Skew-symmetrically perturbed gradient flow for convex optimization

Futoshi Futami FUTOSHI.FUTAMI.UK@HCO.NTT.CO.JP
Tomoharu Iwata TOMOHARU.IWATA.GY@HCO.NTT.CO.JP
Naonori Ueda NAONORI.UEDA.FR@HCO.NTT.CO.JP
Communication Science Laboratories, NTT, KYOTO, JAPAN

Ikko Yamane IKKO.YAMANE@DAUPHINE.PSL.EU
LAMSADE, CNRS, Université Paris-Dauphine, PSL Research University, PARIS/ RIKEN AIP, TOKYO, JAPAN

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

Recently, many methods for optimization and sampling have been developed by designing continuous dynamics followed by discretization. The dynamics that have been used for optimization have their corresponding underlying functionals to be minimized. On the other hand, a wider class of dynamics have been studied for sampling, which is not necessarily limited to functional minimization. For example, dynamics perturbed with skew-symmetric matrices, which cannot be seen as minimization of functionals, have been widely used to reduce asymptotic variance. Following this success in sampling, exploring such perturbed dynamics in the context of optimization can open a new avenue to optimization algorithm design. In this work, we introduce a perturbation technique for sampling into optimization for strongly convex functions. We show that perturbation applied to the gradient flow yields rapid convergence in optimization for strongly convex functions. Based on this continuous dynamics, we propose an optimization algorithm for strongly convex functions with a novel discretization framework that combines the Euler method with the leapfrog method which is used in the Hamilton Monte Carlo method. Our numerical experiments show that the perturbation technique is useful for optimization.

Keywords: Convex optimization, skew-symmetric matrices, gradient flow, discretization

1. INTRODUCTION

Analysis of continuous dynamics and discretization methods has been a driving force in recent developments in optimization and sampling. In optimization, inspired by the relation between the gradient flow as continuous dynamics and the gradient descent as discretized dynamics (Scieur et al., 2017), acceleration methods such as Nesterov's scheme have been analyzed as second-order differential equations (Su et al., 2014). Recent analysis showed that the various first-order optimization methods are closely related to continuous dynamics and discretization methods (Scieur et al., 2017). Zhang et al. (2018) and Shi et al. (2019) showed that using the high-order discretization results in acceleration.

For sampling, Wibisono (2018) analyzed the Langevin dynamics (LD) as a gradient flow in the space of probability measures and proposed a method for discretizing continuous dynamics based on a technique used in optimization. Motivated by this connection, many useful optimization techniques have been introduced into sampling (e.g., Durmus and Majewski (2019)). In particular, Muehlebach and Jordan (2019) introduced the continuous dynamics, which is used in optimization,

into sampling for acceleration. In this way, studying the continuous dynamics and discretization methods in optimization has brought significant advances in recent efficient sampling algorithms.

Most continuous dynamics designed for optimization have their corresponding underlying functionals to be minimized. For example, the gradient flow and the second-order differential equations for acceleration are derived through minimization of the Bregman Lagrangian (Wibisono et al., 2016). On the other hand, designing the dynamics in sampling is not limited to minimizing functionals. For example, a *perturbation* approach that adds a small perturbation composed of a skew-symmetric matrix to the original LD has been gathering attention (Hwang et al., 2005, 2015; Duncan et al., 2016, 2017a; Kaiser et al., 2017). This perturbation technique never changes the stationary distribution but reduces the asymptotic variance for sampling. An interesting point of this perturbed dynamics is that it cannot be seen as a minimization of a functional.

Based on this success of the perturbation technique in sampling, we expect that understanding perturbed dynamics in the context of optimization may pave a new avenue for designing optimization algorithms. In this paper, we show, for the first time to the best of our knowledge, that such perturbed continuous dynamics are also useful when optimizing strongly convex functions.

However, when we adopt such perturbed dynamics into optimization, two major challenges arise. First, the advantage of the perturbation in optimization is unclear although this technique reduces the *asymptotic variance* in sampling. In optimization, we adopt the final state of a parameter as a solution, and thus the asymptotic variance is not even defined. Second, it is not obvious what kind of discretization is preferable for such perturbed dynamics. Existing work on perturbed dynamics in sampling only focused on continuous dynamics since the obtained samples can be adjusted by Metropolis-Hasting steps (Bishop, 2006).

We address the above challenges and show that the convergence rate of perturbed dynamics is improved compared to un-perturbed dynamics in the continuous and discrete-time settings.

First, we present new continuous dynamics using skew-symmetric matrices that converge more rapidly than the gradient flow under mild conditions. To show faster convergence, we analyze the perturbed Hessian matrix. Since it is neither symmetric nor skew-symmetric, it remains unclear whether diagonalization is possible. We clarify what kind of perturbation preserves the diagonalization of the Hessian matrix and then show the largest and smallest eigenvalues are changed by perturbation. This leads to faster convergence of the continuous dynamics.

Second, we provide a novel discretization method for the proposed dynamics and analyze its convergence properties. We show that a simple Euler method cannot guarantee faster convergence. To achieve faster convergence, inspired by Hamilton Monte Carlo (Bishop, 2006), we propose a new discretization method that combines the Euler and leapfrog methods to effectively exploit the particular structure of skew-symmetric matrices. Finally, we present methods for tuning the hyper-parameters of our proposed method, including those of the skew-symmetric perturbations.

2. PRELIMINARIES

In this section, we briefly introduce the gradient flow, gradient descent, and the perturbation technique in sampling.

2.1. Gradient flow and gradient descent

Consider a strongly convex loss function $F(x)$ on $\mathbb{R}^d$. We assume that F is an m-strong and M-smooth function. To minimize $F(x)$, we consider the gradient flow:

$$\frac{dx(t)}{dt} = -\nabla_x F(x(t)), \quad x(0) = x_0, \tag{1}$$

for which one can show

$$\|x(t) - x^*\| \leq e^{-mt}\|x_0 - x^*\|, \tag{2}$$

which ensures the convergence to the optimal point $x^* = \arg\min_{x\in\mathbb{R}^d} F(x)$. In many cases, we approximate Eq. (1) with a discretization method since we cannot directly implement it due to its continous nature. A widely used method is the gradient descent (GD). We use x_k to express a candidate of a solution obtained from the k-th iterate of the GD. Then, the GD algorithm is given by recursion:

$$x_{k+1} = x_k - \eta\nabla_x F(x_k), \tag{3}$$

where $\eta > 0$ is the step size. The convergence behavior is characterized as follows. GD converges if the step size satisfies $\eta \in [0, \frac{2}{M})$. Furthermore, if $\eta = \frac{2}{M+m}$, we have

$$\|x_k - x^*\| \leq e^{-\frac{m}{M}k}\|x_0 - x^*\|. \tag{4}$$

Based on the definition of strong convexity and smoothness, we can regard m and M as an upper bound and a lower bound of eigenvalues of Hessian matrix $H_x = \nabla_x^2 F(x)$ for all x. Equivalently, $H_x \succeq mI$ and $MI \succeq H_x$ hold, where I is the $d \times d$ identity matrix. Hereinafter, we simply express H_x as H. Then, the convergence of the GD is charactderized by the ratio of the largest and smallest eigenvalue of Hessian matrix. Thus, analyzing the properties of Hessian matrix is important to understand the convergence behavior.

2.2. Perturbation techinque in sampling

A perturbation to the LD is used for sampling from Gibbs distribution $\pi(x) \propto e^{-U(x)}$, where $U : \mathbb{R}^d \to \mathbb{R}$ is a potential function. Let X_t denote a random variable on $\mathbb{R}^d$ and let W denote the Wiener process. Then the perturbation to the LD is given by

$$dX_t = -(I + J)\nabla U(X_t)dt + \sqrt{2}dW, \tag{5}$$

where J is a skew-symmetric matrices that satisfies and $J = -J^\top$, and I is the identity matrix. The stationary distribution of this dynamics is $\pi(x)$. Compared to the standard LD, which corresponds to $J = 0$, the perturbed dynamics shows smaller variance in the asymptotic limit. The perturbation of J changes the smallest real part of the eigenvalue of the infinitesimal generator of Eq. (5), which is larger than the standard LD. See the following works for details: Hwang et al. (2005, 2015); Duncan et al. (2016, 2017a); Kaiser et al. (2017); Futami et al. (2020, 2021). Intuitively, the change of the eigenvalue of the generator indicates that, if U is a strongly convex function, the smallest real part of the eigenvalue of $(I + J)\nabla^2 U$ is larger than that of $\nabla^2 U$. Lelièvre et al. (2013) showed that when ∇U is a linear function, the optimal J improves the smallest and largest real part of the eigenvalue of $(I + J)\nabla^2 U$ to $\mathrm{Tr}(\nabla^2 U)/d$.

3. PROPOSED METHOD

In this section, we first present continuous dynamics. Then, we propose two types of discretization methods and an algorithm to tune the hyper-parameters.

3.1. Theoretical properties of the perturbed Hessian matrix

Inspired by the perturbation in sampling, we incorporate a skew-symmetric matrix J to the gradient term in the gradient flow:

$$\frac{dx(t)}{dt} = -\nabla_x F(x(t)) - \alpha J \nabla_x F(x(t)), \tag{6}$$

where $\alpha \in \mathbb{R}$ expresses the strength of the perturbation and J satisfies

$$J^\top = -J, \qquad \|J\|_F \leq d, \tag{7}$$

where $\|\cdot\|_F$ is the Frobenius norm. We call this dynamics *skew-symmetrically perturbed gradient flow*. Inspired by the perturbation to the LD, we expect that introducing the skew-symmetric matrix changes the eigenvalues of Hessian matrix and leads to rapid convergence. We denote the original Hessian matrix by $H = \nabla^2 F$ and the perturbed Hessian matrix by $H' = (I + \alpha J)H$. To analyze the skew-symmetrically perturbed gradient flow, we need to understand H' by elucidating the following three factors: 1) whether the stationary point of the perturbed dynamics is x^*, 2) the condition in which H' is diagonalizable, and 3) the condition in which the eigenvalues are improved. In this section, we assume that J is a general skew-symmetric matrix that satisfies Eq. (7). We discuss the concrete algorithm to generate J that has nice properties in Section 3.4.

Stationary point: First, we study question 1) by analyzing the stationary point of the perturbed dynamics. From Eq. (6) and the property of the optimal point $\nabla_x F(x^*) = 0$, it is clear that x^* is also the stationary point of the perturbed dynamics. Furthermore, we can show that x^* remains the unique stationary point that satisfies $(I + \alpha J)\nabla_x F(x^*) = 0$ since $(I + \alpha J)$ has an inverse matrix; see Appendix C.

Diagonalization: Next, we discuss the diagonalizability of H'. We emphasize that the diagonalizability is an important property for convergence analysis of the continuous dynamics (see Appendix D for details). Although without the diagonalizability, we can analyze the dynamics by Jordan decomposition, it produces unsatisfactory constants in the convergence bound. Since H is a real-valued symmetric matrix, it is always diagonalizable. On the other hand, since H' is not symmetric, the diagonalizability is not assured. Remarkably, the following proposition provides the practical guarantee for the diagonalization of J'.

Proposition 1 *Suppose that J is a random matrix whose upper triangular entries follow a probability distribution that is absolutely continuous with respect to the Lebesgue measure. Then, $(I + \alpha J)H$ is diagonalizable with probability 1.*

Improvement of eigenvalues: We discuss how the real parts of the eigenvalues of H' are changed from those of H. Denote the pairs of the eigenvectors and the eigenvalues of H' as $\{(v_i^\alpha(x), \lambda_i^\alpha(x))\}_{i=1}^d$. Order them as $\mathrm{Re}(\lambda_1^\alpha(x)) \leq \cdots \leq \mathrm{Re}(\lambda_d^\alpha(x))$. Thus, the eigenvectors and eigenvalues of H are expressed by $\{(v_i^0(x), \lambda_i^0(x))\}_{i=1}^d$. Let $m' \coloneqq \inf_{x\in\mathbb{R}^d} \mathrm{Re}(\lambda_1^\alpha(x))$ and $M' \coloneqq \sup_{x\in\mathbb{R}^d} \mathrm{Re}(\lambda_d^\alpha(x))$. These m' and M' can be regarded as the modified constants of (m, M) of the objective $F(x)$. The following proposition describes the relation between the eigenvalues:

Proposition 2 *For all x, the real parts of the eigenvalues of $(I+\alpha J)H$ satisfy*

$$\lambda_1^0(x) \leq \mathrm{Re}\left(\lambda_1^\alpha(x)\right) \leq \cdots \leq \mathrm{Re}\left(\lambda_d^\alpha(x)\right) \leq \lambda_d^0(x). \tag{8}$$

In addition, denote the set of the eigenvectors of eigenvalue $\lambda_1^0(x)$ as V_1^0. Let us denote the size of V_1^0 as $|V_1^0|$. If the following condtions are satisfied, then we have $\lambda_1^0(x) = \mathrm{Re}\left(\lambda_1^\alpha(x)\right)$:

$$\begin{cases} |V_1^0| = 1, \text{ and } v \in V_1^0,\ Jv = 0, \\ |V_1^0| > 1, \text{ and for any } v, v' \in V_1^0,\ \lambda_1^0(x)\alpha Jv = (\mathrm{Im}\left(\lambda_1^\alpha(x)\right))v' \text{ and } \lambda_1^0(x)\alpha Jv' = -(\mathrm{Im}\left(\lambda_1^\alpha(x)\right))v. \end{cases} \tag{9}$$

We have similar sufficient conditions for $\lambda_d^0(x) = \mathrm{Re}\left(\lambda_d^\alpha(x)\right)$. Furthurmore, the following relation holds:

$$\mathrm{Re}\left(\lambda_1^\alpha(x)\right) \leq \mathrm{Tr}H/d \leq \mathrm{Re}\left(\lambda_d^\alpha(x)\right). \tag{10}$$

Thus, from the above proposition, we have $m \leq m'$ and $M' \leq M$ by definition. Moreover, if α is small enough, we can evaluate the change of the largest and smallest eigenvalues:

Proposition 3 *Suppose H has d distinct eigenvalues. With the same notation as in Proposition 2, for all x and for any $i \in \{1, \ldots, d\}$, we have*

$$\mathrm{Re}\left(\lambda_i^\alpha(x)\right) = \lambda_i^0(x) + \alpha^2 \sum_{k=1, k\neq i}^{d} \frac{|v_k^0(x) J v_i^0(x)|^2}{\lambda_k^0(x) - \lambda_i^0(x)} + \mathcal{O}(\alpha^3). \tag{11}$$

The proof is shown in Appendix C.4. Note that the first-order term in α is zero owing to the skew-symmetric property of J. From this proposition, for example,

$$\mathrm{Re}\left(\lambda_1^\alpha(x)\right) = \lambda_1^0(x) + \alpha^2 \sum_{k=2}^{d} \frac{|v_k^0(x) J v_1^0(x)|^2}{\lambda_k^0(x) - \lambda_1^0(x)} + \mathcal{O}(\alpha^3) \tag{12}$$

holds up to the second order. Since for all $k \geq 1$, $\lambda_k^0(x) > \lambda_1^0(x)$ holds, the second term above is positive, indicating $\mathrm{Re}\left(\lambda_1^\alpha(x)\right) > \lambda_1^0(x)$ for any sufficiently small α. Similarly, $\mathrm{Re}(\lambda_d^\alpha(x)) < \lambda_d^0(x)$ holds for any sufficiently small α.

3.2. Continuous dynamics

Based on the above analysis, since the largest and smallest eigenvalues are improved by introducing the skew-symmetric matrix, we expect that it will improve the convergence speed of the dynamics. We present our first main theorem that describes the effect of the skew-symmetric gradient on convergence.

Proposition 4 *If $(I+\alpha J)H$ is diagonalizable, the convergence of Eq. (6) is*

$$\|x(t) - x^*\| \leq e^{-m't}\|x_0 - x^*\|. \tag{13}$$

We outline the proof since it includes an important property of continuous dynamics.
Proof (Outline) Let $r(t) := x(t) - x^*$ and define a functional $\mathcal{L}(t) = r(t)^\top r(t)$. Then

$$\frac{d\mathcal{L}}{dt} = -2r(t)^\top (I+\alpha J)\left(\nabla F(x(t)) - \nabla F(x^*)\right) = -2\int_0^1 r(t)^\top (I+\alpha J) H(\bar{x}(\tau)) r(t) d\tau,$$

where $\bar{x}(\tau) := x^* + \tau(x(t) - x^*)$. We used the Taylor expansion and expressed its residual by the integral. Since $(I + \alpha J)H$ is diagonalizable, we can analyze the dynamics based on the eigenvalue decomposition. We assume that it has l real eigenvalues $\lambda'_1, \dots, \lambda'_l$ and $2m$ complex eigenvalues, $\mu_1 = \alpha_1 \pm i\beta_1, \dots, \mu_m = \alpha_m \pm i\beta_m$. Thus, $d = l + 2m$. We express the corresponding eigenvectors as $\{v_j\}_{j=1}^l$ for the real eigenvalues and $\{w_j = a_j + ib_j\}_{j=1}^m$ for the complex eigenvalues. We express the corresponding conjugate eigenvectors as $\{\bar{w}_j\}$. Define $d \times d$ matrix V as

$$V = [v_1, \dots, v_l, a_1, b_1, \dots, a_m, b_m]. \tag{14}$$

Then, we decompose $(I + \alpha J)H(x(\tau))$ into a block diagonal matrix known as the Jordan canonical form (Golub and Van Loan, 2012);

$$(I + \alpha J)H(x^* + \tau(x(t) - x^*)) = VDV^{-1}, \tag{15}$$

where

$$D = \underbrace{\begin{pmatrix} \lambda'_1 & & & & & & \\ & \ddots & & & & & \\ & & \lambda'_l & & & & \\ & & & \alpha_1 & 0 & & \\ & & & 0 & \alpha_1 & & \\ & & & & & \ddots & \\ & & & & & \alpha_m & 0 \\ & & & & & 0 & \alpha_m \end{pmatrix}}_{D_1} + \underbrace{\begin{pmatrix} 0 & & & & & & \\ & \ddots & & & & & \\ & & 0 & & & & \\ & & & 0 & \beta_1 & & \\ & & & -\beta_1 & 0 & & \\ & & & & & \ddots & \\ & & & & & 0 & \beta_m \\ & & & & & -\beta_m & 0 \end{pmatrix}}_{D_2}. \tag{16}$$

Then,

$$\begin{aligned} \frac{d\mathcal{L}}{dt} = -2\int_0^1 r(t)^\top VDV^{-1}r(t)d\tau &= -2\int_0^1 r(t)^\top V(D_1 + D_2)V^{-1}r(t)d\tau \\ &= -2\int_0^1 r(t)^\top VD_1V^{-1}r(t)d\tau \\ &\le -2\mathrm{Re}(\lambda_1^\alpha(x(\tau)))r(t)^\top r(t), \end{aligned} \tag{17}$$

where $\mathrm{Re}(\lambda_1^\alpha(x(\tau))) = \min\{\lambda'_1, \dots, \lambda'_l, \alpha_1, \dots, \alpha_m\}$. We used the skew-symmetric property of D_2 and applied the Gronwall inequality to obtain the proposition. ■

Compared to the standard GD in Eq. (2), the acceleration is confirmed since $m' \ge m$ holds. This improvement can be quantified by Eq.(12). In the proof, the key factor is the skew-symmetric property of D_2, with which we can eliminate the imaginary part of the eigenvalue from the convergence rate. See Appendix D.3 for details.

3.3. Discretization

We need to discretize the continuous dynamics to implement it. In this section, we first observe that the Euler discretization degrades the convergence rate and propose a discretization scheme that integrates the Euler and leapfrog methods to overcome this issue.

Euler discretization: First, the Euler discretization is given by

$$x_{k+1} = x_k - \eta(I + \alpha J)\nabla_x F(x_k), \tag{18}$$

where η is a stepsize. The convergence behavior is analyzed in the following way.

Proposition 5 *Define* $r := \alpha \max_i \left(\sum_{j=1}^{d} |J_{ij}| \right)$. *Suppose that in Eq. (18),* $(I + \alpha J)H$ *is diagonalizable. Also suppose that* α *and* η *satisfy* $r \leq m'$ *and* $\eta \in (0, \frac{2}{M'+r}]$. *Then,* x_k *converges to* x^* *as* $k \to \infty$ *and the rate of convergence is*

$$\|x_k - x^*\| \leq e^{-\frac{m'-r}{M'}k} \|x_0 - x^*\|. \tag{19}$$

The proof shown in Appendix E is outlined here.

Proof (Outline) From the discretized dynamics Eq. (18), subtract x^* from both sides and define $r_k := x_k - x^*$. Then, we have

$$\|r_{k+1}\| = \|r_k - \eta(I + \alpha J)\nabla F(x_k)\|. \tag{20}$$

We define $h(x) := x - \eta(I + \alpha J)\nabla F(x)$. The above equation can be expressed:

$$\|r_{k+1}\| = \|h(x_k) - h(x^*)\|. \tag{21}$$

Then, we apply the mean-value theorem. There exists a point $\xi_k = (1-\beta)x_k + \beta x^*,\ \beta \in [0, 1) \subset \mathbb{R}$ (expressed by $\xi_k \in [x_k, x^*) \subset \mathbb{R}^d$ for simplicity), such that

$$\|r_{k+1}\| \leq \| \left(I - \eta(I + \alpha J)H(\xi_k)\right) \| \|r_k\|. \tag{22}$$

Here we used the operator norm $\| \cdot \|$ defined by

$$\|M\| := \sup_{x \neq 0} \frac{\|Mx\|}{\|x\|} = \|M^\dagger M\|^{1/2} = s(M), \tag{23}$$

for any matrix M, where $s(M)$ is M's largest singular value.

To bound $\| \left(I - \eta(I + \alpha J)H\right) r_k \|$, we evaluate the singular value of $H' = I - \eta(I + \alpha J)H$. Note that from the Jordan canonical form,

$$H' = I - \eta V D V^{-1} = I - \eta V D_1 V^{-1} - \eta V D_2 V^{-1} \tag{24}$$

holds. We define $P = I - \eta V D_1 V^{-1}$ and $Q = -\eta V D_2 V^{-1}$. The largest singular value of H' (denoted by $s(H')$) is upper bounded by the largest singular values of P and Q (Bhatia, 2013) (denoted as $s(P)$ and $s(Q)$),

$$s(H') \leq s(P) + s(Q). \tag{25}$$

Note that $s(P)$ and $s(Q)$ depend on η. Thus, all we need is to bound each term. The remaining part of the proof is shown in Appendix E. ■

The key factor is that the convergence rate depends on the singular value. Given a matrix that has complex eigenvalues, its singular values depend on both the real and imaginary parts of the eigenvalues. Thus, unlike the continuous dynamics, discretized dynamics is characterized by both the real and imaginary parts of the eigenvalues. Since propositions 4 and 5 suggest a large gap between the Euler discretization and continuous dynamics, the convergence rate of Eq. (18) is not always improved compared to that of the GD.

We can intuitively understand why the Euler discretization does not work well by focuing on J. Consider the change in $F(x(t))$ in the continuous case:

$$\frac{dF(x(t))}{dt} = \nabla F(x(t))^\top \frac{dx(t)}{dt} = -\|\nabla F(x(t))\|^2 - \alpha \nabla F(x(t))^\top J \nabla F(x(t)) = -\|\nabla F(x(t))\|^2. \tag{26}$$

The value of F is preserved for J due to the skew-symmetric property. Although such preservation is a critical property, the Euler method does not take it into consideration.

Euler-leapfrog discretization: To exploit the preservation property of J, we propose a new discretization method that combines the Euler and the leapfrog methods, which is widely used in Hamilton Monte Carlo (Bishop, 2006). We split the dynamics into two parts. One is related to J, and we discretize it by the leapfrog method. The other part is unrelated to J, and we discretize it by the Euler method. To implement the leapfrog method, we introduce auxiliary variable $y_k \in \mathbb{R}^d$ and optimize augmented objective function $\tilde{F}(x, y) = F(x) + \frac{1}{2c}\|y\|^2$ where c is a positive constant, whose condition is described in Proposition 6. Then, we update $\{x_k\}$ and $\{y_k\}$ by

$$\begin{cases} x_{k+\frac{1}{2}} = x_k - \frac{\eta\alpha}{c} J y_k, \\ y_{k+\frac{1}{2}} = y_k - \eta\alpha J \nabla F(x_{k+\frac{1}{2}}), \end{cases} \tag{27}$$

$$\begin{cases} y_{k+1} = y_{k+\frac{1}{2}} - \frac{\eta}{c} y_{k+\frac{1}{2}}, \\ x_{k+1} = x_{k+\frac{1}{2}} - \eta \nabla F(x_{k+\frac{1}{2}}). \end{cases} \tag{28}$$

Eq. (27) corresponds to the leapfrog step, which discretizes the dynamics related to J. Eq. (28) corresponds to the Euler step, which discretizes the gradient flow. We call this the Euler-leapfrog (ELF) discretization. In Appendix F.4, we compared the ELF method with other discretization methods. Before we present a formal statement, we explain the intuition of our ELF method in matrix form:

$$\left\| \begin{pmatrix} x_{k+1} - x^* \\ y_{k+1} - y^* \end{pmatrix} \right\| \leq \left\| \underbrace{\begin{pmatrix} -\eta H(\xi_{k+\frac{1}{2}}) & 0 \\ 0 & (1-\eta/c)I \end{pmatrix}}_{=\tilde{H}(c,\eta)} \underbrace{\begin{pmatrix} I & -\eta\alpha c^{-1} J \\ -\eta\alpha J H(\xi_{k+\frac{1}{2}}) & I + \eta^2\alpha^2 c^{-1} J H(\xi_{k+\frac{1}{2}}) J \end{pmatrix}}_{=L(\eta,c,\alpha,J)} \right\| \left\| \begin{pmatrix} x_k - x^* \\ y_k - y^* \end{pmatrix} \right\|, \tag{29}$$

where $\xi_{k+\frac{1}{2}} \in [x_{k+\frac{1}{2}}, x^*)$ is a constant in $\mathbb{R}^d$, specified by the mean-value theorem; see Appendix F.1 for details. In Eq. (29), $\tilde{H}$ corresponds to the Euler step of Eq. (28) and L corresponds to the leapfrog step of Eq. (27). If we appropriately select α, the singular values of L will be 1. This is the characteristic property of the leapfrog step. From the submultiplicativity of the matrix norm, $\|\tilde{H}L\| \leq \|\tilde{H}\|\|L\| \leq \|\tilde{H}\| = 1 - \frac{m}{M}$ under appropriate conditions for η and c. Furthermore, if we generate J following the rules described in Section 3.4, the ELF method will converge faster than the GD. Summarizing these results, we have the following theorem, whose proof is shown in Appendix F.1:

Proposition 6 *In Eqs. (27),(28), if η, c, and α satisfy $\eta \in (0, \frac{2}{M}]$, $c^{-1} \in (0, \frac{2}{\eta}]$, and $\alpha^2 \leq 4c(\eta^2 M s_d^2)^{-1}$, where s_d is the largest singular value of J, x_k converges to x^* as $k \to \infty$. If we set $\eta = \frac{2}{m+M}$, $c^{-1} \in (m, M]$, and $\alpha < 2\sqrt{c(\eta^2 M s_d^2)^{-1}}$,*

$$\|x_k - x^*\| \leq e^{-\kappa(\alpha,m,M,c,J)k}\|x_0 - x^*\| \tag{30}$$

holds for positive constant $\kappa(\alpha, m, M, c, J)$ that satisfies $\kappa(\alpha, m, M, c, J) \geq 2m/(m + M)$. If $m \neq M$ and $\ker J = \{0\}$ are satisfied, then $\kappa(\alpha, m, M, c, J) > 2m/(m + M)$ holds.

From above proposition, if we choose hyperparameters appropriately, the ELF method shows faster convergence than gradient descent.

3.4. Tuning hyper-parameters for the ELF method

Here, we present an algorithm to tune J, α, and c in the ELF method to satisfy conditions of Propositions 1 and 6. Detailed explanation of the algorithm is shown in Appendix F.2. We assume that $m \neq M$. Our proposed algorithm is summarized in Algorithm 1 and its theoretical property is shown in Theorem 7.

First, we discuss how to generate J. Lelièvre et al. (2013) obtained the optimal J when the drift function is linear under continuous time. However, to get the optimal J, we require $\mathrm{O}(d^3)$ time per iteration, which is computationally demanding. Such J may cause numerical instability for discretized dynamics shown in Section 5. Instead, we propose using a random matrix for J, as suggested from Proposition 1, and fix it during the optimization to reduce the computational cost. Although this choice is not optimal, it successfully alters the trajectory and improves the convergence rate but does not cause the numerical instability due to the large singular values. See Section 5 for details.

J needs to be generated to satisfy the assumption of Proposition 1. We also want to ensure the condition for $\kappa > 2m/(m+M)$ in Proposition 6 so that acceleration will occur. Also, from Proposition 6, $\ker J = \{0\}$ is a sufficient condition for that. To satisfy $\ker J = \{0\}$, we first generate matrix J' wherein the upper triangular entries $(i < j)$ are

$$J'_{ij} = \begin{cases} 1 + \rho_{ij}/d \quad \rho_{ij} \sim \mathcal{N}(0, \epsilon) & \text{if } i \text{ is odd and } i = j+1, \\ \rho_{ij}/d \quad \rho_{ij} \sim \mathcal{N}(0, \epsilon) & \text{otherwise,} \end{cases} \tag{31}$$

where $\mathcal{N}(0, \epsilon)$ denotes the zero-mean Gaussian distribution with small variance ϵ. For example, we set $\epsilon = 10^{-4}$ in numerical experiments. Finally, we set J as $J = J'^{\top} - J'$. This J is diagonalizable, and the eigenvalues are very close to $\pm i$ if d is even, which means that $\ker J = \{0\}$. If d is odd, however, eigenvalues are $\pm i$ and 0, which implies that $\ker J \neq \{0\}$. To resolve this problem with the case of odd d, a simple idea is to introduce dummy variable $\tilde{x}$ so that the dimension of the objective function will be even. Then, we optimize $F(x, \tilde{x}) = F(x) + \gamma \|\tilde{x}\|^2$ where γ is a positive constant. We can use $\gamma = \frac{1}{2c}$ so that the convergence rate of $F(x, \tilde{x})$ is dominated by the original $F(x)$.

Next, we set c as $c^{-1} = \frac{\|\nabla f(x) - \nabla f(y)\|}{\|x - y\|}$, where x and y are arbitrary distinct points, e.g., those chosen from the initial point and its neighborhood. Condition $m \leq c^{-1} \leq M$ in Proposition 6 holds by definition.

Finally, α must satisfy $0 < \chi < 4$, where $\chi := \eta^2 \alpha^2 c^{-1} s_d^2 M$, which is required for the ELF method to accelerate the convergence. We also empirically observed that the ELF method works well with χ around 1. Based on these insights, we set α so that $\alpha^2 = \frac{c}{2\eta s_d^2}$, which ensures $0 < \chi \leq 1 < 4$ since $\eta M \leq 2$. From the construction of J, the largest singular value s_d is upper bounded by $s_d^2 \leq \max_i (1 + \sum_{j \neq i} |J_{ij}|/d)^2$ from the Gerchgorin theorem (Golub and Van Loan, 2012). We use this as an estimate of s_d^2. In practice, setting η to a large value is advisable within condition $\eta M \leq 2$ so that χ will be close to 1. Summarizing the above discussions, we generate α and J by Algorithm 1, analyzed by the following proposition.

Proposition 7 *Suppose that d is even, $m \neq M$, and J and α are generated by Algorithm 1. Then, with high probability, the conditions of Proposition 6 are satisfied and $\kappa > \frac{2m}{m+M}$ holds.*

The detailed proof is shown in Appendix F.3. This proposition guarantees that the ELF shows better convergence than the GD with high probability. We confirm that matrix J obtained by Algorithm 1

Algorithm 1 Tuning hyperparameters α and J

1: **Input:** η, c, ϵ (e.g., $\epsilon = 10^{-4}$)
2: **Output:** α, J
3: Make a random matrix J' by Eq. (31).
4: Calculate $J = J'^{\top} - J'$
5: Calculate $s_d^2 = \max_i (1 + \sum_{j \neq i} |J_{ij}|/d)^2$
6: Set $\alpha = \sqrt{c(2\eta s_d^2)^{-1}}$

indeed shows better convergence behavior in numerical experiments. When d is odd, we solve $F(x, \tilde{x}) = F(x) + \frac{1}{2c}\|\tilde{x}\|^2$, where $\tilde{x}$ is a dummy variable so that we can apply Proposition 7.

Compared to the optimal J that requires $\mathcal{O}(d^3)$ (Lelièvre et al. (2013)), the calculation cost of Algorithm 2 is $\mathcal{O}(d)$ time. Our hyper-parameter tuning also works even in nonlinear dynamics, although the optimal J given by Lelièvre et al. (2013) can only be applicable to linear drift functions. In the numerical experiments in Section 5, we observed that the optimal J of Lelièvre et al. (2013) is unstable for discretized dynamics. When implementing the ELF discretization, we can re-use the gradient calculation in Eqs. (27) and (28), and thus, the computation cost of the ELF method is not much larger than that of the Euler discretization.

4. DISCUSSION AND RELATED WORK

In this section, we discuss the relationship between our proposed method, perturbation technique in sampling, and other optimization methods.

4.1. Relation to perturbation technique in sampling

Although our work is inspired by perturbation technique in sampling (Hwang et al., 2005, 2015; Duncan et al., 2016, 2017a; Kaiser et al., 2017), it is different in the sense that we focused on the property of J and discretizations. For the first time, our work propose using a random matrix for J and analyze the desirable conditions. We present a concrete algorithm to construct J in the ELF method. No previous work has considered the relation between J and discretization methods. Lelièvre et al. (2013) derived the optimal J, but it is limited to linear dynamics and is computationally demanding. Our numerical experiments in Section 5 also show that such an optimal J causes a numerical issue for discretized dynamics. Although Duncan et al. (2017b) worked on the splitting method, they focused on its asymptotic behavior with a general skew-symmetric matrix.

4.2. Relation to preconditioning methods

Our methods can be understood as preconditioning schemes. One of the most successful preconditioning methods is Newton's method and its approximations. These methods take metric information into consideration and multiply the inverse of Hessian matrix to the gradient. Thus, the gradient of each dimension is re-scaled, and the condition number of these dynamics becomes one in the re-scaled space. See Appendix G for details. However, since calculating such inverse matrices is computationally demanding, many variants of methods have been established.

Our proposed dynamics correlate different dimension by skew-symmetric matrices, and the perturbed Hessian matrix shows that the smallest real part of the eigenvalue is larger than that of the un-perturbed Hessian matrix. This results in a faster convergence compared to the un-perturbed dynamics and makes the trajectory smoother than the GD. See Section 5. As Lelièvre et al. (2013)

argued, for linear dynamics, we can construct optimal J, and the smallest and largest real parts of the eigenvalues become $\mathrm{Tr}H/d$, which means that the condition number becomes one, which is the same as Newton's method. Concerning the computational cost, our methods need an additional matrix and a vector product computation, which is usually much smaller than Newton's method.

4.3. Condition number and ℓ_2 regularization

Our method and ℓ_2 regularization are similar in the sense that the smallest and largest eigenvalues of the Hessian matrix change. If $\gamma \in \mathbb{R}^+$ is a regularization parameter, then objective function $F(x) + \gamma\|x\|^2$ is a $(m+\gamma)$-strong convex and a $(M+\gamma)$-smooth function. Thus, the condition number becomes $\frac{M+\gamma}{m+\gamma}$. This indicates that the convergence rate improved. However, the obtained solution is biased. On the other hand, our method improves the convergence rate without biasing the solution.

4.4. Relation to other continuous dynamics for optimization

Studying optimization algorithms through continuous dynamics has become an important approach. For example, Scieur et al. (2017) recently described the relation between the gradient flow and several discretization methods with a variety of optimization methods, including accelerated optimization methods such as the Nesterov method. Since our proposed dynamics is a perturbed gradient flow, we can combine more sophisticated higher-order discretization methods to ours following by Scieur et al. (2017). We note that the continuous dynamics of Nesterov's scheme is known as a second-order differential equation (Wibisono et al., 2016), while our continuous dynamics are first-order differential equations. Future work might introduce perturbation to that second-order equation.

5. NUMERICAL EXPERIMENTS

We confirmed our theoretical findings through numerical experiments. First, we confirmed the acceleration of continuous dynamics. Then, we observed the convergence behavior of two different discretization methods: the Euler and Euler-leapfrog (ELF) methods. We also show additional numerical experiments in Appendix I.

5.1. Least square experiments

We considered $F(x) = \frac{1}{N}\sum_{i=1}^{N}(A_{i*}x - y_i)^2$ where $y = (y_1, \dots y_N)^\top$ and A_{i*} denotes the ith row of $A \in \mathbb{R}^{N\times d}$. We generated design matrix A with entries following $\mathcal{N}(0,1)$. y was generated by $y = Az + \epsilon$ where $z \sim \mathcal{N}(0, I_{d\times d})$ and $\epsilon \sim \mathcal{N}(0, I_{N\times N})$. Then, Hessian matrix is $H = A^\top A$. Since the properties of J depend on whether d is odd or even, we considered $(d, N) = (400, 600)$ and $(401, 600)$. For $d = 401$, we introduced a dummy variable and solved the problem with $d = 402$. Detailed experimental settings and further discussions are presented in Appendix H.

First, we compared the continuous dynamics of the gradient flow (GF) and the perturbed dynamics under three types of J. One is a completely random matrix of which each entry follows the standard Gaussian (Random-J); the second is obtained by Algorithm 1 (Alg-J); the third is the optimal matrix obtained by the method in Lelièvre et al. (2013) (Opt-J) (its algorithm is shown in Appendix H). The results are shown in Fig. 1. For the perturbed dynamics, the results are the averages of ten repetitions for different realizations of random perturbation. Table 1 shows how the largest and the smallest real part of the eigenvalues of the Hessian matrix are changed by the perturbation. As shown in Proposition 4, the larger the smallest real part of the eigenvalue is, the faster convergence we have.

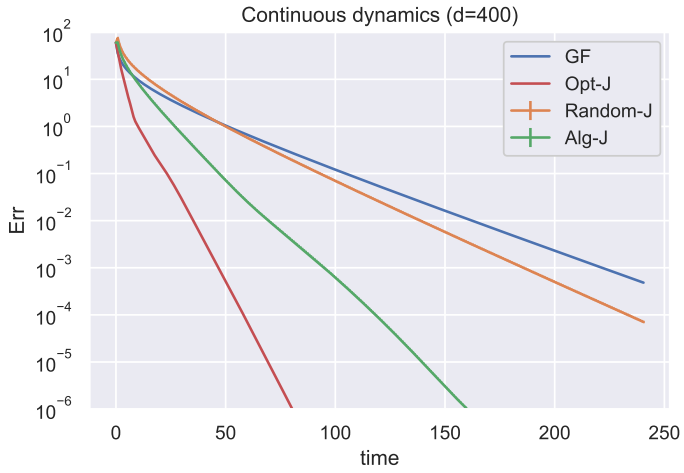


Figure 1: Convergence behavior of continuous dynamics

Table 1: Smallest and largest real parts of eigenvalues of Hessian matrix $A^\top A$

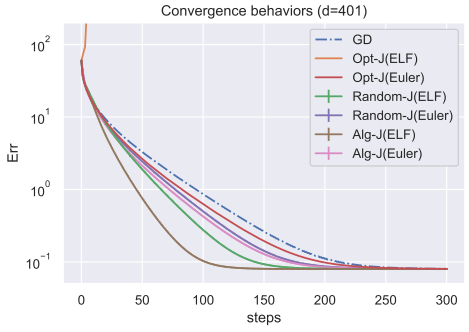
	$\mathrm{Re}\lambda_1$	$\mathrm{Re}\lambda_d$
GF	0.03	3.26
Random-J	0.05 $\pm$0	2.64 $\pm$ 0.003
Alg1-J	0.10 $\pm$0	2.52 $\pm$ 0
Opt-J	0.20	1.74

The optimal choice of J shows the best performance. We can confirm that completely random J still remains useful for acceleration. We also confirmed that for each different J, all the eigenvalues of the perturbed Hessian matrix are distinct, meaning that the perturbed Hessian matrix is diagonalizable. We also show the histogram of the sigular values for each J in Appendix 7.

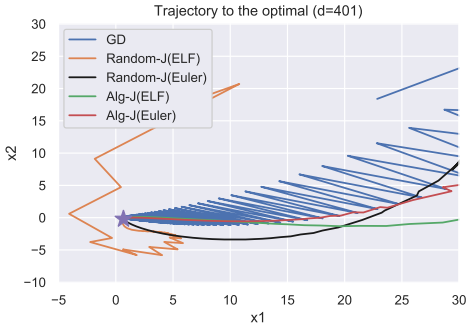
Next, we compared the discretization methods and different choices of J. The choice of J is identical as the continuous settings. We used optimal step sizes. For the ELF, we tuned α following Algorithm 1. For the Euler method, since it was sensitive to the choice of α, we reported the best result among those obtained with several different αs. The results are shown in Fig. 2. As shown in Propositions 5 and 6, although the ELF method shows faster convergence than the GD, the Euler discretization does not. We also found that the optimal choice of J by Lelièvre et al. (2013) is unstable with the ELF method. This is because its singular values are significantly large, and it does not satisfy the conditions of the ELF method. Figs. 2(b) and 2(d) show the trajectories of the GD and the proposed perturbed dynamics. Those of the perturbed dynamics are smoother. This figure suggests that the proposed method achieved rapid convergence.

5.2. Logistic regression experiments

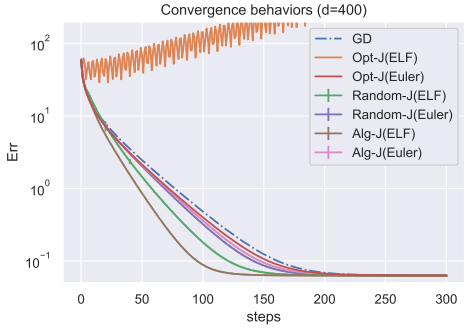
We considered learning parameters of logistic regression for binary classification. Let the input and output pairs of data $\{(z_i, y_i)\}_{i=1}^N$, where $z_i \in \mathbb{R}^d$ and $y_i \in \{1, -1\}$. Let $\tilde{z}_i := (z_i, 1)^\top \in \mathbb{R}^{d+1}$. The objective function is given as $F(x) = \frac{1}{N}\sum_{i=1}^N \ln \sigma(y_i x^\top \tilde{z}_i)$, where $\sigma(x) = (1+\exp(-x))^{-1}$ is the logistic function and $x \in \mathbb{R}^{d+1}$ is the parameter that we optimized. We compared the convergence



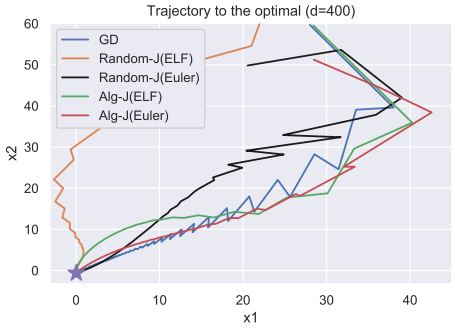
(a) Convergence of $\frac{\|x_k - x^*\|}{\|x^*\|}$



(b) Trajectory to x^*, indicated by star



(c) Convergence of $\frac{\|x_k - x^*\|}{\|x^*\|}$



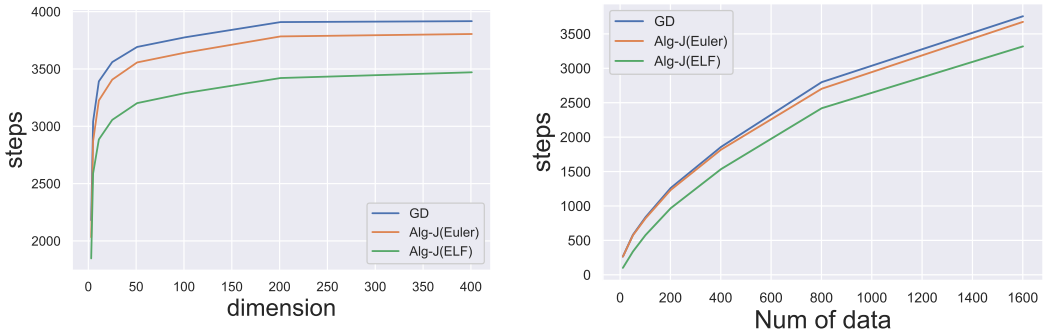
(d) Trajectory to x^*, indicated by star

Figure 2: Comparisons of different discretization. $d = 401$ for (a) and (b). $d = 400$ for (c) and (d)

speed of the GD and our proposed algorithm with skew-symmetric matrices generated by Algorithm 1 (Alg-J) and used the discretizations of the ELF method and the Euler method. Note that in logistic regression, using optimal skew-symmetric matrices is computationally demanding since Hessian matrices depend on the current position of x_k. This means the optimal J changes during the optimization, and thus we need to calculate the optimal J at each step. We also found that using the completely random skew-symmetric matrices with each entry following the standard Gaussian does not accelerate the convergence.

First, we considered toy data experiments to observe the convergence behavior of the GD and the Euler and Euler-leapfrog (ELF) methods of our proposed methods. To generate toy data, we drew each dimension of z from the uniform distribution between -1 and 1 and generated each dimension of the true parameter x from the uniform distribution between -5 and 5. The result is shown in Fig. 3. In Fig. 3(a), we fixed $N = 5000$, changed d, and measured the number of steps required for convergence. In Fig. 3(b), we fixed $d = 501$ and changed N. In both experiments, we confirmed that our proposed algorithm consistently accelerated the convergence in both large sample and large dimension settings.

Next, we used a real dataset to confirm that our proposed algorithm is useful in practice. We used four datasets in the UCI machine learning repository (Dheeru and Karra Taniskidou, 2017), and the result is shown in Fig. 4. Our proposed algorithm using the ELF method consistently accelerated the convergence. We found that using the Euler discretization did not always accelerate the convergence, which is consistent with our Proposition 5.



(a) Change d and fix $N = 5000$ (b) Change N and fix $d = 501$

Figure 3: Convergence behaviors of logistic regression under different d and N

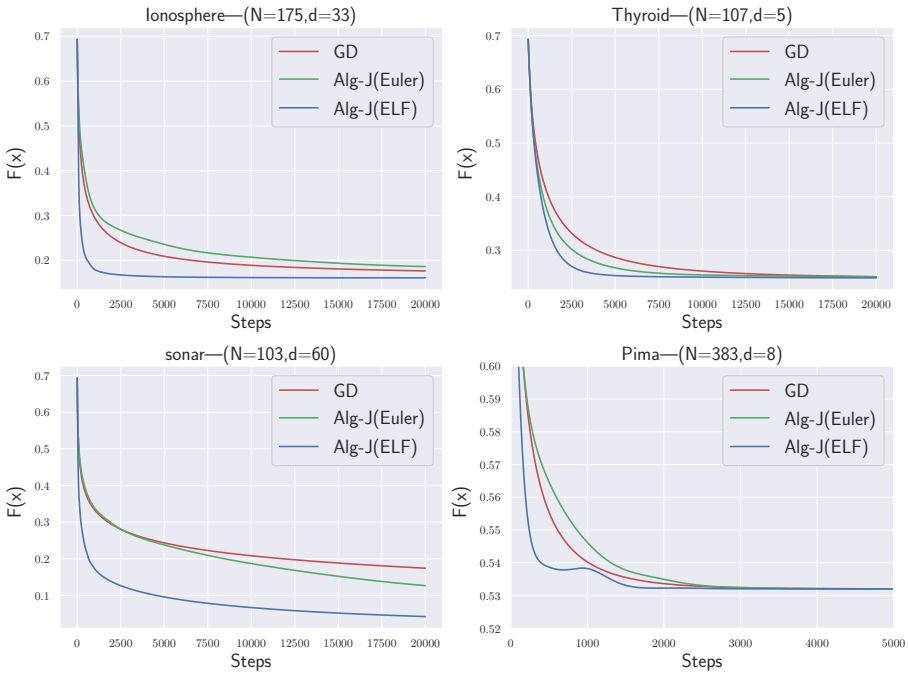


Figure 4: Convergence behavior of logistic regression: N is amount of data points and d is input dimensions.

6. CONCLUSION

We proposed a new continuous dynamics, which was obtained by perturbing the gradient flow by a random skew-symmetric matrix. By analyzing the perturbed Hessian matrix, we proved that perturbed dynamics shows rapid convergence. We presented a new discretization method that combines the Euler and leapfrog methods. It preserved the faster convergence property better than the gradient descent. We also presented an effective algorithm to select hyper-parameters.

An important conclusion of our work is that the perturbation technique in sampling is also useful for optimization. Our result suggests that perturbing the underlying dynamics is different from the standard scheme of minimizing a functional, it is a promising approach for designing optimization algorithms.

Our work can be extended in various ways. In this paper, we focused on the perturbation of a skew-symmetric matrix although there are other types of perturbations in sampling such as the one proposed by Maragliano and Vanden-Eijnden (2006). Incorporating such techniques into optimization would be an interesting research direction. Combining our technique with Nesterov's second-order dynamics scheme is also promising. In sampling, applying our discretization technique to existing Langevin-based sampling may provide potential improvements.

Acknowledgments

FF was supported by JST ACT-X Grant Number JPMJAX190R. IY acknowledges the support of the ANR as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

References

Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Andrew B. Duncan, Tony Lelièvre, and Grigorios. A. Pavliotis. Variance reduction using nonreversible langevin samplers. *Journal of Statistical Physics*, 163(3):457–491, May 2016.

Andrew B. Duncan, Nikolas. Nüsken, and Grigorios. A. Pavliotis. Using perturbed underdamped langevin dynamics to efficiently sample from probability distributions. *Journal of Statistical Physics*, 169(6):1098–1131, Dec 2017a.

Andrew B. Duncan, Grigorios. A. Pavliotis, and Konstantinos. C. Zygalakis. Nonreversible langevin samplers: Splitting schemes, analysis and implementation. *arXiv preprint arXiv:1701.04247*, 2017b.

Alain Durmus and Szymon Majewski. Analysis of langevin monte carlo via convex optimization. *Journal of Machine Learning Research*, 20(73):1–46, 2019.

Futoshi Futami, Issei Sato, and Masashi Sugiyama. Accelerating the diffusion-based ensemble sampling by non-reversible dynamics. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3337–3347. PMLR, 13–18 Jul 2020.

Futoshi Futami, Tomoharu Iwata, Naonori Ueda, and Issei Sato. Accelerated diffusion-based sampling by the non-reversible dynamics with skew-symmetric matrices. *Entropy*, 23(8), 2021. ISSN 1099-4300. doi: 10.3390/e23080993.

Gene H Golub and Charles F. Van Loan. *Matrix computations*, volume 3. JHU press, 2012.

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

Chii-Ruey Hwang, Raoul Normand, and Sheng-Jhih Wu. Variance reduction for diffusions. *Stochastic Processes and their Applications*, 125(9):3522–3540, 2015.

Marcus Kaiser, Robert L. Jack, and Johannes Zimmer. Acceleration of convergence to equilibrium in markov chains by breaking detailed balance. *Journal of Statistical Physics*, 168(2):259–287, Jul 2017.

Tony Lelièvre, 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.

Luca Maragliano and Eric Vanden-Eijnden. A temperature accelerated method for sampling free energy and determining reaction pathways in rare events simulations. *Chemical physics letters*, 426(1-3):168–175, 2006.

Michael Muehlebach and Michael Jordan. A dynamical systems perspective on nesterov acceleration. In *International Conference on Machine Learning*, pages 4656–4662, 2019.

Masashi Okamoto. Distinctness of the eigenvalues of a quadratic form in a multivariate sample. *Ann. Statist.*, 1(4):763–765, 07 1973. doi: 10.1214/aos/1176342472.

Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

Damien Scieur, Vincent Roulet, Francis Bach, and Alexandre d'Aspremont. Integration methods and optimization algorithms. In *Advances in Neural Information Processing Systems*, pages 1109–1118, 2017.

Bin Shi, Simon S Du, Weijie Su, and Michael I Jordan. Acceleration via symplectic discretization of high-resolution differential equations. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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*, pages 2510–2518, 2014.

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

Jingzhao Zhang, Aryan Mokhtari, Suvrit Sra, and Ali Jadbabaie. Direct runge-kutta discretization achieves acceleration. In *Advances in Neural Information Processing Systems*, volume 31, 2018.