HESSIAN-FREE HIGH-RESOLUTION NESTEROV ACCELERATION FOR SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

It is known (Shi et al., 2021) that Nesterov's Accelerated Gradient (NAG) for optimization differs from its continuous time limit (noiseless kinetic Langevin) when its stepsize becomes finite. This work explores the sampling counterpart of this phenonemon and proposes an accelerated-gradient-based MCMC method, based on the optimizer of NAG for strongly convex functions (NAG-SC): we reformulate NAG-SC as a Hessian-Free High-Resolution ODE, change its high-resolution coefficient to a hyperparameter, inject appropriate noise, and discretize the resulting diffusion process. Accelerated sampling enabled by the new hyperparameter is quantified and it is not a false acceleration created by time-rescaling. At continuous-time level, additional acceleration over underdamped Langevin in W_2 distance is proved. At discrete algorithm level, a dedicated discretization is proposed to simulate the Hessian-Free High-Resolution SDE in a cost-efficient manner. For log-strong-concave-and-smooth target measures, the proposed algorithm achieves $\tilde{\mathcal{O}}(\sqrt{d}/\epsilon)$ iteration complexity in W_2 distance, same as underdamped Langevin dynamics, but with a reduced constant. Empirical experiments are conducted to numerically verify our theoretical results.

1 INTRODUCTION

Optimization is a major machinery that drives both the theory and practice of machine learning in recent years. Since the seminal work of Nesterov (1983), acceleration has played a key role in gradient-based optimization methods. A notable example is Nesterov's Accelerated Gradient (NAG), which is an instance of a more general family of "momentum methods". NAG in fact consists of multiple methods, including NAG-C and NAG-SC, respectively for convex and strongly convex functions. Both of them provably converge faster than vanilla gradient descent (GD) in their corresponding setups (Nesterov, 1983; 2013). Newer perspectives of acceleration continue to be revealed, e.g., Su et al. (2014); Wibisono et al. (2016); Wilson et al. (2021); Hu & Lessard (2017); Attouch et al. (2018); Shi et al. (2021), many based on the interplay between continuous and discrete times. This work aims at turning NAG-SC into a sampler based on this interplay.

In fact, approaches for sampling statistical distributions, such as gradient-based Markov Chain Monte Carlo (MCMC) methods, are also of great importance in machine learning, for example due to their links to statistical inference and abilities to represent uncertainties lacking in optimization-based methods. Although not entirely the same thing, optimization and sampling are closely related: besides seeing a large class of sampling dynamics as optimization dynamics with additional noise, viewing sampling as optimization in probability space is another profound perspective that led to fruitful discoveries (e.g., Jordan et al. (1998); Liu & Wang (2016); Dalalyan (2017a); Wibisono (2018); Zhang et al. (2018); Frogner & Poggio (2020); Chizat & Bach (2018); Chen et al. (2018a); Ma et al. (2021); Erdogdu & Hosseinzadeh (2021)). In fact, an unadjusted Euler-Maruyama discretization of overdamped Langevin dynamics (abbreviated as OLD here) is commonly considered as the analog of GD in sampling (although many other discretizations are also possible), and often referred to as Unadjusted Langevin Algorithm (ULA) (Roberts et al., 1996) and/or Langevin Monte Carlo (LMC). The convergence properties of the continuous dynamics of OLD, as well as asymptotic and non-asymptotic analyses of its discretizations have been extensively studied (e.g., Roberts et al. (1996); Villani (2008); Pavliotis (2014); Dalalyan (2017b); Durmus & Moulines (2016); Dalalyan (2017a); Durmus et al. (2019a;b); Vempala & Wibisono (2019); Cheng & Bartlett (2018); Dwivedi et al. (2019); Ma et al. (2019); Chewi et al. (2021); Erdogdu & Hosseinzadeh (2021)).

Meanwhile, the notion of acceleration is less quantified in sampling compared to that in optimization, although attention has been rapidly building up. Along this direction, one line is based on diffusion processes such as underdamped Langevin dynamics (ULD). For example, the convergence and nonasymptotics of discretized ULD have been studied in Cheng et al. (2018); Dalalyan & Riou-Durand (2020); Ma et al. (2021), and were demonstrated provably faster than discretized OLD in suitable setups. These are not only great progresses but also forming perspectives complementary to the extensive studies of the convergence of continuous ULD in the mathematical community (e.g, Mattingly et al. (2002); Cao et al. (2019); Dolbeault et al. (2009; 2015); Villani (2009); Eckmann & Hairer (2003); Baudoin (2017); Eberle et al. (2019)). Another important line of research is related to accelerating particle-based approaches for optimization in probability spaces (Liu et al., 2019; Taghvaei & Mehta, 2019; Wang & Li, 2019), although we note there is no clear boundary between these two lines (e.g., Leimkuhler et al. (2018)). Additional interesting ideas also include Chen et al. (2018b); Deng et al. (2020). In general, it has been known that adding an irreversible part to the reversible dynamics of OLD[1] accelerates its convergence (e.g., Hwang et al. (2005); Lelievre et al. (2013); Ohzeki & Ichiki (2015); Rey-Bellet & Spiliopoulos (2015); Duncan et al. (2016)), and this work can be viewed to be under this umbrella. Note, though, the discretization of an accelerated continuous process is also important, and it will be analyzed.

Specifically, we propose an accelerated gradient-based MCMC algorithm termed HFHR. It is motivated by a simple question: how to appropriately inject noise to NAG algorithm **in discrete time**, so that it is turned into an algorithm for momentum-accelerated sampling? Note we don't add noise to the learning-rate$\to 0$ limit of NAG (this has been well studied in Ma et al. (2021)), because a finite-step-size discretization of this limiting ODE may not converge as fast as NAG with the same learning rate. However, we will still use continuous dynamics as intermediate steps.

More precisely, our first step is to combine existing tools to prepare a non-asymptotic formulation for the later steps. The goal is to better account for NAG's behavior when a finite (not infinitesimal) learning rate is used. As pointed out in Shi et al. (2021), a low-resolution limiting ODE (Su et al., 2014), albeit being a milestone leading to a new venue of research (e.g, Wibisono et al. (2016)), does not fully capture the acceleration enabled by NAG — for example, it can't distinguish between NAG and another momentum method of heavy ball (Polyak, 1964). The main reason is, the low-resolution ODE describes the $h \to 0$ limit of NAG, but in practice NAG uses a finite (nonzero) h. High-resolution ODE was thus proposed to include additional $\mathcal{O}(h)$ terms to account for the finite h effect (Shi et al., 2021). The original form of high-resolution ODE involves Hessian of the objective function, which is computationally expensive to evaluate and store for high-dimensional problems, but this difficulty can be overcome using techniques introduced in, e.g., Alvarez et al., 2002; Attouch et al., 2020, which allows us to derive a High-Resolution and Hessian-Free limiting ODE for NAG.

Then we replace the high-resolution term's coefficient in the HFHR ODE by a hyperparameter $\alpha \geq 0$, and then add noise to the resulting ODE in a specific way, which turns it into an SDE suitable for the sampling purpose. This SDE will be termed as HFHR dynamics.

To obtain an actual algorithm, the HFHR SDE is then discretized. We will see, both theoretically and empirically, that nonzero α can lead to accelerated convergence of the sampling algorithm; this acceleration is **not** an artificial consequence of time-rescaling, which would not give acceleration after discretization with an appropriate step size. Meanwhile, note our discretization is just one of the many possible schemes. It was known that high-order discretizations can improve statistical accuracy and even the speed of convergence (see e.g., Chen et al. (2015); Li et al. (2019); Shen & Lee (2019)), although such improvements often come with more computations per iteration. The discretization considered here is just a simple first-order scheme that uses one (full-)gradient evaluation per step, but it better utilizes the structure of HFHR dynamics than Euler-Maruyama.

Our presentation will be structured as follows. After detailing the construction of HFHR, we will analyze its convergence, at both the continuous level (HFHR dynamics) and the discrete level (HFHR algorithm). For precise theoretical results, we will consider the setup of log-strongly-concave target distributions, which are commonly considered in the literature (Kim et al., 2016; Bubeck et al., 2018; Dalalyan, 2017b; Dalalyan & Riou-Durand, 2020; Dwivedi et al., 2019; Shen & Lee, 2019). The additional acceleration of HFHR when compared to ULD in continuous time will be demonstrated explicitly in Thm.5.1. For our discretized HFHR algorithm, a non-asymptotic error bound will

[1]For irreversible-acceleration *not* from OLD, see e.g., Bierkens et al. (2019); Bouchard-Côté et al. (2018).

be obtained (Thm.5.2), which confirms that the additional acceleration in continuous time carries through to the discrete territory. Finally, numerical experiments are provided, verifying the validity and tightness of our theoretical results, and empirically showing HFHR remains advantageous for the nonconvex and high-dimensional example of Bayesian Neural Networks.

The main contribution of this article is the idea of turning NAG-SC optimization algorithm into a sampler, which also introduces a new dynamics that is neither overdamped or underdamped Langevin. Nevertheless, theoretical analyses (e.g., Thm.5.2, Cor.5.4 & Rmk.5.5) and numerical experiments (Sec.6) are also provided to quantify the effectiveness of this idea.

2 BACKGROUND: LANGEVIN DYNAMICS FOR SAMPLING

Consider sampling from Gibbs measure μ whose density is $d\mu = \frac{1}{\int e^{-f(\boldsymbol{y})}d\boldsymbol{y}} e^{-f(\boldsymbol{x})}d\boldsymbol{x}$, where $f : \mathbb{R}^d \mapsto \mathbb{R}$ will be called the potential function. Two diffusion processes popular for sampling (and modeling important physical processes too) are named after Langevin. One is overdamped Langevin dynamics (OLD), and the other is kinetic Langevin dynamics (abbreviated as ULD to comply with a convention of calling it underdamped Langevin). They are respectively given by

$$\text{(OLD)} \quad d\boldsymbol{q}_t = -\nabla f(\boldsymbol{q}_t)dt + \sqrt{2}d\boldsymbol{W}_t \qquad\qquad \text{(ULD)} \quad \begin{cases} d\boldsymbol{q}_t = \boldsymbol{p}_t dt \\ d\boldsymbol{p}_t = -\gamma \boldsymbol{p}_t dt - \nabla f(\boldsymbol{q}_t)dt + \sqrt{2\gamma}d\boldsymbol{B}_t \end{cases}$$

where $\boldsymbol{q}_t, \boldsymbol{p}_t \in \mathbb{R}^d$, $\boldsymbol{W}_t, \boldsymbol{B}_t$ are i.i.d. Wiener processes in $\mathbb{R}^d$, and $\gamma > 0$ is a friction coefficient. Under mild conditions (e.g., Pavliotis (2014)), OLD converges to μ and ULD converges to $d\pi(\boldsymbol{q}, \boldsymbol{p}) = d\mu(\boldsymbol{q})\nu(\boldsymbol{p})d\boldsymbol{p}$, where $\nu(\boldsymbol{p}) = (2\pi)^{-\frac{d}{2}} e^{-\|\boldsymbol{p}\|^2/2}$, so its $\boldsymbol{q}$ marginal follows μ.

OLD and ULD are closely related. In fact, OLD is the $\gamma \to \infty$ overdamping limit of ULD after time dilation (e.g., Pavliotis (2014)). However, OLD is a reversible Markov process but ULD is irreversible, and thus both their equilibrium and non-equilibrium statistical mechanics are different, although closely related too. We will only focus on the convergence to statistical equilibrium (see e.g., Souza & Tao (2019) for non-equilibrium aspects).

Many celebrated approaches exist for establishing the exponential convergence (a.k.a. geometric ergodicity) of OLD, including the seminal work of Roberts et al. (1996), the ones using spectral gap (e.g., Dalalyan, 2017b, Lemma 1), synchronous coupling (Villani, 2008, p33-35)(Durmus et al., 2019b, Proposition 1), functional inequalities such as Poincaré's inequality (Pavliotis, 2014, Theorem 4.4) and log Sobolev inequality (Vempala & Wibisono, 2019, Theorem 1). There are also fruitful results for ULD, including the ones leveraging Lyapunov function (Mattingly et al., 2002, Theorem 3.2), hypocoercivity (Villani, 2009; Dolbeault et al., 2009; 2015; Roussel & Stoltz, 2018), coupling (Cheng et al., 2018, Theorem 5)(Dalalyan & Riou-Durand, 2020, Theorem 1)(Eberle et al., 2019, Theorem 2.3), LSI (Ma et al., 2021, Section 3.1), modified Poincaré's inequality (Cao et al., 2019, Theorem 1), and spectral analysis (Kozlov, 1989; Eckmann & Hairer, 2003).

The study of asymptotic convergence of discretized OLD dates back to at least the 1990s (Meyn et al., 1994; Roberts et al., 1996). The non-asymptotic analysis of LMC discretization of OLD can be found in Dalalyan (2017b) and it shows the discretization achieves ϵ error, in TV distance, in $\tilde{\mathcal{O}}(d/\epsilon^2)$ steps. Subsequent results include $\tilde{\mathcal{O}}(d/\epsilon^2)$ in W_2 (Durmus & Moulines, 2016), $\tilde{\mathcal{O}}(d/\epsilon)$ in KL (Cheng & Bartlett, 2018), $\tilde{\mathcal{O}}(d/\epsilon)$ in W_2 under additional 3rd-order regularity (Durmus et al., 2019b), and $\tilde{\mathcal{O}}(\sqrt{d}/\epsilon)$ in W_2 under additional 3rd-order regularity (Li et al., 2021). For discretized ULD, one has $\tilde{\mathcal{O}}(\sqrt{d}/\epsilon)$ iteration complexity in W_2 (Cheng et al., 2018; Dalalyan & Riou-Durand, 2020) and $\tilde{\mathcal{O}}(\sqrt{d}/\sqrt{\epsilon})$ in KL (Ma et al., 2021). ULD is still generally conceived to be advantageous over OLD and sometimes understood as its momentum-accelerated version.

3 NOTATIONS AND CONDITIONS

We will use 2-Wasserstein distance to quantify convergence, i.e. $W_2(\mu_1, \mu_2) = \left(\inf_{\pi \in \Pi(\mu_1,\mu_2)} \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim\pi}\|\boldsymbol{X} - \boldsymbol{Y}\|^2\right)^{\frac{1}{2}}$ where $\Pi(\mu_1, \mu_2)$ is the set of all couplings of μ_1 and μ_2.

Assume WLOG that $\mathbf{0} \in \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$. The following condition will also be frequently used.

A 1. ***(Standard Strong-Convexity and Smoothness Condition)*** *Function $f \in \mathcal{C}^1(\mathbb{R}^d) : \mathbb{R}^d \mapsto \mathbb{R}$ is m-stronly-convex and L-smooth, if there exist constants $m, L > 0$ such that $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, we have*

$$\|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x})\| \leq L\|\boldsymbol{y} - \boldsymbol{x}\| \text{ and } f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{m}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2$$

For $f \in \mathcal{C}^2$, this condition is equivalent to $mI \preceq \nabla^2 f \preceq LI$.

4 THE CONSTRUCTION OF HFHR DYNAMICS

HFHR is obtained by formulating NAG-SC as a Hessian free high-resolution ODE, lifting the high-resolution term's coefficient as a free parameter, and adding appropriate noises.

More precisely, let's start with NAG-SC algorithm:

$$\boldsymbol{x}_{k+1} = \boldsymbol{y}_k - s\nabla f(\boldsymbol{y}_k) \tag{1}$$
$$\boldsymbol{y}_{k+1} = \boldsymbol{x}_{k+1} + c(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \tag{2}$$

where s is the learning rate (also known as step size), and $c = \frac{1-\sqrt{ms}}{1+\sqrt{ms}}$ is a constant based on s and the strong convexity coefficient m of f; the method also works for non-strongly-convex f though.

A high-resolution ODE description of Eq.(1) & (2) is obtained in (Shi et al., 2021, Section 2)

$$\ddot{\boldsymbol{y}} + \sqrt{s}\left(\frac{2(1-c)}{s(1+c)} + \nabla^2 f(\boldsymbol{y})\right)\dot{\boldsymbol{y}} + \frac{2}{1+c}\nabla f(\boldsymbol{y}) = \boldsymbol{0}, \tag{3}$$

which can better account for the effect of non-infinitesimal s than the $s \to 0$ limit (note c depends on s). However, in this original form, Hessian of f is involved, which is expensive to compute and store especially for high-dimensional problems.

To obtain a Hessian-free high-resolution ODE description of Equation (1) and (2), we first turn the iteration into a 'mechanical' version by introducing position variable $\boldsymbol{q}_k = \boldsymbol{y}_k$ and momentum variable $\boldsymbol{p}_k = (\boldsymbol{y}_k - \boldsymbol{x}_k)/h$. Replacing $\boldsymbol{x}_{k+1}$ in (1) and the first $\boldsymbol{x}_{k+1}$ in (2) by $\boldsymbol{q}_{k+1}$ and $\boldsymbol{p}_{k+1}$, the second $\boldsymbol{x}_{k+1}$ in (2) by $\boldsymbol{q}_k - s\nabla f(\boldsymbol{q}_k)$, and the $\boldsymbol{x}_k$ in (2) by $\boldsymbol{q}_k$ and $\boldsymbol{p}_k$, we obtain

$$\begin{cases} \boldsymbol{q}_{k+1} = \boldsymbol{q}_k + h\boldsymbol{p}_{k+1} - s\nabla f(\boldsymbol{q}_k) \\ \boldsymbol{p}_{k+1} = c\boldsymbol{p}_k - c\frac{s}{h}\nabla f(\boldsymbol{q}_k) \end{cases}$$

Now, choose γ, α and h as $h = \sqrt{cs}, \gamma = \frac{1-c}{h}, \alpha = \frac{s}{h}$. It is easy to see that $\gamma > 0$, $\alpha > 0$, then NAG-SC exactly rewrites as

$$\begin{cases} \boldsymbol{q}_{k+1} = \boldsymbol{q}_k + h\boldsymbol{p}_{k+1} - h\alpha\nabla f(\boldsymbol{q}_k) \\ \boldsymbol{p}_{k+1} = \boldsymbol{p}_k - h\gamma\boldsymbol{p}_k - h\nabla f(\boldsymbol{q}_k) \end{cases}. \tag{4}$$

Note the technique for bypassing the Hessian without introducing any approximation is already well studied in the literature (e.g., Alvarez et al. (2002); Attouch et al. (2020)).

So far, both h and α are actually determined by the hyperparameter s of NAG-SC. However, if we now consider α as an independent variable (i.e., 'lift' it) and let $h \to 0$, we see (4) is a 1st-order discretization (with step size h) of the dynamics

$$\begin{cases} \dot{\boldsymbol{q}} = \boldsymbol{p} - \alpha\nabla f(\boldsymbol{q}) \\ \dot{\boldsymbol{p}} = -\gamma\boldsymbol{p} - \nabla f(\boldsymbol{q}) \end{cases}. \tag{5}$$

Note α, if inherited from NAG-SC, should be $\alpha = \sqrt{s/c} = \mathcal{O}(h)$, which, in a low-resolution ODE will be discarded, and this eventually leads to ULD rather than HFHR. However, we now allow it to be a free parameter and will see that $\alpha \neq \mathcal{O}(h)$ can be advantageous.

Before quantifying these advantages later on, we finish the construction by appropriately injecting Gaussian noises to Equation (5). This is just like how OLD can be obtained by adding noise to the

gradient flow. The right amount and structure of noise turn the ODE into a Markov process that can serve the purpose of sampling, and the detailed form of our noise is given by:

$$\begin{cases} d\boldsymbol{q}_t = (\boldsymbol{p}_t - \alpha\nabla f(\boldsymbol{q}_t))dt + \sqrt{2\alpha}d\boldsymbol{W}_t \\ d\boldsymbol{p}_t = (-\gamma\boldsymbol{p}_t - \nabla f(\boldsymbol{q}_t))dt + \sqrt{2\gamma}d\boldsymbol{B}_t \end{cases} . \tag{6}$$

Here $\alpha \geq 0, \gamma > 0$ are constant parameters, and $\boldsymbol{W}_t, \boldsymbol{B}_t$ are independent standard Brownian motions in $\mathbb{R}^d$. This irreversible process will be named as **Hessian-Free High-Resolution**(**HFHR**) dynamics. We write it as HFHR(α, γ) to emphasize the dependence on α and γ when needed.

Substitution into Fokker-Planck PDE shows HFHR dynamics is unbiased (proof in Appendix B.1):

Theorem 4.1. *π is the invariant distribution of HFHR described in Eq.*(6)*, just like ULD.*

5 Theoretical Analysis of the HFHR Dynamics/Algorithm

5.1 HFHR Dynamics in Continuous Time

Let's establish the exponential convergence of HFHR dynamics and its additional acceleration when compared to ULD, when the target measure has a strongly-convex and smooth potential.

Theorem 5.1. *Assume Conditions A1 holds and further assume $\gamma^2 > L + m$ and $\alpha \leq \frac{\gamma^2 - L - m}{m\gamma}$. Denote the law of $\boldsymbol{q}_t$ by μ_t. Then there exists $\kappa' > 0$ depending only on α and γ, such that*

$$W_2(\mu_t, \mu) \leq \kappa' e^{-(\frac{m}{\gamma} + m\alpha)t} W_2(\mu_0, \mu).$$

Detailed expression of κ' can be found in Appendix A.

Thm. 5.1 state that HFHR converges to the target distribution exponentially fast in log-strongly-concave and smooth setup. For ULD, Dalalyan & Riou-Durand (2020, Theorem 1) obtained exponential convergence result in 2-Wasserstein distance with rate $\frac{\sqrt{m}}{\sqrt{\kappa}+\sqrt{\kappa-1}}$ using a simple and elegant coupling approach, and showed this rate is optimal as it is achieved by the bivariate function $f(x, y) = \frac{m}{2}x^2 + \frac{L}{2}y^2$. In Thm 5.1, we use the same coupling approach to obtain an (asymptotically) equivalent rate $\frac{\sqrt{m}}{2\sqrt{\kappa}}$. Since ULD is HFHR(0,$\gamma$) and our bound agrees with existing result when $\alpha = 0$ and shows faster convergence for $\alpha > 0$, the acceleration of HFHR in continuous time is evidenced. For example, if we set $\gamma = 2\sqrt{L}$ and push α to the upper bound specified in Thm. 5.1, we obtain an $\mathcal{O}(\sqrt{L})$ rate in the log-strongly-concave setup. Compared with the rate in Dalalyan & Riou-Durand (2020), this is a speed-up of order κ.

5.2 HFHR Algorithm in Discrete Time

To obtain an implementable method, we now discretize the time of HFHR dynamics. As our main goal is to show the acceleration enabled by α won't disappear after discretization (unlike a fake acceleration due to time rescaling), we'll just analyze a 1st-order discretization (but a high-accuracy discretization adapted from RMA (Shen & Lee, 2019) will also be provided, in Appendix F).

For simplicity, we'll work with constant step size h. Inspired by Strang splitting for differential equations (Strang, 1968; McLachlan & Quispel, 2002), we consider a symmetric composition for updating over each time interval $[kh, (k+1)h]$: $\boldsymbol{x}_{k+1} := \phi^{\frac{h}{2}} \circ \psi^h \circ \phi^{\frac{h}{2}}(\boldsymbol{x}_k)$ where $\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{q}_{kh} \\ \boldsymbol{p}_{kh} \end{bmatrix}$, ϕ and ψ correspond to solution flows of split SDEs, respectively given by

$$\phi : \begin{cases} d\boldsymbol{q} = \boldsymbol{p}dt \\ d\boldsymbol{p} = -\gamma\boldsymbol{p}dt + \sqrt{2\gamma}d\boldsymbol{B} \end{cases} \qquad \psi : \begin{cases} d\boldsymbol{q} = -\alpha\nabla f(\boldsymbol{q})dt + \sqrt{2\alpha}d\boldsymbol{W} \\ d\boldsymbol{p} = -\nabla f(\boldsymbol{q})dt \end{cases} ,$$

and $\phi^t(\boldsymbol{x}_0)$ and $\psi^t(\boldsymbol{x}_0)$ mean x's value after evolving ϕ and ψ for t time with initial condition $\boldsymbol{x}_0$.

Note that ϕ flow can be solved explicitly since the second equation is an Ornstein-Unlenbeck process and integrating the second equation followed by integrating the first one gives us an explicit solution

$$\begin{cases} \boldsymbol{q}_t = \boldsymbol{q}_0 + \frac{1-e^{-\gamma t}}{\gamma}\boldsymbol{p}_0 + \sqrt{2\gamma}\int_0^t \frac{1-e^{-\gamma(t-s)}}{\gamma} d\boldsymbol{B}(s), \\ \boldsymbol{p}_t = e^{-\gamma t}\boldsymbol{p}_0 + \sqrt{2\gamma}\int_0^t e^{-\gamma(t-s)} d\boldsymbol{B}(s). \end{cases} \tag{7}$$

For an implementation of the stochastic integral part in Equation 7, denoting $\boldsymbol{X} = \sqrt{2\gamma}\int_0^t \frac{1-e^{-\gamma(t-s)}}{\gamma} d\boldsymbol{B}(s)$ and $\boldsymbol{Y} = \sqrt{2\gamma}\int_0^t e^{-\gamma(t-s)} d\boldsymbol{B}(s)$, and the covariance matrix of $(\boldsymbol{X}, \boldsymbol{Y})$ is $\mathrm{Cov}(\boldsymbol{X}, \boldsymbol{Y}) = \begin{bmatrix} \frac{\gamma h + 4e^{-\gamma\frac{h}{2}} - e^{-\gamma h} - 3}{\gamma^2} I_d & \frac{(1-e^{-\gamma\frac{h}{2}})^2}{\gamma} I_d \\ \frac{(1-e^{-\gamma\frac{h}{2}})^2}{\gamma} I_d & (1-e^{-\gamma h}) I_d \end{bmatrix}$. As mean and covariance fully determine a Gaussian distribution, $\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix} = M\boldsymbol{\xi}$ where M is the Cholesky decomposition of $\mathrm{Cov}(\boldsymbol{X}, \boldsymbol{Y})$, $\boldsymbol{\xi}$ is a $2d$ standard Gaussian random vector, i.i.d. at each step, and ϕ^t can thus be exactly simulated.

However, ψ flow is generally not explicitly solvable unless f is a quadratic function in $\boldsymbol{q}$. We simply choose to approximate $\psi^h(\boldsymbol{x}_0)$ with one-step Euler-Maruyama integration $\psi^h(\boldsymbol{x}_0) \approx \widetilde{\psi}^h(\boldsymbol{x}_0)$ given by $\begin{cases} \boldsymbol{q}_h = \boldsymbol{q}_0 - \alpha\nabla f(\boldsymbol{q}_0)h + \sqrt{2\alpha h}\boldsymbol{\eta} \\ \boldsymbol{p}_h = \boldsymbol{p}_0 - \nabla f(\boldsymbol{q}_0)h \end{cases}$ where $\boldsymbol{\eta}$ is a standard d-dimensional Gaussian random vector, again i.i.d. each time $\tilde{\psi}$ is called.

Altogether, one step of an implementable Strang's splitting of HFHR is hence $\phi^{\frac{h}{2}} \circ \widetilde{\psi}^h \circ \phi^{\frac{h}{2}}$ and we call this numerical scheme the HFHR algorithm, which is summarized in Algorithm 1.

Algorithm 1 HFHR Algorithm

Input: potential function f and its gradient ∇f, damping coefficients α and γ, step size h, initial condition $(\boldsymbol{q}_0, \boldsymbol{p}_0)$
procedure DISCRETIZED HFHR($f, \nabla f, \alpha, \gamma, h, \boldsymbol{q}_0, \boldsymbol{p}_0$)
  $k = 0$ and initialize $\begin{bmatrix} \boldsymbol{q}_0 \\ \boldsymbol{p}_0 \end{bmatrix}$
  while not converge **do**
    Generate independent standard Gaussian random vectors $\boldsymbol{\eta}_{k+1} \in \mathbb{R}^d, \boldsymbol{\xi}^1_{k+1}, \boldsymbol{\xi}^2_{k+1} \in \mathbb{R}^{2d}$
    Run $\phi^{\frac{h}{2}}$: $\begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{p}_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_{kh} + \frac{1-e^{-\gamma\frac{h}{2}}}{\gamma}\boldsymbol{p}_{kh} \\ e^{-\gamma\frac{h}{2}}\boldsymbol{p}_{kh} \end{bmatrix} + M\boldsymbol{\xi}^1_{k+1}$
    Run $\widetilde{\psi}^h$: $\begin{bmatrix} \boldsymbol{q}_2 \\ \boldsymbol{p}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1 - \alpha\nabla f(\boldsymbol{q}_1)h + \sqrt{2\alpha h}\boldsymbol{\eta}_{k+1} \\ \boldsymbol{p}_1 - \nabla f(\boldsymbol{q}_1)h \end{bmatrix}$
    Run $\phi^{\frac{h}{2}}$: $\begin{bmatrix} \boldsymbol{q}_3 \\ \boldsymbol{p}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_2 + \frac{1-e^{-\gamma\frac{h}{2}}}{\gamma}\boldsymbol{p}_2 \\ e^{-\gamma\frac{h}{2}}\boldsymbol{p}_2 \end{bmatrix} + M\boldsymbol{\xi}^2_{k+1}$
    $\begin{bmatrix} \boldsymbol{q}_{(k+1)h} \\ \boldsymbol{p}_{(k+1)h} \end{bmatrix} \leftarrow \begin{bmatrix} \boldsymbol{q}_3 \\ \boldsymbol{p}_3 \end{bmatrix}$
    $k \leftarrow k + 1$
  end while
end procedure

As ψ in Strang splitting is replaced by a 1st-order approximation $\tilde{\psi}$, the method is of order 1, however with good constant. This is rigorously established by the following theorem (interested readers are referred to Appendix.D.5-D.7 and Li et al. (2021) for more technical details):

Theorem 5.2. *Under Assumption 1, we further assume $\gamma - \frac{L+m}{\gamma} \geq m\alpha$ and the function $\nabla\Delta f$ satisfies a third-order growth condition, i.e., $\left\|\nabla\Delta f(\boldsymbol{q})\right\| \leq G\sqrt{1+\|\boldsymbol{q}\|^2}, \forall \boldsymbol{q} \in \mathbb{R}^d$ for some $G > 0$. If $(\boldsymbol{q}_0, \boldsymbol{p}_0) \sim \pi_0$, then there exists $h_0, C > 0$ such that when $0 < h < h_0$, we have*

$$W_2(\mu_k, \mu) \leq \sqrt{2}\kappa' e^{-(\frac{m}{\gamma}+m\alpha)kh} W_2(\pi_0, \pi) + \sqrt{2}Ch \tag{8}$$

where κ' is a constant depending only on L, m, γ, α (see Appendix A for detail), μ_k is the law of the q marginal of the k-th iterate in Algorithm 1, and μ is the q marginal of the invariant distribution π.

In particular, $C = \mathcal{O}(\sqrt{d})$ and there exists $b > 0$, independent of α and is of order $\mathcal{O}(\sqrt{d})$, s.t.

$$C \leq \frac{b}{m}(\alpha^2 - \frac{\alpha}{\gamma} + \frac{1}{\gamma^2}). \tag{9}$$

Remark 5.3. *The linear growth (at infinity) condition on $\nabla\Delta f$ is actually not as restrictive as it appears. For example, for monomial potentials, i.e., $f(x) = x^p, p \in \mathbb{Z}_+$, our linear growth condition is met when $p \leq 4$, whereas a standard condition (Pavliotis, 2014, Theorem 3.1) for the existence of SDE solutions holds only when $p \leq 2$. In addition, our condition is related to the Hessian Lipschitz condition commonly used in the literature (e.g., Durmus et al. (2019b); Ma et al. (2021)). Smoothness and Hessian Lipschitzness imply our condition. Meanwhile, examples that satisfy linear growth condition but are not Hessian Lipschitz exist, e.g., $f(x) = x^4$, and thus linear growth condition is not necessarily stronger than Hessian Lipschitzness.*

Inspecting the role of α in Equation (8), we see that α clearly increases the rate of exponential decay, but at the same time it can also increase the discretization error (see (9); assuming h is fixed). However, as the following Corollary 5.4 and its remark will show, the net effect of having a positive $\alpha > 0$, at least for some $\alpha^\star$, is reduced iteration complexity.

Corollary 5.4. *Consider the same assumption as in Thm. 5.2. If $(\boldsymbol{q}_0, \boldsymbol{p}_0) \sim \pi_0$, then there exists $h_0, C > 0$ (same as that in Theorem 5.2; recall $C = \mathcal{O}(\sqrt{d})$) such that for any target error tolerance $\epsilon > 0$, if we choose $h = h^\star \triangleq \min\{h_0, \frac{\epsilon}{2\sqrt{2}C}\}$, then for $\epsilon < 2\sqrt{2}Ch_0$, after*

$$k^\star = 2\sqrt{2}\frac{C}{\frac{m}{\gamma} + m\alpha}\frac{1}{\epsilon}\log\frac{2\sqrt{2}\kappa' W_2(\pi_0, \pi)}{\epsilon} = \tilde{\mathcal{O}}\left(\frac{\sqrt{d}}{\epsilon}\right). \tag{10}$$

steps, we have $W_2(\mu_k, \mu) \leq \epsilon$.

Remark 5.5. *Recall from Thm.5.2 that $C \leq \frac{b}{m}(\alpha^2 - \frac{\alpha}{\gamma} + \frac{1}{\gamma^2})$, so if we consider the minimizer $\alpha^\star$ of an upper bound of $\frac{C}{\frac{m}{\gamma}+m\alpha}$, $\alpha^\star = \operatorname{argmin}_{\alpha\geq 0}\frac{b}{m^2}\frac{\alpha^2 - \frac{\alpha}{\gamma} + \frac{1}{\gamma^2}}{\frac{1}{\gamma}+\alpha} = \frac{\sqrt{3}-1}{\gamma}$. This suggests that by choosing an optimal $\alpha > 0$, one could effectively reduce iteration complexity. Note, however, that this $\alpha^\star$ may not be the true optimal one as bounds may not be tight. If they were, $k^\star_{\alpha^\star} = (2\sqrt{3} - 3)k^\star_{\alpha=0} \approx 0.46k^\star_{\alpha=0}$; i.e., steps needed by ULD (discretized by Alg.1 with $\alpha = 0$) can be halved by HFHR.*

Rmk.5.5 shows HFHR algorithm can lead to a similar bound on iteration complexity as ULD algorithm but with an improved constant, and thus is a more efficient algorithm. This improvement also shows that the acceleration of HFHR can be carried through from continuous time to discrete time. The same conclusion has been consistently observed in numerical experiments too.

Remark 5.6. *Readers interested in more explicit condition number dependence are referred to Appendix E, where we show, for 2D Gaussian target with condition number $\kappa \gg 1$, the convergences of Euler discretization of ULD under optimal parameters and HFHR under suboptimal parameters are, respectively, given by $(1 - 1/\kappa + o(1/\kappa))^n$ and $(1 - 2/\kappa + o(1/\kappa))^n$, where n is the number of iterations. The latter (HFHR) is faster despite that its hyperparameters may not be optimal.*

6 Numerical Experiments

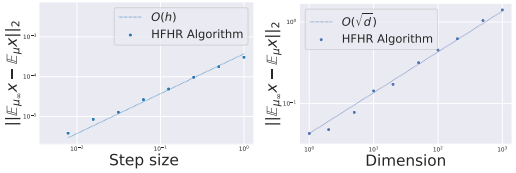
We now empirically study the acceleration enabled by $\alpha \neq 0$ by comparing HFHR algorithm and the popular KLMC discretization of ULD (Dalalyan & Riou-Durand, 2020). For fairness, discretizations of the same order and cost are compared (Appendix F has an additional comparison).

6.1 Verification of theoretical results in Section 5.2

This subsection numerically verifies the d and h dependence of HFHR algorithm as well as the genuine acceleration enabled by α. For this purpose, we will *not* use Gaussian targets, because otherwise HFHR will decouple across different (orthogonal) dimensions, and hence its discretization error having a $\mathcal{O}(\sqrt{d})$ dependence would just be a consequence of using 2-Wasserstein distance for quantifying statistical accuracy. To inspect a more interesting example, we consider a potential that is no longer additive across different dimensions, namely $f(\boldsymbol{x}) = \log\left(e^{x_1} + \cdots + e^{x_d}\right) + \frac{1}{2}\|\boldsymbol{x}\|^2$.

It's not hard to see that all dimensions will be coupled in the HFHR dynamics (and ULD too). Moreover, the new potential f is still a strongly convex function and satisfies the assumption in Theorem 5.2. When the target measure is non-Gaussian, we no longer have a closed form expression for 2-Wasserstein distance and it is computationally expensive to approximate the 2-Wasserstein distance by samples. Therefore, we use the error of mean instead as a surrogate because $\left\|\mathbb{E}_{\mu_k}\boldsymbol{q} - \mathbb{E}_{\mu}\boldsymbol{q}\right\| \leq W_2(\mu_k, \mu)$ and hence the bound in Equation (8) also applies to the error in mean, and so does the iteration complexity bound in Equation (10).

Theorem 5.2 says the final sampling error is upper bounded the discretization error that is linear in h and $\sqrt{d}$. To numerically verify the linear dependence on h, we work with $d = 2$ and ran ULD algorithm for sufficiently long time with tiny step size ($h = 0.0005$) to obtain 10^8 independent realizations and use them (as benchmark) to empirically estimate $\mathbb{E}_{\mu}\boldsymbol{q}$. We then set $\gamma = 2, \alpha = 1$ and use Algorithm 1, with $h \in \{2^k | -7 \leq k \leq 0\}$. For each h, we run $\frac{T}{h}$ (with $T = 50$) iterations in Algorithm 1 to ensure the Markov chains are well-mixed and the contribution to final error from exponential decay is order-of-magnitude smaller than discretization error. The results are plotted in Figure 1a. The observed linear dependence on the step size h is consistent with Theorem 5.2.



(a) Linear dependence of discretization error of Algorithm 1 on h

(b) Linear dependence of discretization error of Algorithm 1 on $\sqrt{d}$

Figure 1: Illustration of the consistency between the theoretical bound in Theorem 5.2 and experiment results.

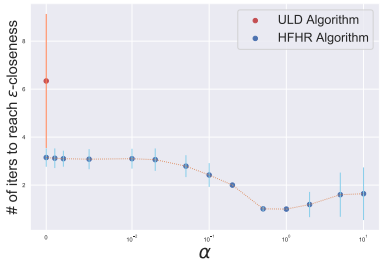


Figure 2: Improvement of Algorithm 1 over ULD algorithm in iteration complexity. (vertical bar = 1 std.)

To numerically verify the $\mathcal{O}(\sqrt{d})$ dependence, we extensively experiment with $d \in \{1, 2, 5, 10, 20, 50, 100, 200, 500, 1000\}$. For each d, we run 1,000 independent realizations of ULD algorithm until well converged with tiny step size ($h = 0.005$) and use their empirical average as the 'true' mean. Then we fix $\gamma = 2, \alpha = 1, h = 0.1, T = 10$ and for each d, we run 1,000 independent realizations of HFHR algorithm for $\frac{T}{h} = 100$ iterations. Experiment results are plotted in Figure 1b and the linear trend demonstrates that the bound in Thm. 5.2 is tight in terms of d-dependence.

The final experiment compares Algorithm 1 with ULD algorithm in terms of iteration complexity. The goal is to demonstrate the genuine acceleration of HFHR is not an artifact due to time rescaling, which would disappear after discretization as the stability limit changes accordingly. To do so, we push both ULD and HFHR to their respective largest h values that still allow monotonic convergence at a large scale, and compare their mixing times. For general nonlinear problems like the one here, Remark 5.5 suggests that with appropriately chosen α, HFHR algorithm effectively reduces the constant factor of the iteration complexity, implying accelerated sampling. To just provide one empirical verification of this improvement, we choose $d = 10$ and use the error of mean $\left\|\mathbb{E}_{\mu_k}\boldsymbol{q} - \mathbb{E}_{\mu}\boldsymbol{q}\right\|$ to measure sampling accuracy. The benchmark, i.e., $\mathbb{E}_{\mu}\boldsymbol{q}$, is again obtained from 1,000 independent realizations of ULD algorithm with tiny step size ($h = 0.005$), ran for long enough to ensure the corresponding Markov chain is well-mixed. The initial measure is chosen as a Dirac measure at $(100 \times \mathbf{1}_d, \mathbf{0}_d)$, where $\mathbf{1}_d, \mathbf{0}_d$ are d-dimensional vectors filled with 1 and 0 respectively. We pick threshold $\epsilon = 0.1$, and for each $\alpha \in \{0, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100\}$, we try all combinations of $(\gamma, h) \in \{0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100\} \times \{0.1 \times [50]\}$ for Algorithm 1 (we also run ULD algorithm when $\alpha = 0$), and empirically find the best combination that requires the fewest iterations to meet $\left\|\mathbb{E}_{\mu_k}\boldsymbol{q} - \mathbb{E}_{\mu}\boldsymbol{q}\right\| \leq \epsilon$. We find that $h = 5$ already surpasses the stability limit of ULD algorithm, hence the range of step size covers the largest step size that are practically useable for ULD algorithm. Experiments are repeated with 100 different seeds to further reduce variance.

The results are shown in Figure 2. When $\alpha > 0$, HFHR algorithm consistently outperforms ULD algorithm (note it also does so when $\alpha = 0$ because HFHR uses a efficiency-wise-comparable but more refined discretization than ULD algorithm). In particular, when $\alpha = 0.5$ and 1, which are

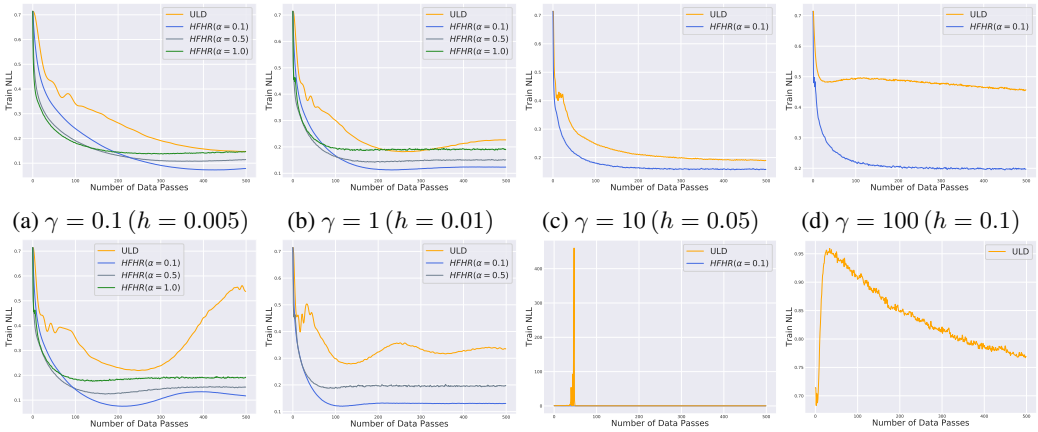
empirically the best values we found for this experiment, HFHR algorithm achieves the specified ϵ-closeness nearly $6\times$ times faster than ULD algorithm, and its decreased mixing time (compared to $\alpha = 0$ for the same algorithm) is consistent with the ≈ 0.46 factor in Rmk.5.5). This empirical study corroborates that the acceleration HFHR dynamics creates also carries through to its discretization, and the acceleration of HFHR algorithm over ULD algorithm can be significant.

6.2 Bayesian Neural Network

Now consider Bayesian neural network (BNN) which is a compelling learning model (Wilson, 2020); however, the focus won't be on its learning capability, and instead we just consider its training, which amounts to a practical, high-dim., multi-modal example of sampling tasks. It no longer satisfies the conditions of our analysis, and our goal is to show HFHR still accelerates. We use fully-connected network with [22, 10, 2] neurons, ReLU, standard Gaussian prior for all parameters, and compare ULD and HFHR on data set Parkinson from UCI repository (Dua & Graff, 2017).

Choices of hyper-parameter for Algorithm 1 and ULD algorithm are systematically investigated. For each pair $(\gamma, \alpha) \in \{0.1, 0.5, 1, 5, 10, 50, 100\}^2$, we empirically tune the step size to the stability limit of ULD algorithm, simulate 10,000 independent realizations, and use the ensemble to conduct Bayesian posterior prediction. HFHR will then use the same step size. For each γ, we plot the negative log likelihood of HFHR algorithm (with different α choices) and ULD algorithm on training and test data in Figure 3. Cases where α is too large for numerical stability are not drawn.

From Figure 3, we find that HFHR converges significantly faster than ULD in a wide range of setups. In general, the log-strongly-concave assumption required in Theorem 5.2 does not hold for multimodal target distributions. However, this numerical result shows that HFHR still accelerates ULD for highly complex models such as BNN, even when there is no obvious theoretical guarantee. It showcases the applicability and effectiveness of HFHR as a general sampling algorithm.



(a) $\gamma = 0.1$ $(h = 0.005)$ (b) $\gamma = 1$ $(h = 0.01)$ (c) $\gamma = 10$ $(h = 0.05)$ (d) $\gamma = 100$ $(h = 0.1)$

(e) $\gamma = 0.1$ $(h = 0.01)$ (f) $\gamma = 1$ $(h = 0.02)$ (g) $\gamma = 10$ $(h = 0.1)$ (h) $\gamma = 100$ $(h = 0.2)$

Figure 3: Training Negative Log-Likelihood (NLL) for various γ. Row 1: step sizes are close to the stability limit of ULD algorithm; Row 2: further increased step size exceeds that stability limit.

7 Conclusion and Discussion

This paper proposes HFHR, an accelerated gradient-based MCMC method for sampling. To demonstrate the acceleration enabled by HFHR, the geometric ergodicity of HFHR (both the continuous and the discretized versions) is quantified, and its convergence is provably faster than Underdamped Langevin Dynamics, which by itself is often already considered as an acceleration of Overdamped Langevin Dynamics. As HFHR is based on a new perspective, which is to turn NAG-SC optimizer with **finite** learning rate into a sampler, there are a number of interesting directions in which this work can be extended. Besides further theoretical investigations that aim at refining the error bounds, examples also include the followings: to scale HFHR up to large data sets, full gradient may be replaced by stochastic gradient (SG) — how to quantify, and hence optimize the performance of SG-HFHR? Can the generalization ability of HFHR trained learning models (e.g., BNN) be quantified, and how does it compare with that by LMC and/or KLMC? These will be future work.

REFERENCES

Felipe Alvarez, Hedy Attouch, Jérôme Bolte, and Patrick Redont. A second-order gradient-like dissipative dynamical system with hessian-driven damping.: Application to optimization and mechanics. *Journal de mathématiques pures et appliquées*, 81(8):747–779, 2002.

Hedy Attouch, Zaki Chbani, Juan Peypouquet, and Patrick Redont. Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity. *Mathematical Programming*, 168 (1-2):123–175, 2018.

Hedy Attouch, Zaki Chbani, Jalal Fadili, and Hassan Riahi. First-order optimization algorithms via inertial systems with hessian driven damping. *Mathematical Programming*, pp. 1–43, 2020.

Fabrice Baudoin. Bakry-emery meet villani. *Journal of Functional Analysis*, 2017.

GNJC Bierkens, Paul Fearnhead, and Gareth Roberts. The zig-zag process and super-efficient sampling for bayesian analysis of big data. *Annals of Statistics*, 47(3), 2019.

Alexandre Bouchard-Côté, Sebastian J Vollmer, and Arnaud Doucet. The bouncy particle sampler: A nonreversible rejection-free markov chain monte carlo method. *Journal of the American Statistical Association*, 113(522):855–867, 2018.

Sébastien Bubeck, Ronen Eldan, and Joseph Lehec. Sampling from a log-concave distribution with projected langevin monte carlo. *Discrete & Computational Geometry*, 59(4):757–783, 2018.

Yu Cao, Jianfeng Lu, and Lihan Wang. On explicit l_2-convergence rate estimate for underdamped langevin dynamics. *arXiv preprint arXiv:1908.04746*, 2019.

Changyou Chen, Nan Ding, and Lawrence Carin. On the convergence of stochastic gradient MCMC algorithms with high-order integrators. *NIPS*, 2015.

Changyou Chen, Ruiyi Zhang, Wenlin Wang, Bai Li, and Liqun Chen. A unified particle-optimization framework for scalable bayesian sampling. In *The Conference on Uncertainty in Artificial Intelligence*, 2018a.

Yi Chen, Jinglin Chen, Jing Dong, Jian Peng, and Zhaoran Wang. Accelerating nonconvex learning via replica exchange langevin diffusion. In *International Conference on Learning Representations*, 2018b.

Xiang Cheng and Peter L Bartlett. Convergence of langevin mcmc in kl-divergence. *PMLR 83*, (83):186–211, 2018.

Xiang Cheng, Niladri S Chatterji, Peter L Bartlett, and Michael I Jordan. Underdamped langevin mcmc: A non-asymptotic analysis. *Proceedings of the 31st Conference On Learning Theory, PMLR*, 2018.

Sinho Chewi, Chen Lu, Kwangjun Ahn, Xiang Cheng, Thibaut Le Gouic, and Philippe Rigollet. Optimal dimension dependence of the metropolis-adjusted langevin algorithm. *COLT*, 2021.

Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in neural information processing systems*, pp. 3036–3046, 2018.

Arnak Dalalyan. Further and stronger analogy between sampling and optimization: Langevin monte carlo and gradient descent. In *Conference on Learning Theory*, pp. 678–689. PMLR, 2017a.

Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3): 651–676, 2017b.

Arnak S Dalalyan and Lionel Riou-Durand. On sampling from a log-concave density using kinetic Langevin diffusions. *Bernoulli*, 26(3):1956–1988, 2020.

Wei Deng, Qi Feng, Liyao Gao, Faming Liang, and Guang Lin. Non-convex learning via replica exchange stochastic gradient mcmc. In *International Conference on Machine Learning*, pp. 2474–2483. PMLR, 2020.

Jean Dolbeault, Clément Mouhot, and Christian Schmeiser. Hypocoercivity for kinetic equations with linear relaxation terms. *Comptes Rendus Mathematique*, 347(9-10):511–516, 2009.

Jean Dolbeault, Clément Mouhot, and Christian Schmeiser. Hypocoercivity for linear kinetic equations conserving mass. *Transactions of the American Mathematical Society*, 367(6):3807–3828, 2015.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Andrew B Duncan, Tony Lelievre, and GA Pavliotis. Variance reduction using nonreversible langevin samplers. *Journal of statistical physics*, 163(3):457–491, 2016.

Alain Durmus and Eric Moulines. Sampling from strongly log-concave distributions with the unadjusted langevin algorithm. *arXiv preprint arXiv:1605.01559*, 5, 2016.

Alain Durmus, Szymon Majewski, and Błażej Miasojedow. Analysis of langevin monte carlo via convex optimization. *The Journal of Machine Learning Research*, 20(1):2666–2711, 2019a.

Alain Durmus, Eric Moulines, et al. High-dimensional bayesian inference via the unadjusted langevin algorithm. *Bernoulli*, 25(4A):2854–2882, 2019b.

Raaz Dwivedi, Yuansi Chen, Martin J Wainwright, and Bin Yu. Log-concave sampling: Metropolis-hastings algorithms are fast. *Journal of Machine Learning Research*, 20(183):1–42, 2019.

Andreas Eberle, Arnaud Guillin, Raphael Zimmer, et al. Couplings and quantitative contraction rates for langevin dynamics. *The Annals of Probability*, 47(4):1982–2010, 2019.

J-P Eckmann and Martin Hairer. Spectral properties of hypoelliptic operators. *Communications in mathematical physics*, 235(2):233–253, 2003.

Murat A Erdogdu and Rasa Hosseinzadeh. On the convergence of langevin monte carlo: The interplay between tail growth and smoothness. *COLT*, 2021.

Charlie Frogner and Tomaso Poggio. Approximate inference with wasserstein gradient flows. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Ye He, Krishnakumar Balasubramanian, and Murat A Erdogdu. On the ergodicity, bias and asymptotic normality of randomized midpoint sampling method. *Advances in Neural Information Processing Systems*, 33, 2020.

Bin Hu and Laurent Lessard. Dissipativity theory for nesterov's accelerated method. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1549–1557. JMLR.org, 2017.

Chii-Ruey Hwang, Shu-Yin Hwang-Ma, Shuenn-Jyi Sheu, et al. Accelerating diffusions. *Annals of Applied Probability*, 15(2):1433–1444, 2005.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

Arlene KH Kim, Richard J Samworth, et al. Global rates of convergence in log-concave density estimation. *The Annals of Statistics*, 44(6):2756–2779, 2016.

S. M. Kozlov. Effective diffusion in the fokker-planck equation. *Mathematical notes of the Academy of Sciences of the USSR*, 45:360–368, 1989.

Benedict Leimkuhler, Charles Matthews, and Jonathan Weare. Ensemble preconditioning for markov chain monte carlo simulation. *Statistics and Computing*, 28(2):277–290, 2018.

Tony Lelievre, Francis Nier, and Grigorios A Pavliotis. Optimal non-reversible linear drift for the convergence to equilibrium of a diffusion. *Journal of Statistical Physics*, 152(2):237–274, 2013.

Ruilin Li, Hongyuan Zha, and Molei Tao. Sqrt (d) dimension dependence of langevin monte carlo. *arXiv preprint arXiv:2109.03839*, 2021.

Xuechen Li, Denny Wu, Lester Mackey, and Murat A Erdogdu. Stochastic Runge-Kutta accelerates Langevin Monte Carlo and beyond. *NeurIPS*, 2019.

Chang Liu, Jingwei Zhuo, Pengyu Cheng, Ruiyi Zhang, and Jun Zhu. Understanding and accelerating particle-based variational inference. In *International Conference on Machine Learning*, pp. 4082–4092, 2019.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in neural information processing systems*, pp. 2378–2386, 2016.

Yi-An Ma, Yuansi Chen, Chi Jin, Nicolas Flammarion, and Michael I Jordan. Sampling can be faster than optimization. *Proceedings of the National Academy of Sciences*, 116(42):20881–20885, 2019.

Yi-An Ma, Niladri Chatterji, Xiang Cheng, Nicolas Flammarion, Peter Bartlett, and Michael I Jordan. Is there an analog of nesterov acceleration for mcmc? *Bernoulli*, 2021.

Jonathan C Mattingly, Andrew M Stuart, and Desmond J Higham. Ergodicity for sdes and approximations: locally lipschitz vector fields and degenerate noise. *Stochastic processes and their applications*, 101(2):185–232, 2002.

Robert I McLachlan and G Reinout W Quispel. Splitting methods. *Acta Numerica*, 11:341, 2002.

Sean P Meyn, Robert L Tweedie, et al. Computable bounds for geometric convergence rates of markov chains. *The Annals of Applied Probability*, 4(4):981–1011, 1994.

Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2). In *Doklady AN USSR*, volume 269, pp. 543–547, 1983.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Masayuki Ohzeki and Akihisa Ichiki. Langevin dynamics neglecting detailed balance condition. *Physical Review E*, 92(1):012105, 2015.

Grigorios A Pavliotis. *Stochastic processes and applications: diffusion processes, the Fokker-Planck and Langevin equations*, volume 60. Springer, 2014.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Luc Rey-Bellet and Konstantinos Spiliopoulos. Irreversible langevin samplers and variance reduction: a large deviations approach. *Nonlinearity*, 28(7):2081, 2015.

Gareth O Roberts, Richard L Tweedie, et al. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

Julien Roussel and Gabriel Stoltz. Spectral methods for langevin dynamics and associated error estimates. *ESAIM: Mathematical Modelling and Numerical Analysis*, 52(3):1051–1083, 2018.

Ruoqi Shen and Yin Tat Lee. The randomized midpoint method for log-concave sampling. In *Advances in Neural Information Processing Systems*, pp. 2098–2109, 2019.

Bin Shi, Simon S Du, Michael I Jordan, and Weijie J Su. Understanding the acceleration phenomenon via high-resolution differential equations. *Mathematical Programming*, pp. 1–70, 2021.

Andre N Souza and Molei Tao. Metastable transitions in inertial langevin systems: What can be different from the overdamped case? *European Journal of Applied Mathematics*, 30(5):830–852, 2019.

Gilbert Strang. On the construction and comparison of difference schemes. *SIAM journal on numerical analysis*, 5(3):506–517, 1968.

Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pp. 2510–2518, 2014.

Amirhossein Taghvaei and Prashant Mehta. Accelerated flow for probability distributions. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6076–6085, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices. In *Advances in Neural Information Processing Systems*, pp. 8092–8104, 2019.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Cédric Villani. Hypocoercivity. *Memoirs of the American Mathematical Society*, 202(950), 2009.

Yifei Wang and Wuchen Li. Accelerated information gradient flow. *arXiv preprint arXiv:1909.02102*, 2019.

Andre Wibisono. Sampling as optimization in the space of measures: The langevin dynamics as a composite optimization problem. In *Conference On Learning Theory*, pp. 2093–3027, 2018.

Andre Wibisono, Ashia C Wilson, and Michael I Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.

Andrew Gordon Wilson. The case for bayesian deep learning. *arXiv preprint arXiv:2001.10995*, 2020.

Ashia C Wilson, Ben Recht, and Michael I Jordan. A lyapunov analysis of accelerated methods in optimization. *Journal of Machine Learning Research*, 22(113):1–34, 2021.

Ruiyi Zhang, Changyou Chen, Chunyuan Li, and Lawrence Carin. Policy optimization as wasserstein gradient flows. In *International Conference on Machine Learning*, pp. 5737–5746, 2018.