. Comput. Appl. Probab., 4(4):337–357, 2002. 1, 2, 3.2, 6.1, 6.2

[34] G.O. Roberts and R.L. Tweedie, *Exponential convergence of Langevin distributions and their discrete approximations*, Bernoulli, 2(4):341–363, 1996. 1, 3.2

[35] A. Salim, D. Kovalev, and P. Richtárik, *Stochastic proximal Langevin algorithm: Potential splitting and nonasymptotic rates*, Adv. Neur. Inf. Process. Syst., 32:6653–6664, 2019. 1

[36] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, SIAM, 2005. 1

[37] T. Xifara, C. Sherlock, S. Livingstone, S. Byrne, and M. Girolami, *Langevin diffusions and the Metropolis-adjusted Langevin algorithm*, Stat. Probab. Lett., 91:14–19, 2014. 1

[38] Y.-N. Zhu and X. Zhang, *Stochastic primal dual fixed point method for composite optimization*, J. Sci. Comput., 84(1):1–25, 2020. 1