

ADAPTIVE PRECONDITIONED GRADIENT DESCENT WITH ENERGY*

HAILIANG LIU[†], LEVON NURBEKYAN[‡], XUPING TIAN[§], AND YUNAN YANG[¶]

Abstract. We propose an adaptive step size with an energy approach for a suitable class of preconditioned gradient descent methods. We focus on settings where the preconditioning is applied to address the constraints in optimization problems, such as the Hessian-Riemannian and natural gradient descent methods. More specifically, we incorporate these preconditioned gradient descent algorithms in the recently introduced Adaptive Energy Gradient Descent (AEGD) framework. In particular, we discuss theoretical results on the unconditional energy-stability and convergence rates across three classes of objective functions. Furthermore, our numerical results demonstrate excellent performance of the proposed method in several challenging test optimization problems.

Keywords. Adaptive step size; preconditioning matrix; natural gradient; constrained optimization; energy stability; convergence rates.

AMS subject classifications. 65K10; 90C26.

1. Introduction

We present an optimization method involving adaptive step size with energy coupled with preconditioned gradient descent, designed for solving the constrained optimization problem:

$$\min_{\theta \in \Theta} L(\theta). \quad (1.1)$$

Here $L \in C^1(\mathbb{R}^n)$ is bounded below, $\Theta \subset \mathbb{R}^n$ represents the set of all possible parameters. In scenarios where $\Theta = \mathbb{R}^n$, one common strategy for solving (1.1) is to apply simple iterative algorithms such as the standard gradient descent (GD):

$$\theta_{k+1} = \theta_k - \eta \nabla L(\theta_k), \quad (1.2)$$

where $\eta > 0$ is the step size. However, if Θ represents a constraint set where the simple GD cannot guarantee the iterates to stay in Θ , the projected gradient descent method (PGD) is often used:

$$\theta_{k+1} = \mathcal{P}_\Theta(\theta_k - \eta \nabla L(\theta_k)), \quad (1.3)$$

where \mathcal{P}_Θ denotes the projection onto Θ . Under relatively mild assumptions on L and Θ , it can be proven that using (projected) GD with a sufficiently small step size η enhances the quality of the iterates over time, $L(\theta_{k+1}) < L(\theta_k)$, and the method converges to a stationary point within Θ [34]. A more refined version of the algorithm determines η by searching for the minimum of the objective function along the descent direction [30] at each iteration.

Closely related classes of algorithms are preconditioned gradient descent methods defined as

$$\theta_{k+1} = \theta_k - \eta T_k \nabla L(\theta_k), \quad (1.4)$$

*Received: May 25, 2024; Accepted (in revised form): December 19, 2024. Communicated by Lexing Ying.

[†]Department of Mathematics, Iowa State University, Ames, IA 50011, USA (hliu@iastate.edu).

[‡]Department of Mathematics, Emory University, Atlanta, GA 30322, USA (lmurbek@emory.edu).

[§]Department of Mathematics, Iowa State University, Ames, IA 50011, USA (xupingt@iastate.edu).

[¶]Department of Mathematics, Cornell University, Ithaca, NY 14853, USA (yunan.yang@cornell.edu).

where T_k is a suitable preconditioning matrix. Algorithms of the form (1.4) include many classical and well-known optimization methods, such as Newton's and Gauss–Newton methods [30], Hessian–Riemannian gradient descent (HRGD) [2], and natural gradient descent (NGD) [3], just to name a few. In this work, we are interested in the last two in the context of constrained optimization problems. More specifically, we think of HRGD as a method for enforcing *convex constraints* and NGD as a method of incorporating the geometry of the *constraints manifold* when the latter admits an *explicit parametrization*.

HRGD corresponds to the choice $T_k = P(\theta_k) (\nabla^2 h(\theta_k))^{-1}$, where h is a suitable convex function, and $P(\theta_k)$ is the $\nabla^2 h(\theta_k)$ -orthogonal projection onto an appropriate linear subspace. Furthermore, NGD corresponds to the choice $T_k = G(\theta_k)^{-1}$, where $G(\theta)$ is a suitable positive semidefinite matrix [31].

Although our focus in this paper is on HRGD and NGD, our proposed approach is versatile and can be extended to problems beyond HRGD and NGD, provided a well-identified preconditioning matrix is available. For instance, it can be applied to the Laplacian Smoothing Gradient Descent (LSGD) method [32], where the authors use $A = I - \sigma \Delta$ to reduce the variance of the SGD method, with I being the $n \times n$ identity matrix and Δ the discrete one-dimensional Laplacian.

Adaptive Energy Gradient Descent (AEGD) [21, 22] is a recently introduced method for enhancing the performance of gradient descent algorithms for unconstrained problems. More specifically, let $c \in \mathbb{R}$ be such that $\inf_{\theta \in \mathbb{R}^n} (L(\theta) + c) > 0$, and $l(\theta) = \sqrt{L(\theta) + c}$.

The algorithm initiates from θ_0 and $r_0 = l(\theta_0)$ and iterates as follows:

$$v_k = \nabla l(\theta_k), \quad (1.5a)$$

$$r_{k+1} = \frac{r_k}{1 + 2\eta \|v_k\|^2}, \quad (1.5b)$$

$$\theta_{k+1} = \theta_k - 2\eta r_{k+1} v_k. \quad (1.5c)$$

Here, $\{r_k^2\}_{k=0}^\infty$ play the role of an energy. In sharp contrast to GD, energy-adaptive gradient methods, like AEGD, exhibit unconditional energy stability; that is, $\{r_k^2\}_{k=0}^\infty$ is a decreasing sequence irrespective of the base step size $\eta > 0$. The excellent performance of AEGD-type schemes has been demonstrated across various optimization tasks [21, 22]. AEGD can be extended to accommodate stochastic effects in gradient estimation [22] or incorporate momentum for further convergence acceleration [21]. Moreover, the results in [22] indicate that c has minimal impact on the performance of (1.5).

The main objective of this paper is to extend the AEGD framework to include preconditioned gradient descent algorithms of the form (1.4). Indeed, setting c and l as above, we consider the following algorithm:

$$v_k = T_k \nabla l(\theta_k), \quad (1.6a)$$

$$r_{k+1} = \frac{r_k}{1 + 2\eta \|v_k\|^2}, \quad (1.6b)$$

$$\theta_{k+1} = \theta_k - 2\eta r_{k+1} v_k, \quad (1.6c)$$

which we call the adaptive preconditioned gradient descent with energy (AEPG).

For HRGD, we consider only affine constraints Θ , whereas for NGD, we consider only $\Theta = \mathbb{R}^n$. In both cases, the updates in (1.6) guarantee that $\theta_k \in \Theta$ holds for all $k \in \mathbb{N}$ as long as $\theta_0 \in \Theta$, with any fixed η . Under these scenarios, the proposed method (1.6) demonstrates unconditional energy stability, regardless of the magnitude of η . This raises the following question:

What is the convergence rate of this preconditioned gradient descent method with an adaptive time step incorporating energy?

We provide answers for two scenarios:

- (1) When $\Theta = \mathbb{R}^n$, and $T_k = A_k^{-1}$ for some symmetric and positive definite A_k , we prove that the AEPG method converges to a minimum of the objective function with an appropriate step size. The convergence rates are influenced by the geometric properties of the objective function, and we provide rates for both convex and non-convex objectives, including those that satisfy the Polyak–Lojasiewics (PL) condition. The analysis in this case covers HRGD without affine constraints, $P_k = I$ and $A_k = \nabla^2 h(\theta_k)$, and NNGD, $A_k = G(\theta_k)$.
- (2) When $\Theta = \{\theta \in \mathbb{R}^n, B\theta = b\}$, and $T_k = P_k G_k^{-1}$, where G_k is symmetric and positive definite, and P_k is the G_k -orthogonal projection onto $\ker(B)$, we prove that the AEPG method converges to a minimum of the objective function with a suitably small step size. We derive convergence rates for both convex and non-convex objectives, including those that satisfy a projected Polyak–Lojasiewics (PL) condition defined in (2.19). The analysis in this case covers the HRGD with affine constraints, $G_k = \nabla^2 h(\theta_k)$ and $P_k = P(\theta_k)$.

A specific example of interest is the Wasserstein Natural Gradient Descent (WNGD), which has recently gained much attention in machine learning and PDE-based optimization applications [4, 17, 18, 31, 37, 42]. Previous works primarily focus on computing descent directions, whereas our emphasis here is on time steps. We apply the AEPG algorithm in Equation (1.6) to WNGD, incorporating the efficient computation of $\{v_k\}$ using methods developed in [31]. See Section 5 for more details on WNGD.

Our approach represents a unique fusion of preconditioned gradient and energy-adaptive strategies. This combination enables acceleration of standard GD in terms of both better descent directions and step sizes, with provable convergence rates under relatively mild conditions on both the objective function and the step size. Our current findings for constrained optimization problems provide insight into energy-driven update rules with preconditioned gradients, creating a path toward additional algorithms employing AEGD-type techniques.

1.1. Contributions. In summary, our main contributions can be outlined as follows.

- (1) **Novel AEPG Algorithm.** We introduce a novel Adaptive Energy-based Preconditioned Gradient (AEPG) algorithm for the optimization problem represented by (1.1). The convergence theory is established across three distinct cases: general differentiable objective functions, nonconvex objective functions satisfying the Polyak–Lojasiewics (PL) condition, and convex objective functions.
- (2) **Application to HRGD and NNGD.** We apply our proposed AEPG to Hessian–Riemannian Gradient Descent (HRGD) and Natural Gradient Descent (NNGD), two specific optimization algorithms that fit the general framework of Equation (1.4). Our approach improves these methods by providing unconditional energy stability.
- (3) **Unifying Framework.** We demonstrate that HRGD can be interpreted as an instance of NNGD. To the best of our knowledge, this is a new connection, providing a unified framework for studying various constrained optimization algorithms.
- (4) **Numerical Experiments.** We conduct several numerical experiments, illustrating that AEPG exhibits faster convergence compared to classical algorithms without using an adaptive step size. This superiority is powerful in handling challenging scenarios such as ill-conditioned or nonconvex problems.

1.2. Review of Related Literature. It is natural to seek an improved descent direction to accelerate the convergence speed of GD. This can be achieved by leveraging curvature information from the objective function, as demonstrated in Newton and quasi-Newton methods. Specifically, for α -smooth objective functions, these methods attain a quadratic and superlinear local convergence rate, respectively [30]. The natural gradient method, initially introduced in [3], falls within this category, using curvature information from the Riemannian metric tensor. In recent years, the Wasserstein natural gradient has attracted significant attention [18, 19, 31]. We refer interested readers to [26] for additional insights on the natural gradient. Adding momentum is another widely employed technique to enhance descent directions. Examples include the heavy-ball (HB) method [33] and Nesterov's accelerated gradient (NAG) [27, 28].

An alternative avenue to expedite GD (or SGD) convergence involves the use of adaptive step sizes. Previous steps' gradients are often used to adjust the step size. This idea is used in renowned works such as AdaGrad [12] and RMSProp [38]. Adam [15], combining momentum and adaptive step size benefits, has demonstrated rapid convergence in early iterations and has gained widespread use in the machine learning community. Further enhancements to Adam can be explored in [11, 23, 24, 35, 44]. These adaptive methods not only update the step size in each iteration but also compute individual step sizes for each parameter. Other adaptive techniques adjust the step size based on the error of the objective function value [3, 41].

Manifold optimization is a well-explored area in the literature. The general feasible algorithm framework on the manifold involves the use of a retraction operator, with parallel translation or vector transport aiding in pulling the update back to the manifold constraint. Computational costs and convergence behaviors of different retraction operators vary significantly; interested readers can find further details in [1, 7, 13, 14, 20] and related references.

Finally, a considerable body of work exists on strategies for improving GD in unconstrained optimization problems. For line search-based GD methods, the classical Armijo rule (1966) and the Wolfe conditions (1969) guide the inexact line search, with additional insights available in [29, 43], the book of [30], and associated references. The BB method [5], a gradient method with modified step sizes motivated by Newton's method but without involving any Hessian, is noteworthy. Numerous studies explore methods capable of handling constraints, such as projected gradient descent, proximal gradient methods, penalty methods, barrier methods, sequential quadratic programming (SQP) methods, interior-point methods, augmented Lagrangian/multiplier methods, primal-dual strategies, and active set strategies; see, for instance, [6, 30]. Therefore, our work contributes to the broader field of continuous optimization with constraints.

1.3. Notation. Throughout this paper, we adopt the following notations: $\|\cdot\|$ denotes the ℓ^2 norm for vectors and the spectral norm for matrices. Additionally, $\lambda_1(\cdot), \dots, \lambda_n(\cdot)$ represent the smallest and largest eigenvalues, respectively. For a function L , we use ∇L and $\nabla^2 L$ to denote its gradient and Hessian, respectively, while L^* denotes the global minimum value of L . For a positive definite matrix A , we introduce the vector-induced norm by A as $\|w\|_A := \sqrt{\langle w, Aw \rangle}$. The set $\{1, 2, \dots, n\}$ is denoted as $[n]$.

1.4. Organization. The rest of this paper is organized as follows. In Section 2, we introduce the AEPG method and present the key theoretical results, including unconditional energy stability, convergence, and convergence rates. Section 3 delves into HRGD, demonstrating how AEPG can solve a specific type of constrained optimization problem using a Hessian-Riemannian metric. Moving to Section 4, we show that the HRGD can be cast as an NGD. In particular, Section 5 discusses the Wasserstein

natural gradient and an efficient numerical method for computation through AEPG. Numerical results and examples are given in Section 6 to further elucidate the discussed concepts. Conclusions and further discussions follow in Section 7. Technical proofs for theoretical results in Section 2 are presented in Appendix A. A construction of the preconditioned matrix for HRDG is presented in Appendix B. Appendix C includes an illustrative example to explain the projected PL condition. Finally, we put parameter setup of the numerical examples in Appendix D.

2. Preconditioned AEGD and theoretical results

In this section, we present the theoretical results for the main preconditioned AEGD algorithm on stability, convergence, and convergence rates.

2.1. Assumptions. We assume that the objective function L is differentiable and bounded from below, and the update rule for AEPG follows (1.6). Throughout the paper, we assume that $\theta_0 \in \Theta$ yields

$$\theta_k \in \Theta, \quad \forall k \in \mathbb{N} \quad \text{and} \quad \eta > 0. \quad (2.1)$$

This assumption is valid in two specific cases we address:

(1) When $\Theta = \mathbb{R}^n$ and $T_k = A_k^{-1}$, where the matrices $(A_k)_{k=0}^{\infty}$ are symmetric and uniformly positive definite; that is,

$$\lambda_1 \|\xi\|^2 \leq \xi^\top A_k \xi \leq \lambda_n \|\xi\|^2, \quad \forall \xi \in \mathbb{R}^n, \quad k \geq 0, \quad (2.2)$$

for some $\lambda_n \geq \lambda_1 > 0$.

(2) When $\Theta = \{\theta \in \mathbb{R}^n, B\theta = b\}$ and $T_k = P_k G_k^{-1}$, where $B \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $(G_k)_{k=0}^{\infty}$ are symmetric and uniformly positive definite, and P_k are the G_k -orthogonal projection operators onto $\ker(B)$. The explicit form of the projection matrix is

$$P_k = I - G_k^{-1} B^\top (B G_k^{-1} B^\top)^{-1} B. \quad (2.3)$$

In Section 2.2 we discuss the unconstrained case, whereas Section 2.3 is devoted to the affine constraints case.

2.2. No equality constraints. In this section, we assume that $\Theta = \mathbb{R}^n$, and so (2.1) is valid.

THEOREM 2.1 (Unconditional energy stability). *AEPG (1.6) is unconditionally energy stable. Specifically, for any step size $\eta > 0$,*

$$r_{k+1}^2 = r_k^2 - (r_{k+1} - r_k)^2 - \frac{1}{\eta} \|\theta_{k+1} - \theta_k\|^2. \quad (2.4)$$

This implies that r_k is strictly decreasing and converges to r^ as $k \rightarrow \infty$. Furthermore,*

$$\sum_{k=0}^{\infty} \|\theta_{k+1} - \theta_k\|^2 \leq \eta r_0^2. \quad \Rightarrow \quad \lim_{k \rightarrow \infty} \|\theta_{k+1} - \theta_k\| = 0. \quad (2.5)$$

Proof. Starting from (1.6), we deduce

$$2r_{k+1}(r_{k+1} - r_k) = 2r_{k+1}v_k^\top(\theta_{k+1} - \theta_k) = -\frac{1}{\eta} \|\theta_{k+1} - \theta_k\|^2.$$

By rewriting with $2b(b-a) = b^2 - a^2 + (b-a)^2$, we obtain equality (2.4). This implies that r_k^2 is monotonically decreasing (also bounded below), ensuring convergence. The

non-negativity of r_k further ensures the convergence of r_k . Summation of (2.4) over k from $0, 1, \dots$ yields (2.5). \square

Note that, starting from (1.6b), the following relation is derived:

$$r_0 - r^* = 2\eta \sum_{j=0}^{\infty} r_{j+1} \|v_j\|^2 \Rightarrow \lim_{k \rightarrow \infty} (r_{k+1} \|v_k\|^2) = 0.$$

In simpler terms, after a finite number of steps, either $\|v_k\|^2$ becomes small or r_k becomes small. For convergence to a stationary point in general objectives ($\|v_k\|^2 \rightarrow 0$), it is essential to control the rate of decrease of r_k . This control can be achieved by imposing a moderate upper bound on the base step size η .

In Lemma 2.1, we establish a sufficient condition on η that ensures a positive lower bound for $(r_k)_{k=0}^{\infty}$, a condition crucial in the convergence proof of Theorem 2.2. This is where the α -smoothness assumption comes into play. We define L as α -smooth if $\|\nabla^2 L(\theta)\| \leq \alpha$ for any $\theta \in \mathbb{R}^n$. Here, $L^* := \inf_{\theta} L$, and we assume c is chosen such that $l^* := \sqrt{L^* + c} > 0$.

LEMMA 2.1. *Assume L is α -smooth, bounded from below by L^* , and r_k generated by AEPG (1.6) with $T_k = A_k^{-1}$. If $r_0 \geq \frac{l(\theta_0) - l^*}{\lambda_1}$, then*

$$r_k \geq r^* > 0, \quad (2.6)$$

provided that:

$$\eta < \eta_s := \frac{4l^* \lambda_1}{\alpha r_0^2} \left(r_0 - \frac{l(\theta_0) - l^*}{\lambda_1} \right) \quad (2.7)$$

where $\lambda_1 > 0$ is the smallest eigenvalue of A_k .

Proof. Note that $l(\theta) = \sqrt{L(\theta) + c}$ is concave with respect to L , and $\frac{dl}{dL} = \frac{1}{2l}$. Thus, we have

$$l(\theta_{j+1}) - l(\theta_j) \leq \frac{1}{2l(\theta_j)} (L(\theta_{j+1}) - L(\theta_j)). \quad (2.8)$$

Using the α -smoothness of L , we have

$$L(\theta_{j+1}) \leq L(\theta_j) + \nabla L(\theta_j)^\top (\theta_{j+1} - \theta_j) + \frac{\alpha}{2} \|\theta_{j+1} - \theta_j\|^2. \quad (2.9)$$

From (1.6a), we derive $\nabla L(\theta_j) = 2l(\theta_j) \nabla l(\theta_j) = 2l(\theta_j) A_j v_j$. Hence,

$$\begin{aligned} \nabla L(\theta_j)^\top (\theta_{j+1} - \theta_j) &= 2l(\theta_j) (A_j v_j)^\top (-2\eta r_{j+1} v_j) \\ &= -4\eta l(\theta_j) r_{j+1} v_j^\top A_j v_j \\ &\leq -4\eta l(\theta_j) \lambda_1 r_{j+1} \|v_j\|^2 \\ &= 2l(\theta_j) \lambda_1 (r_{j+1} - r_j), \end{aligned} \quad (2.10)$$

where the last equality follows from a formulation of (1.6b). Substituting (2.10) and (2.9) into (2.8), we obtain

$$l(\theta_{j+1}) - l(\theta_j) \leq \lambda_1 (r_{j+1} - r_j) + \frac{\alpha}{4l(\theta_j)} \|\theta_{j+1} - \theta_j\|^2.$$

Taking a summation over j from 0 to $k-1$, we have:

$$\begin{aligned} l(\theta_k) - l(\theta_0) &\leq \lambda_1(r_k - r_0) + \frac{\alpha}{4l^*} \sum_{j=0}^{k-1} \|\theta_{j+1} - \theta_j\|^2 \\ &\leq \lambda_1(r_k - r_0) + \frac{\alpha\eta}{4l^*} r_0^2. \end{aligned}$$

Considering $l(\theta_k) \geq l^*$ and the last inequality, we obtain

$$r_k \geq r_0 - \frac{l(\theta_0) - l^*}{\lambda_1} - \frac{\alpha\eta}{4l^*\lambda_1} r_0^2 = \frac{\alpha r_0^2}{4l^*\lambda_1} (\eta_s - \eta) > 0.$$

This establishes a uniform lower bound for r_k . Letting $k \rightarrow \infty$, we obtain (2.6). \square

REMARK 2.1. Let us discuss the selection of $r_0 = l(\theta_0)$, the default choice for consistency with the method derivation. When $\lambda_1 = 1$, then

$$\eta_s = \frac{4}{\alpha} \frac{L^* + c}{L(\theta_0) + c} \sim \frac{4}{\alpha}$$

for c large enough. This asymptotic limit clearly exceeds the upper threshold $2/\alpha$ which is known to be necessary for ensuring GD's stability.

REMARK 2.2. The stipulated lower bound for r_0 is not overly restrictive. It can be expressed equivalently as

$$\sqrt{L(\theta_0) + c} > \frac{\sqrt{L(\theta_0) + c} - l^*}{\lambda_1}.$$

This inequality holds for any value of c satisfying $c + L^* > 0$ when $\lambda_1 \geq 1$. However, for $\lambda_1 < 1$, a larger c becomes necessary to fulfill the condition:

$$c + L^* > \frac{(1 - \lambda_1)^2}{\lambda_1(2 - \lambda_1)} (L(\theta_0) - L^*).$$

THEOREM 2.2. *Under the same assumptions as in Lemma 2.1, with $r_0 \geq \frac{l(\theta_0) - l^*}{\lambda_1}$, let $(\theta_k)_{k=0}^{\infty}$ be generated by AEPG (1.6). Then $\{L(\theta_k)\}_{k=0}^{\infty}$ converges monotonically to $L(\theta^*)$, where θ^* is a local minimizer or a saddle point, if*

$$0 < \eta \leq \min \left\{ \eta_0, \eta_s \right\}, \quad \eta_0 := \frac{\lambda_1 l^*}{\alpha r_0}. \quad (2.11)$$

Proof. If $\eta \leq \eta_0$, the effective step size falls into the regime where $L(\theta_k)$ is shown to be decreasing, indicating convergence. With $r_0 \geq \frac{l(\theta_0) - l^*}{\lambda_1}$ and $\eta \leq \eta_s$, Lemma 2.1 ensures that r_k is bounded below by a positive constant, allowing $\|\nabla L(\theta_k)\|^2$ to converge to zero as k tends to infinity. For the detailed proof, please refer to Appendix A.1. \square

REMARK 2.3. The Stable Manifold Theorem from dynamical systems theory, as applied in [16], implies that if L satisfies a strict saddle property (that is, $\nabla^2 L$ has at least one strictly negative eigenvalue at a saddle point), then the probability of θ^* being a saddle point is zero. In practice, introducing noise can help prevent getting stuck at a saddle point. Let's discuss the convergence rate of AEPG with additional geometrical insights into L , including properties such as convexity or the Polyak–Lojasiewicz (PL)

condition. For a differentiable function $L: \mathbb{R}^n \rightarrow \mathbb{R}$ with $\arg \min L \neq \emptyset$ (indicating the optimization problem has at least one global minimizer), we say L satisfies the PL condition if there exists $\mu > 0$ such that:

$$\frac{1}{2} \|\nabla L(\theta)\|^2 \geq \mu(L(\theta) - L(\theta^*)), \quad \forall \theta \in \mathbb{R}^n, \quad \theta^* \in \arg \min L. \quad (2.12)$$

This condition implies that $\nabla L(\theta) = 0$ implies $L(\theta) = L(\theta^*)$ and $\theta \in \arg \min L$. In other words, critical points are global minimizers.

It's important to note that strongly convex functions ($\lambda_1(\nabla^2 f) \geq \mu$) satisfy the PL condition (2.12). However, a function that satisfies the PL condition may not necessarily be convex. For instance, consider the function:

$$L(\theta) = \theta^2 + 3 \sin^2 \theta, \quad \theta \in \mathbb{R},$$

which is nonconvex but satisfies the PL condition with $\mu = \frac{1}{32}$ and $\min L = 0$.

THEOREM 2.3. *Assume $(\theta_k)_{k=0}^{\infty} \subset \Theta$ when $\Theta \neq \mathbb{R}^n$. The convergence rates of AEPG (1.6) are given in three distinct scenarios:*

(i) *For any $\eta > 0$ and $r_0 > 0$, we have*

$$\min_{j < k} \|\nabla L(\theta_j)\|^2 \leq \frac{2r_0 \lambda_n^2}{\eta r_k k} \left(\max_{j < k} L(\theta_j) + c \right), \quad (2.13)$$

with λ_n given in (2.2). Under the assumptions of Theorem 2.2 with η satisfying (2.11), we have $r_k > r^* > 0$, and the following convergence rates:

(ii) *If L is PL with a global minimizer θ^* , then $\{\theta_k\}$ satisfies (2.14), hence convergent:*

$$\sum_{k=0}^{\infty} \|\theta_{k+1} - \theta_k\| \leq \frac{4\lambda_n}{\sqrt{2\mu\lambda_1}} \sqrt{L(\theta_0) - L(\theta^*)}, \quad (2.14)$$

$$L(\theta_k) - L(\theta^*) \leq e^{-c_0 k r_k / \lambda_n} (L(\theta_0) - L(\theta^*)), \quad c_0 := \frac{\mu\eta}{l(\theta_0)}.$$

(iii) *If L is convex with a minimizer θ^* , then*

$$L(\theta_k) - L(\theta^*) \leq \frac{c_1 \lambda_n \|\theta_0 - \theta^*\|^2}{k r_k}, \quad c_1 = \frac{2l(\theta_0)}{\eta}. \quad (2.15)$$

Proof. The proof of (ii) and (iii) is somewhat standard and is therefore deferred to Appendix A.2. Here we present a proof for (i).

(i) Using the scheme (1.6b), we have

$$r_{j+1} - r_j = -2\eta r_{j+1} \|v_j\|^2.$$

Take a summation over j from 0 to $k-1$ gives

$$r_0 - r_k = 2\eta \sum_{j=0}^{k-1} r_{j+1} \|v_j\|^2 \geq 2\eta r_k \sum_{j=0}^{k-1} \|v_j\|^2 \geq \frac{2\eta r_k}{\lambda_n^2} \sum_{j=0}^{k-1} \|\nabla l(\theta_j)\|^2,$$

which, upon rearrangement, leads to

$$k \min_{j < k} \|\nabla l(\theta_j)\|^2 \leq \sum_{j=0}^{k-1} \|\nabla l(\theta_j)\|^2 \leq \frac{r_0 \lambda_n^2}{2\eta r_k}.$$

Using $4l^2\|\nabla l\|^2 = \|\nabla L\|^2$, and dividing both sides by k gives result (2.13). \square

REMARK 2.4.

- (1) Unlike GD, which uses a fixed step size constrained by the typically unknown smoothness constant α (thus making standard GD prone to non-convergence with larger step sizes), the gradient norm sequence's convergence rate to zero for AEPG in scenario (i) extends to general non-convex objective functions for any $\eta > 0$. Notably, these observations remain valid regardless of the presence of α . It is natural to ponder whether this convergence rate can be enhanced. Interestingly, the answer is negative, at least within the general class of functions under consideration here.
- (2) Regarding the upper bound in Equation (2.13), we can establish the following inequality:

$$L(\theta_k) \leq L(\theta_0) + \frac{\alpha\eta r_0^2}{2}. \quad (2.16)$$

This bound holds when L is α -smooth. Using the α -smoothness of L and the descent direction of the search, we can express this as:

$$L(\theta_{j+1}) \leq L(\theta_j) + \nabla L(\theta_j)^\top (\theta_{j+1} - \theta_j) + \frac{\alpha}{2} \|\theta_{j+1} - \theta_j\|^2 \leq L(\theta_j) + \frac{\alpha}{2} \|\theta_{j+1} - \theta_j\|^2.$$

By summing over j from 0 to $k-1$ and using Equation (2.5),

$$L(\theta_k) - L(\theta_0) \leq \frac{\alpha}{2} \sum_{j=0}^{k-1} \|\theta_{j+1} - \theta_j\|^2 \leq \frac{\alpha\eta}{2} r_0^2.$$

It is important to note that such a bound is unavailable for GD unless the step size is sufficiently small.

- (3) It is crucial to highlight that the aforementioned results remain valid for a variable η , as long as it adheres to the constraint specified in (2.11) and is not excessively small. The convergence theory with a variable η provides the flexibility to adjust η when necessary. For instance, the inclusion of the sequence $(\theta_k)_{k=0}^\infty \subset \Theta$ becomes essential when $\Theta \neq \mathbb{R}^n$. This need for adaptability is addressed in Algorithm 1, which we will introduce in Section 3.2, where η is selected based on a line search at every iteration.

2.3. Equality constraints. In this section, we assume that

$$\Theta = \{\theta \in \mathbb{R}^n \mid B\theta = b\},$$

for some $B \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and

$$T_k = P(\theta_k)G(\theta_k)^{-1}, \quad (2.17)$$

where $G(\theta)$ is a symmetric positive definite matrix, and $P(\theta) : \mathbb{R}^n \rightarrow \ker(B)$ is the $G(\theta)$ -orthogonal projection operator on $\ker(B)$.

Before discussing convergence analysis results, we provide a motivation for the choice (2.17). To this end, assume that \mathbb{R}^n is endowed with a Riemannian structure given by a metric tensor $G(\theta)$, $\theta \in \mathbb{R}^n$; that is, for all $\theta \in \mathbb{R}^n$ we have an inner product

$$\langle v_1, v_2 \rangle_{G(\theta)} = v_1^\top G(\theta) v_2, \quad v_1, v_2 \in T_\theta \mathbb{R}^n, \quad (2.18)$$

where $T_\theta \mathbb{R}^n \cong \mathbb{R}^n$ is the tangent space of \mathbb{R}^n at θ . Note that

$$T_\theta \Theta \cong \ker(B) = \{v \in \mathbb{R}^n \mid Bv = 0\}.$$

The choice (2.17) is motivated by the following lemma.

LEMMA 2.2. *For every smooth $f: \mathbb{R}^n \rightarrow \mathbb{R}$ one has that*

$$\arg \min_v \left\{ \frac{df(\theta)}{dt} \mid \dot{\theta} = v, v \in T_\theta \Theta, \|v\|_{G(\theta)} \leq 1 \right\} = -\frac{P(\theta)G(\theta)^{-1}\nabla f(\theta)}{\|P(\theta)G(\theta)^{-1}\nabla f(\theta)\|_{G(\theta)}}.$$

Hence, the steepest descent direction of f on the submanifold (Θ, G) is $-P(\theta)G(\theta)^{-1}\nabla f(\theta)$.

Proof. We have that

$$\frac{df(\theta)}{dt} = v^\top \nabla f(\theta) = v^\top G(\theta)G(\theta)^{-1}\nabla f(\theta) = \langle v, G(\theta)^{-1}\nabla f(\theta) \rangle_{G(\theta)} = \langle v, \nabla^G f(\theta) \rangle_{G(\theta)},$$

where we denote by $\nabla^G f(\theta) = G(\theta)^{-1}\nabla f(\theta)$ the metric gradient. So we have the problem

$$\begin{aligned} & \arg \min_v \left\{ \langle v, \nabla^G f(\theta) \rangle_{G(\theta)} \mid v \in T_\theta \Theta, \|v\|_{G(\theta)} \leq 1 \right\} \\ &= \arg \min_v \left\{ \langle v, P(\theta)\nabla^G f(\theta) \rangle_{G(\theta)} \mid v \in T_\theta \Theta, \|v\|_{G(\theta)} \leq 1 \right\} \\ &= -\frac{P(\theta)\nabla^G f(\theta)}{\|P(\theta)\nabla^G f(\theta)\|_{G(\theta)}} = -\frac{P(\theta)G(\theta)^{-1}\nabla f(\theta)}{\|P(\theta)G(\theta)^{-1}\nabla f(\theta)\|_{G(\theta)}}, \end{aligned}$$

where the first equation follows from the definition of the orthogonal projection. \square

REMARK 2.5. As previously mentioned, selecting (2.17) in (1.6) yields $\theta_{k+1} - \theta_k = -2\eta r_{k+1} v_k \in T_{\theta_k} \Theta \cong \ker(B)$. Therefore, (1.6) returns feasible updates; that is, (2.1) is valid.

The following theorem provides a concise summary of the stability and convergence properties of (1.6) in the context of the chosen (2.17).

THEOREM 2.4. *Let $\theta_0 \in \Theta = \{\theta \in \mathbb{R}^n, B\theta = b\}$, and $G(\theta)$ be symmetric and positive definite with $0 < \lambda_1 \leq \|G(\theta)\|_2 \leq \lambda_n$ for all $\theta \in \Theta$. Furthermore, let (θ_k) be generated by (1.6) with (2.17). The following statements hold:*

- (1) **Unconditional Energy Stability:** *It satisfies unconditional energy stability, as stated in Theorem 2.1.*
- (2) **Positive Lower Bound:** *The statement in Lemma 2.1 regarding a positive lower bound for r_k remains valid.*
- (3) **Monotonic Convergence:** *Under the same assumptions and conditions on η as described in Theorem 2.2, $L(\theta_k)_{k=0}^\infty$ decreases monotonically and converges with*

$$\lim_{k \rightarrow \infty} \|P_k^\top \nabla L(\theta_k)\| \rightarrow 0.$$

- (4) **Convergence Rates:** *Convergence rates are obtained in three distinct scenarios:*

(i) For any $\eta > 0$ and $r_0 > 0$, it holds that

$$\min_{j < k} \|P_j^\top \nabla L(\theta_j)\|^2 \leq \frac{2r_0 \lambda_n^2}{\eta r_k k} \left(\max_{j < k} L(\theta_j) + c \right).$$

Under the same assumptions and conditions on η as described in Theorem 2.2, where $r_k > r^* > 0$, the following convergence rates are established:

(ii) If L satisfies the projected PL condition with a minimum $\theta^* \in \Theta$:

$$\frac{1}{2} \|P(\theta)^\top \nabla L(\theta)\|^2 \geq \mu(L(\theta) - L(\theta^*)), \quad \forall \theta \in \Theta, \quad (2.19)$$

where $\mu > 0$, then the sequence $\{\theta_k\}$ has finite length, and hence converges. Furthermore,

$$L(\theta_k) - L(\theta^*) \leq e^{-c_0 k r_k / \lambda_n} (L(\theta_0) - L(\theta^*)), \quad c_0 := \frac{\mu \eta}{l(\theta_0)}. \quad (2.20)$$

(iii) If L is convex with a minimum $\theta^* \in \Theta$, then:

$$L(\theta_k) - L(\theta^*) \leq \frac{c_1 \lambda_n \|\theta_0 - \theta^*\|^2}{k r_k}, \quad c_1 = \frac{2l(\theta_0)}{\eta}. \quad (2.21)$$

Proof.

(1) The proof for Theorem 2.1 remains applicable in this context.

(2) To demonstrate that (2.10) still holds in the case of (2.17), we recall that P_j is the G_j -orthogonal projection operator, and P_j is an involution, $P_j^2 = P_j$, and G_j -symmetric, $G_j P_j = P_j^\top G_j$. Hence, $T_j = P_j G_j^{-1} = G_j^{-1} P_j^\top$ and $T_j = P_j T_j = P_j G_j^{-1} P_j^\top$. We can express the gradient term as:

$$\begin{aligned} \nabla L(\theta_j)^\top (\theta_{j+1} - \theta_j) &= -4\eta l(\theta_j) r_{j+1} \nabla l(\theta_j)^\top G_j^{-1} P_j^\top \nabla l(\theta_j) \\ &= -4\eta l(\theta_j) r_{j+1} \|P_j^\top \nabla l(\theta_j)\|_{G_j^{-1}}^2. \end{aligned} \quad (2.22)$$

Considering that

$$v_j = T_j \nabla l(\theta_j) = G_j^{-1} P_j^\top \nabla l(\theta_j) \Rightarrow P_j^\top \nabla l(\theta_j) = G_j v_j,$$

we further bound (2.22) by

$$\begin{aligned} \nabla L(\theta_j)^\top (\theta_{j+1} - \theta_j) &= -4\eta l(\theta_j) r_{j+1} \|G_j v_j\|_{G_j^{-1}}^2 \\ &\leq -4l(\theta_j) \eta \lambda_1 r_{j+1} \|v_j\|^2 = 2l(\theta_j) \lambda_1 (r_{j+1} - r_j). \end{aligned}$$

This establishes the validity of (2.10), and the remaining portion of the proof aligns with that of Lemma 2.1.

The proofs for (3) and (4) mirror those of Theorem 2.3, and therefore are deferred to Appendix A.3. \square

REMARK 2.6. Rather than relying on the conventional gradient $\nabla L(\theta)$, the convergence and convergence rate of $L(\theta_k)$ are influenced by the projected gradient $P^\top \nabla L(\theta)$. This is also evident at the continuous level: the projected PL condition (2.19) implies:

$$\frac{d}{dt} L(\theta(t)) = -\|P^\top \nabla L(\theta)\|_{G^{-1}}^2 \leq -\frac{2\mu}{\lambda_n} (L(\theta(t)) - L^*).$$

Thus, for any $t > 0$,

$$L(\theta(t)) - L^* \leq e^{-2\mu t/\lambda_n} (L(\theta(0)) - L^*),$$

where $\theta(0) = \theta_0$, representing the initial guess.

REMARK 2.7. In general, verifying the projected PL condition directly can be challenging. To illustrate this new structural condition, consider an example involving loss functions of the form:

$$L(\theta) = \frac{1}{2} (\beta\theta_1^2 + \alpha\theta_2^2),$$

subject to a general linear constraint

$$q(\theta) = a\theta_1 + b\theta_2 - 1 = 0,$$

where $ab \neq 0, \alpha \geq \beta > 0$ are constants. This constrained minimization problem is convex, thereby admitting a unique solution:

$$\theta^* = \left(\frac{a\alpha}{a^2\alpha + b^2\beta}, \frac{b\beta}{a^2\alpha + b^2\beta} \right).$$

Upon a careful calculation (refer to Appendix C for details), it can be verified that the projected PL condition holds for any $\theta \in \Theta$, where

$$\mu = \frac{a^2\alpha + b^2\beta}{a^2 + b^2}.$$

To apply the method and theoretical results to a specific optimization task, it remains essential to identify and compute matrices G_j^{-1} . This aspect will be addressed in the subsequent sections.

REMARK 2.8. It is observed that the convergence rate for convex objectives in (4)-(iii) remains the same as that in the scenario $T_k = A_k^{-1}$.

3. Hessian-Riemannian metric

In Section 2.3, we explored preconditioning matrices of the form (2.17), where G is a generic metric. In this section, we discuss a more specific choice for G based on the Hessian of a suitable convex function. The preconditioned gradient descent algorithm (1.4) based on such G is known as the Hessian-Riemannian gradient descent [2].

For analysis purposes, we previously assumed only affine equality constraints. In this section, we do not perform analysis and include more general convex inequality constraints. We believe the reader will benefit from this more general exposition, and it will motivate our future work. More precisely, we introduce

$$\mathcal{M} = \{\theta \mid U(\theta) \geq 0\}, \tag{3.1}$$

where U is a concave function, and assume

$$\Theta = \{\theta \mid U(\theta) \geq 0, B\theta = b\}. \tag{3.2}$$

Thus, (1.1) reduces to

$$\min L(\theta)$$

$$\begin{aligned} \text{s.t. } & B\theta = b, \quad (\text{linear equality constraints}) \\ & U(\theta) \geq 0. \quad (\text{inequality constraints}) \end{aligned}$$

In large-scale nonlinear programming, popular methods for solving the above problem include the interior-point method and active-set SQP methods [30]. In this section, we show that this problem can be more efficiently solved using AEPG (1.6) with a suitable choice of T_k .

3.1. Hessian-Riemannian formulation. We start by reviewing the Hessian-Riemannian framework in the geometric context preceding Lemma 2.2. We first consider only inequality constraints; that is,

$$\min\{L(\theta) \mid \theta \in \mathcal{M}\}, \quad (3.3)$$

where $L \in C^1$ is bounded from below, and $\mathcal{M} \subset \mathbb{R}^n$ is a closed convex set such that $\text{int}(\mathcal{M}) \neq \emptyset$. Standard gradient descent for (3.3) may not necessarily stay in \mathcal{M} . To address this limitation, one approach is to introduce a suitable Riemannian metric on \mathcal{M} that shrinks the gradients near $\partial\mathcal{M}$ to prevent updates from leaving \mathcal{M} .

Assume \mathcal{M} is endowed with a metric G as in (2.18). Then for a smooth $f: \mathcal{M} \rightarrow \mathbb{R}$ the metric gradient and chain rule are

$$\nabla^G f(\theta) = G(\theta)^{-1} \nabla f(\theta), \quad \frac{df(\theta)}{dt} = \left\langle \nabla^G f(\theta), \dot{\theta} \right\rangle_{G(\theta)},$$

where $t \mapsto \theta(t) \in \mathcal{M}$ is an arbitrary smooth curve (see Lemma 2.2).

For a convex function L , the variational characterization of $\theta^* \in \arg \min_{\mathcal{M}} L$ is given by:

$$\langle \nabla^G L(\theta), \theta - \theta^* \rangle_{G(\theta)} \geq 0, \quad \forall \theta \in \mathcal{M}.$$

Therefore, if there exists a function V such that $\nabla^G V(\theta) = \theta - \theta^*$, then V serves as a natural Lyapunov function for the gradient flow

$$\dot{\theta} = -\nabla^G L(\theta).$$

Indeed, it follows that

$$\frac{dV(\theta)}{dt} = \langle \nabla^G V(\theta), \dot{\theta} \rangle_{G(\theta)} = -\langle \nabla^G L(\theta), \theta - \theta^* \rangle_{G(\theta)} \leq 0.$$

The existence of such Lyapunov functions is valuable for proving convergence results for gradient descent algorithms. The following theorem [2] characterizes G for which such V can be found. Furthermore, these V are nothing else but Bregman divergences.

THEOREM 3.1 (Theorem 3.1 [2]). *A metric $G \in C^1(\text{int}(\mathcal{M}))$ ensures that for any given $\xi \in \text{int}(\mathcal{M})$, there exists a functional $V_\xi: \text{int}(\mathcal{M}) \rightarrow \mathbb{R}$ satisfying $\nabla^G V_\xi(\theta) = \theta - \xi$ if and only if there exists a strictly convex function $h \in C^3(\text{int}(\mathcal{M}))$ such that $\forall \theta \in \text{int}(\mathcal{M})$, $G(\theta) = \nabla^2 h(\theta)$. Additionally, defining $D_h: \text{int}(\mathcal{M}) \times \text{int}(\mathcal{M}) \rightarrow \mathbb{R}$ by*

$$D_h(\xi, \theta) = h(\xi) - h(\theta) - \langle \nabla h(\theta), \xi - \theta \rangle \quad (3.4)$$

and taking $V_\xi = D_h(\xi, \cdot)$, we get $\nabla^G V_\xi(\theta) = \theta - \xi$.

The preceding discussion motivates the application of the metric

$$G(\theta) = \nabla^2 h(\theta), \quad \theta \in \text{int}(\mathcal{M}), \quad (3.5)$$

for a suitable convex function $h: \text{int}(\mathcal{M}) \rightarrow \mathbb{R}$. There are many choices of h for a fixed \mathcal{M} . The *Legendre type* identified in [2], is a class of h suitable for enforcing \mathcal{M} and performing convergence analysis. Precisely, we say that h is of Legendre type if:

- (1) $h \in C^2(\text{int}(\mathcal{M}))$ is strictly convex,
- (2) $|\nabla h(\xi_j)| \rightarrow +\infty$ for all $\{\xi_j\} \subset \text{int}(\mathcal{M})$ converging to a boundary point of \mathcal{M} .

For constraints described by U as in (3.1), we discuss the construction of h in Appendix B.

3.2. Adding linear equality constraints. We now consider the case when (3.2) holds; that is, we have the optimization problem:

$$\min\{L(\theta) \mid U(\theta) \geq 0, B\theta = b\}. \quad (3.6)$$

Following Section 2.3, we can combine the Hessian-Riemannian metric and the projection operator onto $\ker(B)$. Thus, (3.6) is in the form of (1.1) and can be solved by (1.6) with

$$T_k = P(\theta_k) \nabla^2 h(\theta_k)^{-1} = \nabla^2 h(\theta_k)^{-1} (I - B^\top (B \nabla^2 h(\theta_k)^{-1} B^\top)^{-1} B \nabla^2 h(\theta_k)^{-1}). \quad (3.7)$$

The algorithmic steps for solving (3.6) can be summarized in Algorithm 1.

Algorithm 1 AEPG for solving problem (3.6) with U in the form of (B.3). Good default setting for parameters are $c=1$ and $r_0=l(\theta_0)=\sqrt{L(\theta_0)+c}$

Require: G : a metric in \mathcal{M} and $G(\theta) = \nabla^2 h(\theta)$ for some h of the Legendre type; B : equality constraint matrix; c : a parameter such that $L(\theta) + c > 0$ for all $\theta \in \mathcal{M}$; η^* : an upper bound for the step size; $\epsilon \in [0, \frac{1}{2}]$: a small positive constant; and K : the total number of iterations.

Require: θ_0 : initial guess of θ satisfying $B\theta_0 = b$ and $U_i(\theta_0) \geq 0$ for $i \in [p]$

- 1: **for** $k=0$ to $K-1$ **do**
- 2: Compute: $\nabla^2 h(\theta_k)^{-1}$
- 3: $\nabla l(\theta_k) = \nabla L(\theta_k)/(2l(\theta_0))$
- 4: $A_k^\dagger = \nabla^2 h(\theta_k)^{-1} (I - B^\top (B \nabla^2 h(\theta_k)^{-1} B^\top)^{-1} B \nabla^2 h(\theta_k)^{-1})$
- 5: $v_k = A_k^\dagger \nabla l(\theta_k)$ (compute Riemannian gradient)
- 6: Line search: $\eta_k = \text{clip}(\text{argmax}\{\eta \mid I_i(\eta) \geq \epsilon U_i(\theta_k), i \in [p]\}, 0, \eta^*)$
- 7: $r_{k+1} = r_k / (1 + 2\eta_k \|v_k\|^2)$ (update energy)
- 8: $\theta_{k+1} = \theta_k - 2\eta_k r_{k+1} v_k$ (update parameters)
- 9: **return** θ_K

REMARK 3.1. The line search step in Algorithm 1 is to ensure that $(\theta_k)_{k \geq 0}$ stay in $\text{int}(\mathcal{M})$. Simultaneously, our experiments show that this can be simply guaranteed by choosing η suitably small.

REMARK 3.2. We should point out that the choice of r_0 can significantly affect the performance of AEPG. In our experiments, we observe that the choice $r_0 = l(\theta_0)/\lambda_1$ is better than the default choice $r_0 = l(\theta_0)$ in some scenarios, see Table 6.2. This is

consistent with the theoretical results obtained in Lemma 2.1 under the condition $r_0 > \frac{l(\theta_0) - l_*}{\lambda_1}$, where λ_1 is the smallest eigenvalue of A_k . This remark also applies to Algorithm 2.

4. Natural gradient descent

In this section, we show that HRGD can be cast as an NGD.

4.1. Natural Gradient Descent. We present the Natural Gradient Descent (NGD) following the exposition in [31]. Let (\mathcal{M}, g) be a (formal) Riemannian manifold, $\Theta \subset \mathbb{R}^k$ a closure of a non-empty open set, $\phi: \Theta \mapsto \mathcal{M}$ a smooth forward model that parametrizes \mathcal{M} explicitly, and $f: \mathcal{M} \rightarrow \mathbb{R}$ a smooth function. Furthermore, consider the optimization problem

$$\min_{\theta \in \Theta} f(\phi(\theta)). \quad (4.1)$$

Let $\{\partial_{\theta_i}^g \phi(\theta)\}_{i=1}^k \subset T_{\phi(\theta)} \mathcal{M}$ be the tangent vectors. Then the NGD direction for this problem is given by

$$p^{nat} = -G(\theta)^{-1} \nabla_{\theta} f(\phi(\theta)), \quad (4.2)$$

where

$$G_{ij}(\theta) = \langle \partial_{\theta_i}^g \phi(\theta), \partial_{\theta_j}^g \phi(\theta) \rangle_{g(\phi(\theta))}, \quad 1 \leq i, j \leq k, \theta \in \Theta,$$

is called an *information matrix*. This choice of the metric corresponds to the steepest descent as measured in the “natural metric” of the model-manifold (\mathcal{M}, g) . Hence, there is an inherent robustness with respect to the parameterization $\theta \mapsto \phi(\theta)$ [31].

For simplicity, we assume that $G(\theta)$ is invertible for all $\theta \in \Theta$. Otherwise, $G(\theta)^{-1}$ should be replaced by the pseudoinverse $G(\theta)^\dagger$.

To establish a connection between NGD and Hessian-Riemannian Gradient Descent (HRGD), we first present a variational formulation of p^{nat} in (4.2). The following lemma is elementary and can be found in many works on NGD. Nevertheless, we present it here for the convenience of the reader. For a more comprehensive discussion of the subject, we refer to [31] and numerous references therein.

LEMMA 4.1. *Let p^{nat} be given by (4.2). Then one has that*

$$p^{nat} = \arg \min_{p \in \mathbb{R}^k} \left\| \nabla^g f(\phi(\theta)) + \sum_{i=1}^k p_i \partial_{\theta_i}^g \phi(\theta) \right\|_{g(\phi(\theta))}^2. \quad (4.3)$$

Proof. Expanding the square norm, we have:

$$\left\| \nabla^g f(\phi(\theta)) + \sum_{i=1}^k p_i \partial_{\theta_i}^g \phi(\theta) \right\|_{g(\phi(\theta))}^2 = p^\top G(\theta) p + 2p^\top \nabla_{\theta} f(\phi(\theta)) + \|\nabla^g f(\phi(\theta))\|_{g(\phi(\theta))}^2, \quad (4.4)$$

where we used the chain rule

$$\langle \nabla^g f(\phi(\theta)), \partial_{\theta_i}^g \phi(\theta) \rangle_{g(\phi(\theta))} = \partial_{\theta_i} f(\phi(\theta)), \quad 1 \leq i \leq k.$$

Hence, the first-order optimality condition with respect to p in (4.4) yields (4.3). \square

4.2. Hessian-Riemannian Gradient Descent as a Natural Gradeint Descent. Consider again the constrained optimization problem

$$\min_{x \in \mathbb{R}^d} \{f(x) : U(x) \geq 0, Bx = b\}, \quad (4.5)$$

where U is a concave function, and $B \in \mathbb{R}^{m \times d}$ such that $\text{rank}(B) = m < d$. As before, we denote by

$$\mathcal{M} = \{x : U(x) \geq 0\}, \quad (4.6)$$

and assume that $h : \text{int}(\mathcal{M}) \rightarrow \mathbb{R}$ is a convex function of Legendre type. Recall that the HRGD direction is given by

$$\dot{x} = -P(x)\nabla^2 h(x)^{-1}\nabla f(x), \quad (4.7)$$

where $P(x) : \mathbb{R}^d \rightarrow \ker(B)$ is the $\nabla^2 h(x)$ -orthogonal projection. Our goal is to show that (4.7) can be interpreted as an NGD direction.

Since $\text{rank } B = m$, the solutions of $Bx = b$ have a parametric representation by an affine map $\phi : \mathbb{R}^{d-m} \rightarrow \mathbb{R}^d$; that is,

$$\{x : Bx = b\} = \{\phi(\theta) : \theta \in \mathbb{R}^n, n := d - m\}.$$

Moreover, without loss of generality, we can assume that the Jacobian of ϕ has the form

$$D_\theta \phi(\theta) = \begin{pmatrix} W \\ I \end{pmatrix},$$

where

- $W \in \mathbb{R}^{m \times n}$,
- $I \in \mathbb{R}^{n \times n}$ is the identity matrix,
- $\text{rank } D_\theta \phi(\theta) = n$,
- the column vectors of $D_\theta \phi(\theta)$ form a basis for $\ker(B)$.

Hence, denoting by $\Theta = \phi^{-1}(\mathcal{M})$, we obtain that (4.5) can be written as

$$\min_{\theta \in \Theta} f(\phi(\theta)),$$

recovering the setup in (4.1).

LEMMA 4.2. *Consider the problem (4.5). Suppose that Θ, ϕ are given as above, and \mathcal{M} is equipped with the Hessian metric*

$$\langle v_1, v_2 \rangle_{g(x)} = v_1 \cdot \nabla^2 h(x)v_2, \quad v_1, v_2 \in T_x \mathcal{M} \cong \mathbb{R}^d.$$

Furthermore, assume that $\dot{\theta} = p^{\text{nat}}$, where p^{nat} is defined as in (4.2). Then for $x = \phi(\theta)$ we have that the equality (4.7) is valid.

Proof. From Lemma 4.1 we have that

$$p^{\text{nat}} = \underset{p \in \mathbb{R}^{n-m}}{\text{argmin}} \|\nabla^g f(\phi(\theta)) + D_\theta \phi(\theta) p\|_{\nabla^2 h(\phi(\theta))}^2.$$

Since the columns of $D_\theta\phi(\theta)$ form a basis in $\ker(B)$, we have that vectors of the form $D_\theta\phi(\theta)p$ span the whole subspace $\ker(B)$, and so

$$D_\theta\phi(\theta)p^{nat} = -P(\phi(\theta))\nabla^g f(\phi(\theta)) = -P(\phi(\theta))\nabla^2 h(\phi(\theta))^{-1}\nabla f(\phi(\theta)),$$

where $P(\phi(\theta))$ is the $\nabla^2 h(\phi(\theta))$ -orthogonal projection on $\ker(B)$, and we used the fact that

$$\nabla^g f(x) = \nabla^2 h(x)^{-1}\nabla f(x), \quad \forall x \in \text{int}(\mathcal{M}).$$

Hence, using the Chain Rule, we obtain

$$\dot{x} = D_\theta\phi(\theta)\dot{\theta} = D_\theta\phi(\theta)p^{nat} = -P(\phi(\theta))\nabla^2 h(\theta)^{-1}\nabla f(\phi(\theta)) = -P(x)\nabla^2 h(x)^{-1}\nabla f(x).$$

□

5. Wasserstein metric

In Section 4, we explored the NGD in the context of a general (formal) Riemannian manifold (\mathcal{M}, g) . In this section, our focus shifts to the Riemannian manifold induced by the Wasserstein metric, commonly known as the optimal transportation metric. This metric has gained recent popularity in data science and inverse problem communities, which explains our motivation to pay special attention to the Wasserstein metric and discuss the computational aspects of the corresponding NGD and AEPG algorithms.

In particular, we present how to compute tangent vectors $\{\partial_{\theta_i}^g \phi(\theta)\}_{i=1}^k$ and discuss efficient methods of computing the NGD direction p^{nat} in (4.2) following the discussion in [31]. Subsequently, we combine the Wasserstein NGD with the AEPG algorithm (1.6) and obtain an adaptive Wasserstein NGD algorithm described in Algorithm 2.

Let $\mathcal{M} = \mathcal{P}_{2,ac}(\mathbb{R}^d)$ be the set of Borel probability measures in \mathbb{R}^d with finite second moments that are absolutely continuous with respect to the Lebesgue measure in \mathbb{R}^d . In what follows, we slightly abuse notation, using same symbols for both probability measures and their density functions.

The quadratic Wasserstein distance is then defined as

$$\begin{aligned} W_2(\rho_1, \rho_2) &= \inf_{\pi \in \mathcal{P}_2(\mathbb{R}^{2d})} \left(\int_{\mathbb{R}^{2d}} |x - y|^2 d\pi(x, y) \right)^{\frac{1}{2}} \\ \text{s.t. } \int_{\mathbb{R}^{2d}} \phi(x) d\pi(x, y) &= \int_{\mathbb{R}^d} \phi(x) d\rho_1(x), \quad \forall \phi \in C_c^\infty(\mathbb{R}^d), \\ \int_{\mathbb{R}^{2d}} \psi(y) d\pi(x, y) &= \int_{\mathbb{R}^d} \psi(y) d\rho_2(y), \quad \forall \psi \in C_c^\infty(\mathbb{R}^d), \end{aligned} \tag{5.1}$$

for all $\rho_1, \rho_2 \in \mathcal{M}$. It turns out that W_2 can be interpreted as a geodesic distance on a (formal) Riemannian manifold as follows [1]. For $\rho \in \mathcal{M}$ we set

$$T_\rho \mathcal{M} = \overline{\{\nabla \phi : \phi \in C_c^\infty(\mathbb{R}^d)\}}^{L_\rho^2(\mathbb{R}^d; \mathbb{R}^d)}, \tag{5.2}$$

and

$$\langle v_1, v_2 \rangle_{g(\rho)} = \int_{\mathbb{R}^d} v_1(x) \cdot v_2(x) \rho(x) dx, \quad \forall v_1, v_2 \in T_\rho \mathcal{M}. \tag{5.3}$$

Our goal is to solve the problem

$$\min_{\theta \in \Theta} L(\theta) := f(\rho(\theta, \cdot)), \tag{5.4}$$

where $\Theta \subset \mathbb{R}^n$ is a closure of a non-empty open set, $f: \mathcal{M} \rightarrow \mathbb{R}$, and $\rho(\theta, \cdot) \in \mathcal{M}$ for all $\theta \in \Theta$.

In this setting, the NGD direction of L is given by

$$p^W = -G(\theta)^{-1} \nabla L(\theta), \quad (5.5)$$

where $G(\theta) \in \mathbb{R}^{n \times n}$ is the information matrix

$$G_{ij}(\theta) = \int_{\mathbb{R}^d} \partial_{\theta_i}^W \rho(\theta, x) \cdot \partial_{\theta_j}^W \rho(\theta, x) \rho(\theta, x) dx, \quad 1 \leq i, j \leq n, \quad (5.6)$$

and $\{\partial_{\theta_i}^W \rho\} \subset T_{\rho} \mathcal{M}$ are the suitable tangent vectors.

Lemma 4.1 yields that

$$p^W = \operatorname{argmin}_{p \in \mathbb{R}^n} \left\| \partial_{\rho}^W f + \sum_{i=1}^n p_i \partial_{\theta_i}^W \rho \right\|_{L_{\rho}^2(\mathbb{R}^d; \mathbb{R}^d)}^2, \quad (5.7)$$

where $\partial_{\rho}^W f$ is the Wasserstein gradient of f at ρ .

PROPOSITION 5.1 (Proposition 2.2 in [31]). *Let $\partial_{\rho} f$ and $\{\partial_{\theta_i} \rho\}_{i=1}^n$ be, respectively, the L^2 derivative and tangent vectors; that is, the derivative and tangent vectors in the standard sense of calculus of variations. Then we have that*

$$\partial_{\rho}^W f = \nabla \partial_{\rho} f, \quad (5.8)$$

$$\partial_{\theta_i}^W \rho = \operatorname{argmin}_v \left\{ \|v\|_{L_{\rho}^2(\mathbb{R}^d; \mathbb{R}^d)}^2 \mid -\nabla \cdot (\rho v) = \partial_{\theta_i} \rho \right\}, \quad i = 1, \dots, n. \quad (5.9)$$

An interesting fact is that the minimization problem (5.9) can be characterized by using a potential function [40, Sections 8.1.2 and 8.2], [25, Section 4], [9, Section 3], [18, Section 2].

LEMMA 5.1. *One has that*

$$\min_v \left\{ \|v\|_{L_{\rho}^2(\mathbb{R}^d; \mathbb{R}^d)}^2 \mid -\nabla \cdot (\rho v) = \partial_{\theta_i} \rho \right\} \quad (5.10)$$

$$= \min_{\phi} \left\{ \|\nabla \phi(x)\|_{L_{\rho}^2(\mathbb{R}^d; \mathbb{R}^d)}^2 \mid \int_{\mathbb{R}^d} \phi(x) dx = 0, -\nabla \cdot (\rho \nabla \phi(x)) = \partial_{\theta_i} \rho \right\}. \quad (5.11)$$

This minimization problem thus admits the following solution:

$$\partial_{\theta_i}^W \rho = (\nabla(-\Delta_{\rho})^{-1}) \partial_{\theta_i} \rho, \quad i = 1, \dots, n. \quad (5.12)$$

REMARK 5.1. Note that (5.12) is well-defined at the continuous level, while for computational efficiency, we still use (5.9).

By Proposition 5.1, given the L^2 tangent vectors $\{\partial_{\theta_1} \rho, \dots, \partial_{\theta_n} \rho\}$ and gradient $\partial_{\rho} f$, the Wasserstein natural gradient can be calculated in two steps:

(1) Compute $\tilde{v}_i = \sqrt{\rho} \partial_{\theta_i}^W \rho$ for $i = 1, \dots, n$ by

$$\tilde{v}_i = \operatorname{argmin}_{\tilde{v}} \left\{ \|\tilde{v}\|_{L^2(\mathbb{R}^d; \mathbb{R}^d)}^2 \mid \mathbf{M} \tilde{v} = \partial_{\theta_i} \rho \right\}, \quad \text{where } \mathbf{M} \tilde{v} = -\nabla \cdot (\sqrt{\rho} \tilde{v}). \quad (5.13)$$

Here \tilde{v}_i is uniquely defined by $\partial_{\theta_i} \rho$, denoted by $\mathbf{M}^{\dagger}(\partial_{\theta_i} \rho)$.

(2) Compute the Wasserstein natural gradient by

$$p^W = \underset{p \in \mathbb{R}^n}{\operatorname{argmin}} \left\| \sqrt{\rho} \nabla \partial_\rho f + \sum_{i=1}^n p_i \tilde{v}_i \right\|_{L^2(\mathbb{R}^d; \mathbb{R}^d)}^2. \quad (5.14)$$

Upon further spatial discretization, p^W can be conveniently used for updating θ_k in AEPG algorithm stated below. For further details about the spatial discretization, we refer to [31].

Algorithm 2 AEPG for solving the problem (5.4). Good default setting for parameters are $c = 1$ and $r_0 = l(\theta_0) = \sqrt{L(\theta_0) + c}$.

Require: $\rho(\theta)$, f a loss function; c : a constant such that $L(\theta) + c > 0$, where $L(\theta) = f(\rho(\theta))$; η : base step size; and T : the total number of iterations.

Require: θ_0 : initial guess of θ .

- 1: **for** $k = 0$ to $T - 1$ **do**
- 2: compute p_k^W via (5.13) and (5.14) (update natural gradient)
- 3: $v_k = -p_k^W / 2l(\theta_k)$
- 4: $r_{k+1} = r_k / (1 + 2\eta \|v_k\|^2)$ (update energy)
- 5: $\theta_{k+1} = \theta_k - 2\eta r_{k+1} v_k$ (update parameters)
- 6: **return** θ_T

6. Numerical examples

This section presents a series of optimization examples to illustrate the following two points¹

- (1) The advantages of natural gradient over the standard gradient.
- (2) The enhanced convergence of AEPG over HRGD (1.4) with T_k given by (3.7) and WNGD ((1.4) with $T_k = G(\theta_k)^{-1}$, where $G(\theta)$ is the information matrix given by (5.6)), particularly in addressing ill-conditioned or nonconvex problems.

In Subsection 6.1, we first present benchmark convex and nonconvex constrained optimization problems in the form of (3.3). These problems are solved by HRGD by constructing a Hessian matrix $\nabla^2 h$ dictated by the form of constraints and then applying the AEPG method. We show the advantage of AEPG over HRGD, especially in handling ill-conditioned or nonconvex problems. Furthermore, we apply AEPG to address the D-optimal design problem, showcasing that with the preconditioning matrix identified by the Hessian–Riemannian metric, AEPG exhibits advantages in both efficiency and accuracy.

In Subsection 6.2, we delve into an optimization problem on the Wasserstein Riemannian manifold presented in the form of (4.1). We employ the least-squares formulation (5.13) and (5.14) to efficiently compute the Wasserstein natural gradient. Our results indicate that methods utilizing the standard gradient (GD and AEGD) may get stuck at a local minimum, whereas methods employing the Wasserstein natural gradient (WNGD and AEPG) reliably converge to the global minimum.

Throughout all experiments, we fine-tune the step size of each method to ensure they solve the problem with the minimum number of iterations or the least computational time. For AEPG, we use default setting $c = 1$ and $r_0 = l(\theta_0)$ (refer to Algorithm 1 and Algorithm 2) unless otherwise stated.

¹The code is available at <https://github.com/txping/AEPG>.

6.1. Hessian-Riemannian method. In the first two examples, we assess the performance of HRGD and AEPG on functions with varying condition numbers (specifically, the condition number of $\nabla^2 f$). More precisely, we set the stopping criterion as $|f(x) - f^*| < \epsilon$ and compare the number of iterations each algorithm takes to achieve the specified accuracy. Additionally, we calculate the ratio of HRGD iterations to AEPG iterations. The summarized results are presented in Tables 6.1 and 6.2. Both sets of results indicate that AEPG significantly enhances the convergence of HRGD, particularly in the context of ill-conditioned and nonconvex problems.

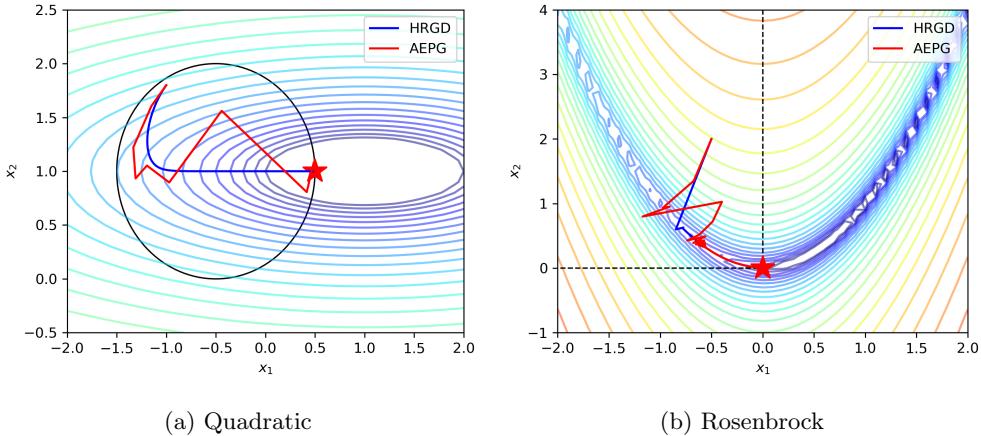


Fig. 6.1: Contour plot and trajectories of AEPG and HRGD on two constrained optimization problems: the quadratic problem with $\alpha=10$ (a), and the Rosenbrock problem with $\alpha=100$ (b). In each plot, the red star represents the minimum point.

6.1.1. Convex objectives.

We begin with the constrained quadratic problem:

$$\begin{aligned} \min \quad & f(x_1, x_2) = (x_1 - 1)^2 + \alpha(x_2 - 1)^2, \\ \text{s.t.} \quad & (x_1 + 0.5)^2 + (x_2 - 1)^2 \leq 1, \end{aligned}$$

where α is a positive constant. With the provided constraint, the minimum value of f is 0.25 achieved at $(0.5, 1)$. In our experiments, we vary the value of α , which corresponds to the condition number of $\nabla^2 f$, and solve the problems using HRGD (1.4) and AEPG (Algorithm 1) with $T_k = \nabla^2 h(x_k)^{-1}$. Here, we set

$$h(x_1, x_2) = u \ln u - u, \quad u := u(x_1, x_2) = 1 - (x_1 + 0.5)^2 - (x_2 - 1)^2.$$

The initial point is set at $(-1, 1.8)$. The results are presented in Table 6.1, and the trajectories of the two methods are visualized in Figure 6.1 (a).

6.1.2. Nonconvex objectives. We then consider the benchmark 2D-Rosenbrock function of the form

$$f(x_1, x_2) = (x_1 - 1)^2 + \alpha(x_2 - x_1^2)^2,$$

where α is a positive constant. This serves as a standard test case for optimization algorithms. The global minimum $(1, 1)$ lies within a long, narrow, parabolic flat valley.

| α | ϵ | Step sizes η | | Number of iterations | | | Computational time (s) | | |
|----------|------------|-------------------|------|----------------------|------|-------|------------------------|--------|-------|
| | | HRGD | AEPG | HRGD | AEPG | Ratio | HRGD | AEPG | Ratio |
| 1 | 1e-7 | 1e-1 | 3e-1 | 416 | 103 | 4 | 0.0234 | 0.0075 | 3 |
| 10 | 1e-6 | 8e-3 | 2e-1 | 3175 | 47 | 67 | 0.1592 | 0.0046 | 35 |
| 100 | 1e-5 | 9e-4 | 9e-3 | 23120 | 723 | 32 | 1.1257 | 0.0533 | 21 |
| 1000 | 1e-4 | 3e-4 | 9e-4 | 14190 | 1715 | 8 | 2.6121 | 0.1349 | 19 |
| 10000 | 1e-3 | 2e-5 | 6e-5 | 147284 | 5075 | 29 | 14.9539 | 0.2805 | 53 |

Table 6.1: Number of iterations and computational time (in seconds) required by HRGD and AEPG to achieve ϵ accuracy and the corresponding step sizes on the constrained quadratic problem. Here α controls the condition number of $\nabla^2 f$; the stopping criteria is $|f(x) - f^*| < \epsilon$; Ratio = number of iterations (computational time) of HRGD / number of iterations (computational time) of AEPG.

Finding the valley is simple, but pinpointing the actual minimum of the function proves less trivial. In this example, we examine the constrained problem:

$$\min \quad f(x_1, x_2) = (x_1 - 1)^2 + \alpha(x_2 - x_1^2)^2, \quad \text{s.t.} \quad x_1 < 0, x_2 > 0.$$

Under the given constraint, the minimum value of f is 1 at $(0,0)$. We vary the value of α , which corresponds to the condition number of $\nabla^2 f$, and solve the problem using HRGD and AEPG (Algorithm 1) with $T_k = \nabla^2 h(x_k)^{-1}$. Here, we use

$$h(x_1, x_2) = K(-x_1) + K(x_2), \quad K(s) = s \ln s - s,$$

leading to $\nabla^2 h(x)^{-1} = \text{diag}(-x_1, x_2)$. The initial point is set at $(-0.5, 2)$. The results are presented in Table 6.2. Numerically, we observe that with a suitably larger step size, AEPG still converges to the minimum point, as shown in Figure 6.1 (b).

| α | ϵ | Step sizes η | | Number of iterations | | | Computational time (s) | | |
|----------|------------|-------------------|------|----------------------|------|-------|------------------------|--------|-------|
| | | HRGD | AEPG | HRGD | AEPG | Ratio | HRGD | AEPG | Ratio |
| 1 | 1e-7 | 2e-1 | 2e-3 | 7896 | 4802 | 2 | 0.2913 | 0.2257 | 1 |
| 10 | 1e-6 | 2e-2 | 2e-4 | 7935 | 1956 | 4 | 0.2604 | 0.1294 | 2 |
| 100 | 1e-5 | 2e-3 | 2e-5 | 8712 | 689 | 12 | 0.3606 | 0.0572 | 6 |
| 1000 | 1e-4 | 2e-4 | 1e-6 | 28705 | 1327 | 21 | 0.8766 | 0.0843 | 10 |
| 10000 | 1e-3 | 2e-5 | 1e-7 | 226524 | 2813 | 80 | 6.6848 | 0.1436 | 46 |

Table 6.2: Number of iterations and computational time (in seconds) required by HRGD and AEPG to achieve ϵ accuracy and the corresponding step sizes on the constrained Rosenbrock problem. For this problem, $r_0 = l(\theta_0)/\lambda_1$ where $\lambda_1 = 0.01$ is used in AEPG. Here α controls the condition number of $\nabla^2 f$; the stopping criteria is $|f(x) - f^*| < \epsilon$; Ratio = number of iterations (computational time) of HRGD / number of iterations (computational time) of AEPG.

6.1.3. D-Optimal Design Problem. Consider the problem of estimating a vector $x \in \mathbb{R}^m$ from measurements $y \in \mathbb{R}^n$ given by the relationship

$$y = Ux + \delta, \quad \delta \sim \mathcal{N}(0, 1),$$

where $U = [u_1, \dots, u_n]^\top$ is a matrix that contains n test (column) vectors $u_i \in \mathbb{R}^m$. During the experiment design phase, a reasonable goal is to minimize the covariance matrix,

which is proportional to $(U^\top U)^{-1}$. Using the D-optimality criteria, the problem is formulated as a minimum determinant problem [39]:

$$\begin{aligned} \min_{\theta} \quad & L(\theta) := \log \det \left(\sum_{i=1}^n \theta_i u_i u_i^\top \right)^{-1}, \\ \text{s.t.} \quad & \sum_{i=1}^n \theta_i = 1 \quad \text{and} \quad \theta_i \geq 0, \quad i \in [n]. \end{aligned} \quad (6.1)$$

This is a convex problem with the unit simplex as the feasible region. In computational geometry, the D -optimal design problem arises as a dual problem of the minimum volume covering ellipsoid (MVCE) problem and finds applications in computational statistics and data mining [39].

To apply AEPG (Algorithm 1) to solve (6.1), we define the Hessian matrix by $\nabla^2 h(\theta_k)$, with

$$h(\theta) = \sum_{i=1}^n K(\theta_i), \quad K(s) = s \ln s - s.$$

From this, we have $\nabla^2 h(\theta)^{-1} = \text{diag}(\theta_1, \dots, \theta_n)$. Note that $B = [1, 1, \dots, 1]$, resulting in the preconditioning matrix, as defined by (3.7), taking the form:

$$\begin{aligned} T_k &= \nabla^2 h(\theta_k)^{-1} - \nabla^2 h(\theta_k)^{-1} B^\top (B \nabla^2 h(\theta_k)^{-1} B^\top)^{-1} B \nabla^2 h(\theta_k)^{-1} \\ &= \text{diag}(\theta_{k,1}, \dots, \theta_{k,n}) - \theta_k \theta_k^\top, \end{aligned} \quad (6.2)$$

where $B \nabla^2 h(\theta_k)^{-1} B^\top = \sum_{i=1}^n \theta_{k,i} = 1$ is used. Notably, from the structure of the preconditioning matrix in (6.2), it is evident that the computation of AEPG is independent of m (the dimension of the test vectors u_i). Hence, AEPG is well-suited for solving D-optimal design problems constructed with high-dimensional test vectors.

Several alternative algorithms have been proposed for solving (6.1), such as the interior point method and the Frank–Wolfe (FW) method [10]. While the interior point method requires the second-order derivative of f , the FW method is a first-order gradient method. To make the comparison more convincing, we also consider FW with away steps (FW-away), an effective strategy that enhances the vanilla FW algorithm's convergence speed and solution accuracy. Further details on applying FW and FW-away to solve the D-optimal design problem can be found in [8, Chapter 5.2.7]. Parameter setup of AEPG can be found in Table D.1 in Appendix D.

In our experiments, we maintain $n=1000$ (number of test vectors) and compare AEPG (Algorithm 1) with other algorithms for various values of m : 10, 30, 50, 80, 100, 200, 300, 400, 500. We generate test vectors u_i using independent random Gaussian distributions with zero mean and unit variance. The initial point is set as $\theta_0 = (\frac{1}{n}, \dots, \frac{1}{n})$. All computations are performed using Python 3.7 on a 2 GHz PC with 16 GB Memory.

Comparison with the interior point method (IPM). For cases where $m=10, 30, 50, 80, 100$, we conduct a comparison analysis between AEPG with IPM². Specifically, we establish the stopping criteria as $|L(\theta) - L^*| < 10^{-7}$ and compare the computation time required for both methods to meet the stopping criteria. The summarized results in Table 6.3 reveal that the computation time of AEPG is significantly less than that of IPM, especially for relatively large m ($m > 10$).

²The interior point method is applied through the Python package PICOS [36].

| m | n | Computational time (s) | | |
|-----|------|------------------------|------|-------|
| | | IPM | AEPG | Ratio |
| 10 | 1000 | 1.39 | 2.84 | 0.5 |
| 30 | 1000 | 6.15 | 2.93 | 2 |
| 50 | 1000 | 21.33 | 3.02 | 7 |
| 80 | 1000 | 112.92 | 2.63 | 43 |
| 100 | 1000 | 253.37 | 2.44 | 104 |

Table 6.3: Comparison of computational time (in seconds) between AEPG and IPM for the D-optimal design problem. The datasets have varying dimensions of test vectors m with a fixed number of test vectors $n=1000$. Ratio = computational time of IPM / computational time of AEPG.

Comparison with the Frank-Wolfe (FW) method. For scenarios where $m=200, 300, 400, 500$, the IPM failed to solve the problems. Consequently, we compare AEPG with the FW method and FW-away method. The results are visually represented in Figure 6.2. Across all cases, the vanilla FW algorithm fails to converge to solutions meeting the stopping criteria, and AEPG consistently requires less time than FW-away to reach the minimum value, particularly when $m=300, 400, 500$.

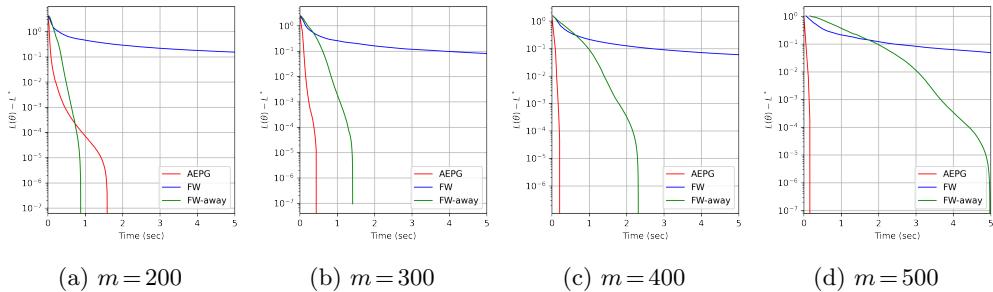


Fig. 6.2: Comparison of computational time (in seconds) between AEPG and the FW/FW-away method for the D-optimal design problem. The datasets have varying dimensions of test vectors m and a fixed number of test vectors $n=1000$.

In both comparative experiments, we observe that unlike the FW-away algorithm and the IPM, whose computation time increases as m gets larger, the computation time of AEPG remains nearly constant across all cases. The advantage of AEPG over the IPM and the FW-away algorithm becomes increasingly evident as m gets larger.

6.2. Wasserstein natural gradient. Consider a 2-dimensional Gaussian mixture model for ρ defined as:

$$\rho(x; \theta) = w\mathcal{N}(x; (\theta_1, 3), I) + (1-w)\mathcal{N}(x; (\theta_2, 2), I),$$

where $w \geq 0$ is a weight factor. In accordance with [31], we formulate the data fitting problem:

$$\min_{\theta} \left\{ L(\theta) = \frac{1}{2} \int_{\Omega} |\rho(x; \theta) - \rho^*(x)|^2 dx \right\},$$

where Ω is a compact domain, and $\rho^*(x)$ is the observed reference density function. In our experiments, we set $w = 0.05$, $\Omega = [0, 5]^2$, and $\rho^*(x) = \rho(x; (1, 3))$. The initial point is set at $(4, 4.2)$.

We apply the least-squares formulation (5.13) and (5.14) to compute the Wasserstein natural gradient and compare the performance of WNGD (1.4) and AEPG (Algorithm 1) with $T_k = G(\theta_k)^{-1}$, where $G(\theta)$ is the Wasserstein information matrix (5.6). The trajectories are presented in Figure 6.3(b). The step sizes used for WNGD and AEPG are 9 and 11, respectively, chosen to optimize performance. Notably, both WNGD and AEPG successfully locate the global minima, with AEPG converging in fewer iterations. Additionally, we showcase the trajectories of GD and AEGD (using standard gradient) in Figure 6.3(a), observing that both methods with standard gradient get stuck at a local minimum.

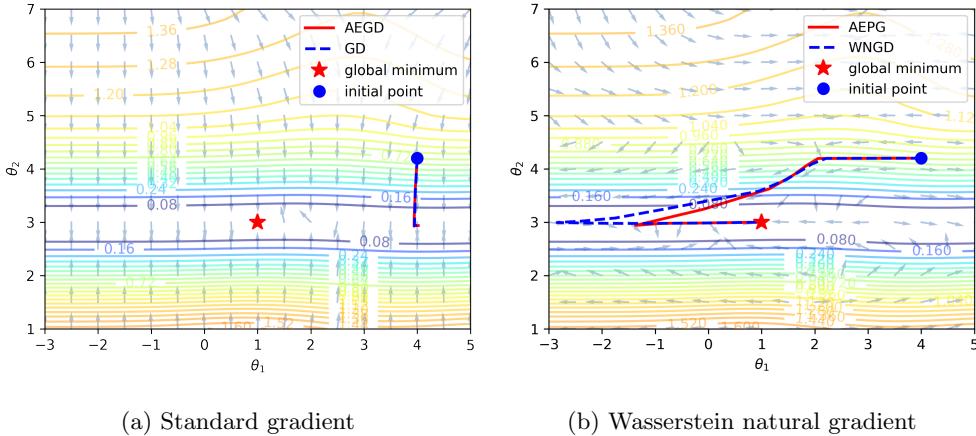


Fig. 6.3: Gaussian mixture model: level sets, vector fields, and convergent paths using (a) methods with standard gradient and (b) methods with the Wasserstein natural gradient.

7. Discussion

This research introduces a unified framework for the application of AEGD (Adaptive Gradient Descent with Energy) techniques, using a general preconditioning descent direction to address a class of constrained optimization problems. The key idea is to incorporate the advantages of adapting the descent direction to the problem's geometry and adjusting the step size through an energy variable.

Theoretical insights reveal that AEPG (Adaptive Energy Preconditioned Gradient) is unconditionally energy stable, independent of the step size. Under the condition of a suitably small base step size, we establish that AEPG is guaranteed to find the minimum value of the objective function. Convergence rates are derived for three types of objective functions: general differentiable functions, nonconvex functions satisfying Polyak–Lojasiewic's (PL) condition, and convex functions when the descent direction is preconditioned by a positive definite matrix against a possible projection matrix.

We investigate two application scenarios where optimization problems with explicit or implicit constraints are considered. In one instance, optimization problems with parameters in a convex set are addressed by endowing the feasible set with a Riemannian metric, following the strategy outlined in [2]. Extension to cases with linear equality

constraints is achieved by adjusting the preconditioning matrix through a projection operator, with convergence theories established under mild assumptions. Another application pertains to optimization problems over the probability density space with the Wasserstein metric. To efficiently compute the natural gradient $G(\theta)^{-1}\nabla l(\theta)$ in AEPG, we adopt the strategy proposed in [31], treating the natural gradient as the solution to a least-squares problem.

Numerical results show that AEPG outperforms vanilla preconditioned gradient descent algorithms such as HRGD and NGD, particularly on ill-conditioned or nonconvex problems. Notably, these results indicate that the choice of the preconditioning matrix not only impacts the convergence rate but influences the stationary point where the iterates converge within a nonconvex optimization landscape.

Optimal choices for Riemannian metrics remain unclear for manifolds with intricate structures. Good options should probably combine specific problem characteristics and applications. Identifying a suitable preconditioning matrix that accelerates convergence without introducing computational overhead is crucial. Moreover, the development of efficient methods for computing the natural gradient in large-scale optimization problems is imperative.

Declarations.

Funding. This work has been partially supported by National Science Foundation (NSF) grants DMS1812666 and DMS2135470 to Hailiang Liu and Xuping Tian; Air Force Office of Scientific Research (AFOSR), Multi-University Research Initiative (MURI) FA 9550 18-1-0502 grant to Levon Nurbekyan; Office of Naval Research (ONR) grant N00014-24-1-2088, and National Science Foundation (NSF) grant DMS-1913129 to Yunan Yang.

Data Availability. The synthetic data supporting Table 3 and Figure 2 are included at <https://github.com/txping/AEPG>.

Conflict of interest. The authors have no relevant financial or non-financial interests to disclose.

Appendix A. Technical proofs. In this section, we provide proofs for some theoretical results mentioned in Section 2:

A.1. Proof of Theorem 2.2. To analyse the convergence behavior of AEPG (1.6), we reformulate it as

$$\theta_{k+1} = \theta_k - \eta_k A_k^{-1} \nabla L(\theta_k), \quad \eta_k := \eta \frac{r_{k+1}}{l(\theta_k)}. \quad (\text{A.1})$$

Using the α -smoothness of L , we have

$$\begin{aligned} L(\theta_{k+1}) &= L(\theta_k - \eta_k A_k^{-1} \nabla L(\theta_k)) \\ &\leq L(\theta_k) - \eta_k \nabla L(\theta_k)^\top A_k^{-1} \nabla L(\theta_k) + \frac{\eta_k^2 \alpha}{2} \|A_k^{-1} \nabla L(\theta_k)\|^2 \\ &\leq L(\theta_k) - \eta_k \|\nabla L(\theta_k)\|_{A_k^{-1}}^2 + \frac{\eta_k^2 \alpha}{2 \lambda_1} \|\nabla L(\theta_k)\|_{A_k^{-1}}^2 \\ &= L(\theta_k) - \eta_k \left(1 - \frac{\eta_k \alpha}{2 \lambda_1}\right) \|\nabla L(\theta_k)\|_{A_k^{-1}}^2. \end{aligned} \quad (\text{A.2})$$

This ensures that $L(\theta_k)$ is strictly decreasing as long as $\|\nabla L(\theta_k)\|_{A_k^{-1}}^2 \neq 0$, providing that

$$0 < \eta_k < \frac{2\lambda_1}{\alpha}. \quad (\text{A.3})$$

Hence we have $L(\theta_k)$ converges as $k \rightarrow \infty$. We further sum (A.2) over k to get

$$\sum_{k=0}^{\infty} \eta_k \left(1 - \frac{\eta_k \alpha}{2\lambda_1}\right) \|\nabla L(\theta_k)\|_{A_k^{-1}}^2 \leq \sum_{k=0}^{\infty} (L(\theta_k) - L(\theta_{k+1})) \leq L(\theta_0) - \inf L(\theta_k) < \infty.$$

This implies

$$\lim_{k \rightarrow \infty} \eta_k \left(1 - \frac{\eta_k \alpha}{2\lambda_1}\right) \|\nabla L(\theta_k)\|_{A_k^{-1}}^2 = 0. \quad (\text{A.4})$$

With $\eta \leq \eta_s$, Lemma 2.1 guarantees that $\eta_k := \eta \frac{r_{k+1}}{l(\theta_k)} \geq \eta \frac{r^*}{l(\theta_0)}$ when $r_0 \geq \frac{l(\theta_0) - l^*}{\lambda_1}$. With $\eta \leq \eta_0 = \frac{\lambda_1 l^*}{\alpha r_0}$, we have $\eta_k := \eta \frac{r_{k+1}}{l(\theta_k)} \leq \frac{\lambda_1}{\alpha}$. These two bounds imply that $\eta_k \left(1 - \frac{\eta_k \alpha}{2\lambda_1}\right) \geq \frac{\eta r^*}{2l(\theta_0)} > 0$, which together with (A.4) ensures that $\|\nabla L(\theta_k)\|_{A_k^{-1}}^2 \rightarrow 0$, hence $\|\nabla L(\theta_k)\| \rightarrow 0$ since A^{-1} is positive definite. Therefore, $\{L(\theta_k)\}_{k=0}^{\infty}$ converges monotonically to $L(\theta^*)$, where θ^* is a local minimizer or a saddle point. Note that here we use the fact that θ^* can not be a local maximizer since $L(\theta_k)$ is decreasing in k .

A.2. Proof of Theorem 2.3.

(ii) Using the α -smoothness assumption and similar derivation as (A.2), we have

$$\begin{aligned} L(\theta_{k+1}) - L(\theta_k) &\leq -\eta_k \left(1 - \frac{\eta_k \alpha}{2\lambda_1}\right) \|\nabla L(\theta_k)\|_{A_k^{-1}}^2 \\ &\leq -\frac{\eta_k}{2} \|\nabla L(\theta_k)\|_{A_k^{-1}}^2 \quad (\text{since } \eta_k \leq \lambda_1/\alpha) \\ &\leq -\frac{\eta_k}{2\lambda_n} \|\nabla L(\theta_k)\|^2. \end{aligned} \quad (\text{A.5})$$

Denote $w_k = L(\theta_k) - L(\theta^*)$, then the PL property of function L implies

$$\frac{1}{2} \|\nabla L(\theta_k)\|^2 \geq \mu(L(\theta_k) - L(\theta^*)) = \mu w_k. \quad (\text{A.6})$$

With this property, (A.5) can be written as

$$w_{k+1} - w_k \leq -\frac{\mu \eta_k}{\lambda_n} w_k \quad \Rightarrow \quad w_{k+1} \leq \left(1 - \frac{\mu \eta_k}{\lambda_n}\right) w_k. \quad (\text{A.7})$$

By induction,

$$w_k \leq w_0 \prod_{j=0}^{k-1} \left(1 - \frac{\mu \eta_j}{\lambda_n}\right) = w_0 \exp \left(\sum_{j=0}^{k-1} \log \left(1 - \frac{\mu \eta_j}{\lambda_n}\right) \right) \leq w_0 \exp \left(-\mu \sum_{j=0}^{k-1} \frac{\eta_j}{\lambda_n} \right).$$

Note for $j < k$ that $\eta_j = \eta \frac{r_{j+1}}{l(\theta_j)} \geq \eta \frac{r_k}{l(\theta_0)}$, hence,

$$w_k \leq w_0 \exp \left(-\frac{c_0 k r_k}{\lambda_n} \right), \quad c_0 = \frac{\mu \eta}{l(\theta_0)}.$$

To ensure convergence of θ_k , we use α -smoothness for L and scheme (A.1) to get

$$\begin{aligned} L(\theta_{k+1}) - L(\theta_k) &\leq \nabla L(\theta_k)^\top (\theta_{k+1} - \theta_k) + \frac{\alpha}{2} \|\theta_{k+1} - \theta_k\|^2 \\ &\leq -\frac{1}{\eta_k} \|\theta_{k+1} - \theta_k\|_{A_k}^2 + \frac{\alpha}{2} \|\theta_{k+1} - \theta_k\|^2 \\ &\leq \left(-\frac{\lambda_1}{\eta_k} + \frac{\alpha}{2}\right) \|\theta_{k+1} - \theta_k\|^2 \\ &\leq -\frac{\lambda_1}{2\eta_k} \|\theta_{k+1} - \theta_k\|^2 \quad (\text{since } \eta_k \leq \lambda_1/\alpha), \end{aligned}$$

which further implies

$$w_k - w_{k+1} \geq \frac{\lambda_1}{2\eta_k} \|\theta_{k+1} - \theta_k\|^2. \quad (\text{A.8})$$

The PL property (A.6) when combined with (A.1) gives

$$\|A_k(\theta_{k+1} - \theta_k)\|^2 = \eta_k^2 \|\nabla L(\theta_k)\|^2 \geq 2\mu\eta_k^2 w_k \Rightarrow \frac{1}{\sqrt{w_k}} \geq \frac{\sqrt{2\mu}\eta_k}{\lambda_n \|\theta_{k+1} - \theta_k\|}. \quad (\text{A.9})$$

Using $w_k > w_{k+1}$, we have

$$\sqrt{w_k} - \sqrt{w_{k+1}} \geq \frac{1}{2\sqrt{w_k}} (w_k - w_{k+1}) \geq \frac{\sqrt{2\mu}\lambda_1}{4\lambda_n} \|\theta_{k+1} - \theta_k\|.$$

Here the last inequality is by (A.8) and (A.9). Taking summation over k gives

$$\sum_{k=0}^{\infty} \|\theta_{k+1} - \theta_k\| \leq \frac{4\lambda_n}{\sqrt{2\mu}\lambda_1} \sqrt{w_0}.$$

This yields (2.14), which ensures the convergence of $\{\theta_k\}$.

(iii) With the convexity assumption on L , we have

$$w_k := L(\theta_k) - L(\theta^*) \leq \nabla L(\theta_k)^\top (\theta_k - \theta^*) \leq \|\nabla L(\theta_k)\| \|\theta_k - \theta^*\|,$$

where we used the Cauchy-Schwarz inequality. We claim that for convex f ,

$$\|\theta_k - \theta^*\| \leq \|\theta_0 - \theta^*\|. \quad (\text{A.10})$$

The proof of this claim is deferred to the end of this subsection. Thus we have

$$\|\nabla L(\theta_k)\| \geq \frac{w_k}{\|\theta_k - \theta^*\|} \geq \frac{w_k}{\|\theta_0 - \theta^*\|}. \quad (\text{A.11})$$

This when combined with (A.5) (since $\eta_k \leq \lambda_1/\alpha$) leads to

$$w_{k+1} \leq w_k - \frac{\eta_k}{2\lambda_n \|\theta_0 - \theta^*\|^2} w_k^2. \quad (\text{A.12})$$

This implies $w_k \geq w_{k+1}$. Multiplying $\frac{1}{w_{k+1} w_k}$ on both sides gives

$$\frac{1}{w_k} \leq \frac{1}{w_{k+1}} - \frac{\eta_k}{2\lambda_n \|\theta_0 - \theta^*\|^2} \frac{w_k}{w_{k+1}} \leq \frac{1}{w_{k+1}} - \frac{\eta_k}{2\lambda_n \|\theta_0 - \theta^*\|^2}.$$

Using $\eta_j = \eta \frac{r_{j+1}}{l(\theta_j)} \geq \eta \frac{r_k}{l(\theta_0)}$, we proceed to obtain

$$\begin{aligned} \frac{1}{w_k} &\geq \frac{1}{w_{k-1}} + \frac{\eta_{k-1}}{2\lambda_n \|\theta_0 - \theta^*\|^2} \\ &\geq \frac{1}{w_0} + \frac{1}{2\lambda_n \|\theta_0 - \theta^*\|^2} \sum_{j=0}^{k-1} \eta_j \\ &\geq \frac{k\eta r_k}{2\lambda_n l(\theta_0) \|\theta_0 - \theta^*\|^2}. \end{aligned}$$

Hence for $\max_{j < k} \eta_j \leq \lambda_1/\alpha$, we have

$$L(\theta_k) - L(\theta^*) = w_k \leq \frac{c_1 \lambda_n \|\theta_0 - \theta^*\|^2}{kr_k}, \quad c_1 = \frac{2l(\theta_0)}{\eta}.$$

Finally, we prove claim (A.10). We proceed with

$$\begin{aligned} \theta_{k+1} - \theta^* &= \theta_k - \theta^* - \eta_k A_k^{-1} \nabla L(\theta_k) \\ &= \theta_k - \theta^* - \eta_k A_k^{-1} (\nabla L(\theta_k) - \nabla L(\theta^*)) \\ &= \theta_k - \theta^* - \eta_k A_k^{-1} \left(\int_0^1 \nabla^2 L(\theta^* + s(\theta_k - \theta^*)) ds \right) (\theta_k - \theta^*). \end{aligned}$$

Denote $d_k := \|\theta_k - \theta^*\|$, we have

$$\begin{aligned} d_{k+1} &= \|(I - \eta_k A_k^{-1} B_k)(\theta_k - \theta^*)\| \\ &\leq \max_{0 \leq \xi \leq \frac{\alpha}{\lambda_1}} |1 - \eta_k \xi| d_k. \end{aligned}$$

For $\eta_k \leq \lambda_1/\alpha$, we have

$$d_{k+1} \leq d_k,$$

hence $d_k \leq d_0$ for any integer k . This completes the proof.

A.3. Proof of Theorem 2.4.

(3) The proof is similar to the proof for Theorem 2.2. Using α -smoothness, we have

$$\begin{aligned} L(\theta_{k+1}) &= L(\theta_k - \eta_k T_k \nabla L(\theta_k)) \\ &\leq L(\theta_k) - \eta_k \nabla L(\theta_k)^\top T_k \nabla L(\theta_k) + \frac{\eta_k^2 \alpha}{2} \|T_k \nabla L(\theta_k)\|^2. \end{aligned} \quad (\text{A.13})$$

Since $T_k = P_k G_k^{-1} = G_k^{-1} P_k^\top$, we have

$$\begin{aligned} L(\theta_{k+1}) &\leq L(\theta_k) - \eta_k \|P_k^\top \nabla L(\theta_k)\|_{G_k^{-1}}^2 + \frac{\alpha \eta_k^2}{2\lambda_1} \|P_k^\top \nabla L(\theta_k)\|_{G_k^{-1}}^2 \\ &= L(\theta_k) - \eta_k \left(1 - \frac{\eta_k \alpha}{2\lambda_1} \right) \|P_k^\top \nabla L(\theta_k)\|_{G_k^{-1}}^2. \end{aligned} \quad (\text{A.14})$$

This ensures that $L(\theta_k)$ is strictly decreasing as long as $\|P_k^\top \nabla L(\theta_k)\|_{G_k^{-1}}^2 \neq 0$, providing that (A.3) holds. For the rest of the proof, we refer the readers to (A.1).

(4) We now turn to estimate convergence rates, while we use

$$P_k = I - G_k^{-1} B^\top (B G_k^{-1} B^\top)^{-1} B$$

so that $B P_k = 0$.

(i) In the proof of (i) for Theorem 2.3, we replace $v_k = G_k^{-1} \nabla l(\theta_k)$ by $G_k^{-1} P_k^\top \nabla l(\theta_k)$, thus obtain the stated convergence bound for $\min_{j < k} \|P_j^\top \nabla L(\theta_j)\|^2$.

(ii) We first show convergence of θ_k . Using $\eta_k \leq \frac{\lambda_1}{\alpha}$, we further bound (A.14) by

$$L(\theta_{k+1}) \leq L(\theta_k) - \frac{\eta_k}{2} \|P_k^\top \nabla L(\theta_k)\|_{G_k^{-1}}^2. \quad (\text{A.15})$$

This and

$$\theta_{k+1} - \theta_k = -\eta_k G_k^{-1} P_k^\top \nabla L(\theta_k) \in K, \quad K := \{p \mid Bp = 0\}$$

lead to

$$L(\theta_{k+1}) - L(\theta_k) \leq -\frac{\lambda_1}{2\eta_k} \|\theta_{k+1} - \theta_k\|^2 \Rightarrow w_k - w_{k+1} \geq \frac{\lambda_1}{2\eta_k} \|\theta_{k+1} - \theta_k\|^2. \quad (\text{A.16})$$

The projected PL property (2.19) when combined with (A.16) gives

$$\|G_k(\theta_{k+1} - \theta_k)\|^2 = \eta_k^2 \|P_k^\top \nabla L(\theta_k)\|^2 \geq 2\mu\eta_k^2 w_k \Rightarrow \frac{1}{\sqrt{w_k}} \geq \frac{\sqrt{2\mu}\eta_k}{\lambda_n \|\theta_{k+1} - \theta_k\|}. \quad (\text{A.17})$$

Combining (A.16) with (A.17) gives

$$\sqrt{w_k} - \sqrt{w_{k+1}} \geq \frac{1}{2\sqrt{w_k}} (w_k - w_{k+1}) \geq \frac{\sqrt{2\mu}\lambda_1}{4\lambda_n} \|\theta_{k+1} - \theta_k\|.$$

This ensures that $\{\theta_k\}$ is a Cauchy sequence, hence $\theta_k \rightarrow \theta^*$ as $k \rightarrow \infty$.

Next we show the convergence rate of $w_k = L(\theta_k) - L(\theta^*)$. From (A.15) we have

$$w_{k+1} - w_k = L(\theta_{k+1}) - L(\theta_k) \leq -\frac{\eta_k}{2\lambda_n} \|P_k^\top \nabla L(\theta_k)\|^2 \quad (\text{A.18})$$

for $\eta_k \leq \frac{\lambda_1}{\alpha}$. By the projected PL condition (2.19), we have

$$\|P_k^\top \nabla L(\theta_k)\|^2 \geq 2\mu w_k,$$

hence

$$w_{k+1} \leq w_k \left(1 - \frac{\mu\eta_k}{\lambda_n}\right),$$

which is exactly (A.7), hence an induction argument will imply

$$w_k \leq w_0 \exp\left(-\frac{c_0 k r_k}{\lambda_n}\right), \quad c_0 = \frac{\mu\eta}{l(\theta_0)}.$$

(iii) For linear constraint $q(\theta) = B\theta - b$, we have $\theta_{k+1} - \theta_k \in K = \{p \mid Bp = 0\}$ with $B = \nabla q(\theta_k)$, hence $\theta_k - \theta^* \in K = \{p \mid Bp = 0\}$. By the convexity of L ,

$$\begin{aligned} w_k := L(\theta_k) - L(\theta^*) &\leq \nabla L(\theta_k)^\top (\theta_k - \theta^*) = \nabla L(\theta_k)^\top P_k (\theta_k - \theta^*) \\ &= (P_k^\top \nabla L(\theta_k))^\top (\theta_k - \theta^*) \leq \|P_k^\top \nabla L(\theta_k)\| \|\theta_k - \theta^*\|, \end{aligned}$$

which implies

$$\|P_k^\top \nabla L(\theta_k)\| \geq \frac{w_k}{\|\theta_k - \theta^*\|} \geq \frac{w_k}{\|\theta_0 - \theta^*\|}.$$

This when combined with (A.18) gives

$$w_{k+1} - w_k \leq -\frac{\eta_k}{2\lambda_n} \|P_k^\top \nabla L(\theta_k)\|^2 \leq -\frac{\eta_k w_k^2}{2\lambda_n \|\theta_0 - \theta^*\|^2}.$$

This is the same as (A.12), and the remaining argument in the proof for Theorem 2.3 (iii) applies here.

Appendix B. Construction of h . We present a construction of h for \mathcal{M} when (3.1) holds with a smooth concave U . We define

$$h(\theta) = K(U(\theta)), \quad \theta \in \text{int}(\mathcal{M}),$$

where $K: (0, \infty) \rightarrow \mathbb{R}$ is a smooth function. It is important to note that:

$$\begin{aligned} \nabla h(\theta) &= K'(U(\theta)) \nabla U(\theta), \\ \nabla^2 h(\theta) &= K''(U(\theta)) \nabla U(\theta) \otimes \nabla U(\theta) + K'(U(\theta)) \nabla^2 U(\theta), \end{aligned} \tag{B.1}$$

and it is clear that K should be chosen to satisfy the following conditions:

- (i) $\lim_{s \rightarrow 0^+} K'(s) = -\infty$,
- (ii) $\forall s > 0, K''(s) > 0$,
- (iii) $\forall s > 0, K'(s) < 0$ when U is not an affine function.

Two commonly used functions for K are $-\ln(s)$ and $s \ln(s) - s$. Additional admissible choices can be found in [2].

LEMMA B.1. *If K is selected to satisfy conditions (i)-(iii) as mentioned above, the strict convexity of $h = K(U)$ can only be ensured if*

$$\xi \cdot \nabla U(\theta) = 0, \quad \xi^T \nabla^2 U(\theta) \xi = 0 \implies \xi = 0, \quad \forall \theta \in \text{int}(\mathcal{M}). \tag{B.2}$$

Proof. For any $\theta \in \text{int}(\mathcal{M})$, and $\xi \in \mathbb{R}^n$, based on the properties of K , we have

$$\xi^T \nabla^2 h(\theta) \xi = K''(U(\theta)) |\nabla U(\theta) \xi|^2 + K'(U(\theta)) \xi^T \nabla^2 U(\theta) \xi^T \geq 0.$$

Equality holds if and only if

$$\xi \cdot \nabla U(\theta) = 0 \quad \text{and} \quad \xi^T \nabla^2 U(\theta) \xi = 0.$$

Hence, (B.2) implies the strict convexity of h . □

REMARK B.1. When $K(U)$ does not satisfy (B.2), adding a correction term is sufficient to ensure the strict convexity of h in \mathcal{M} . For instance, for the domain

$$\mathcal{M} = \{\theta \in \mathbb{R}^n \mid U_i(\theta) \geq 0, i \in [p]\} \tag{B.3}$$

with each U_i being concave, a valid choice is

$$h(\theta) = \sum_{i=1}^p K(U_i(\theta)) + \tilde{h}(\theta), \quad \theta \in \mathcal{M},$$

where \tilde{h} is added to ensure the strict convexity of h in \mathcal{M} . If \tilde{h} is strictly convex, then

$$\begin{aligned}\xi^T \nabla^2 h(\theta) \xi &= \sum_{i=1}^p (K''(U_i(\theta)) |\nabla U_i(\theta) \xi|^2 + K'(U_i(\theta)) \xi^T \nabla^2 U_i(\theta) \xi^T) + \xi^T \nabla^2 \tilde{h}(\theta) \xi \\ &\geq \xi^T \nabla^2 \tilde{h}(\theta) \xi > 0, \quad \forall \xi \neq 0.\end{aligned}$$

Non-strictly convex alternatives for \tilde{h} are viable, provided they result in a strongly convex function h when combined with $(U_i)_{i=1}^p$.

EXAMPLE B.1. *We illustrate several specific examples below.*

(1) *When U is strictly concave for the domain $\mathcal{M} = \{\theta \in \mathbb{R}^n \mid U(\theta) \geq 0\}$, with $U = 1 - \theta^T S \theta$, where S symmetric and positive definite, a suitable choice for h is given by*

$$h(\theta) = K(1 - \theta^T S \theta). \quad (\text{B.4})$$

(2) *In case where ∇U spans the entire space \mathbb{R}^n for the domain $\mathcal{M} = \{\theta \in \mathbb{R}^n \mid a_i \leq \theta_i, i \in [n]\}$, a suitable option is*

$$h(\theta) = \sum_{i=1}^n K(\theta_i - a_i). \quad (\text{B.5})$$

(3) *For the domain $\mathcal{M} = \{\theta \in \mathbb{R}^n \mid a_i \leq \theta_i, i \in [p]\}$, where $1 \leq p < n$, a valid choice for h is given by*

$$h(\theta) = \sum_{i=1}^p K(\theta_i - a_i) + \frac{1}{2} \sum_{i=p+1}^n \theta_i^2, \quad (\text{B.6})$$

is a valid choice.

Appendix C. On the projected PL condition. For the constrained minimization problem:

$$\begin{aligned}\min \quad L(\theta) &= \frac{1}{2} (\beta \theta_1^2 + \alpha \theta_2^2), \\ \text{subject to} \quad q(\theta) &= a\theta_1 + b\theta_2 - 1 = 0,\end{aligned}$$

where $ab \neq 0, \alpha \geq \beta > 0$. The minimum point of this problem is given by

$$\theta^* = \left(\frac{a\alpha}{a^2\alpha + b^2\beta}, \frac{b\beta}{a^2\alpha + b^2\beta} \right).$$

Using $\nabla q(\theta) = (a, b)$ and Equation (2.3), the projection matrix is

$$P = \frac{1}{a^2 + b^2} \begin{pmatrix} b^2 & -ab \\ -ab & a^2 \end{pmatrix},$$

and the projected gradient is

$$P^\top \nabla L(\theta) = \frac{b\beta\theta_1 - a\alpha\theta_2}{a^2 + b^2} (b, -a)^\top.$$

To verify the projected PL condition, it is observed that

$$\begin{aligned}\|P^\top \nabla L(\theta)\|^2 &= \frac{(b\beta\theta_1 - a\alpha\theta_2)^2}{a^2 + b^2} = \frac{1}{a^2 + b^2} \left((b\beta + \frac{a^2\alpha}{b})\theta_1 - \frac{a\alpha}{b} \right)^2 \\ &= \frac{1}{(a^2 + b^2)b^2} \left((b^2\beta + a^2\alpha)\theta_1 - a\alpha \right)^2 = \frac{(b^2\beta + a^2\alpha)^2}{(a^2 + b^2)b^2} (\theta_1 - \theta_1^*)^2,\end{aligned}$$

where $\theta_2 = (1 - a\theta_1)/b$ was used in the second equality. Also,

$$\begin{aligned}L(\theta) - L(\theta^*) &= \frac{1}{2}(\beta\theta_1^2 + \alpha\theta_2^2) - \frac{1}{2}(\beta\theta_1^{*2} + \alpha\theta_2^{*2}) \\ &= \frac{1}{2}\beta(\theta_1^2 - \theta_1^{*2}) + \frac{1}{2}\alpha(\theta_2^2 - \theta_2^{*2}) \\ &= \frac{1}{2}\beta(\theta_1^2 - \theta_1^{*2}) + \frac{1}{2}\alpha\left(\left(\frac{1-a\theta_1}{b}\right)^2 - \left(\frac{1-a\theta_1^*}{b}\right)^2\right) \\ &= \left(\frac{1}{2}\beta + \frac{1}{2}\alpha\frac{a^2}{b^2}\right)(\theta_1^2 - \theta_1^{*2}) - \frac{a\alpha}{b^2}(\theta_1 - \theta_1^*) \\ &= \frac{a^2\alpha + b^2\beta}{2b^2}(\theta_1 - \theta_1^*)\left((\theta_1 + \theta_1^*) - \frac{2a\alpha}{a^2\alpha + b^2\beta}\right) \\ &= \frac{a^2\alpha + b^2\beta}{2b^2}(\theta_1 - \theta_1^*)^2.\end{aligned}$$

Hence the projected PL condition holds for $\mu > 0$ if

$$\mu \leq \frac{a^2\alpha + b^2\beta}{a^2 + b^2}.$$

Appendix D. Experimental setting.

| m | n | c | η |
|-----|------|-----|--------|
| 10 | 1000 | 9 | 0.08 |
| 30 | 1000 | 10 | 0.02 |
| 50 | 1000 | 10 | 0.01 |
| 80 | 1000 | 7 | 0.01 |
| 100 | 1000 | 7 | 0.01 |
| 200 | 1000 | 1 | 0.009 |
| 300 | 1000 | 1 | 0.006 |
| 400 | 1000 | 1 | 0.005 |

Table D.1: Parameter setup of AEPG for the D-optimal design problem.

REFERENCES

- [1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, 2008. 1, 2, 5
- [2] F. Alvarez, J. Bolte, and O. Brahic, *Hessian Riemannian gradient flows in convex programming*, SIAM J. Control Optim., 43(2):477–501, 2004. 1, 3, 3.1, 3.1, 3.1, 7, B
- [3] S.-I. Amari, *Natural gradient works efficiently in learning*, Neural Comput., 10(2):251–276, 1998. 1, 1.2

- [4] M. Arbel, A. Gretton, W. Li, and G. Montúfar, *Kernelized Wasserstein natural gradient*, Int. Conf. Learn. Rep., 2020. 1
- [5] J. Barzilai and J.M. Borwein, *Two-point step size gradient methods*, IMA J. Numer. Anal., 8(1):141–148, 1988. 1.2
- [6] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, 2014. 1.2
- [7] S. Bonnabel, *Stochastic gradient descent on Riemannian manifolds*, IEEE Trans. Automat. Control, 58(9):2217–2229, 2013. 1.2
- [8] G. Braun, A. Carderera, C.W. Combettes, H. Hassani, A. Karbasi, A. Mokhtari, and S. Pokutta, *Conditional gradient methods*, arXiv preprint, arXiv:2211.14103, 2022. 6.1.3
- [9] Y. Chen and W. Li, *Optimal transport natural gradient for statistical manifolds with continuous sample space*, Inf. Geom., 3(1):1–32, 2020. 5
- [10] S.D. Ahipasaoglu, P. Sun, and M.J. Todd, *Linear convergence of a modified Frank–Wolfe algorithm for computing minimum-volume enclosing ellipsoids*, Optim. Meth. Softw., 23(1):5–19, 2008. 6.1.3
- [11] T. Dozat, *Incorporating Nesterov momentum into Adam*, Proc. Int. Conf. Learn. Rep., 1–4, 2016. 1.2
- [12] J. Duchi, E. Hazan, and Y. Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, J. Mach. Learn. Res., 12(7):2121–2159, 2011. 1.2
- [13] D. Gabay, *Minimizing a differentiable function over a differential manifold*, J. Optim. Theory Appl., 37:177–219, 1982. 1.2
- [14] J. Hu, A. Milzarek, Z. Wen, and Y. Yuan, *Adaptive quadratically regularized Newton method for Riemannian optimization*, SIAM J. Matrix Anal. Appl., 39(3):1181–1207, 2018. 1.2
- [15] D.P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, Proc. Int. Conf. Learn. Rep., 2015. 1.2
- [16] J.D. Lee, M. Simchowitz, M.I. Jordan, and B. Recht, *Gradient descent only converges to minimizers*, Proc. Mach. Learn. Res., 49:1246–1257, 2016. 2.3
- [17] W. Li, A.T. Lin, and G. Montúfar, *Affine natural proximal learning*, in F. Nielsen and F. Barbaresco (eds.), Geometric Science of Information: 4th International Conference, GSI 2019, Toulouse, France, August 27–29, 2019, Proceedings, Springer, 705–714, 2019. 1
- [18] W. Li and G. Montúfar, *Natural gradient via optimal transport*, Inf. Geom., 1:181–214, 2018. 1, 1.2, 5
- [19] W. Li and J. Zhao, *Wasserstein information matrix*, Inf. Geom., 203–255, 2023. 1.2
- [20] C. Liu and N. Boumal, *Simple algorithms for optimization on Riemannian manifolds with constraints*, Appl. Math. Optim., 82:949–981, 2020. 1.2
- [21] H. Liu and X. Tian, *An adaptive gradient method with energy and momentum*, Ann. Appl. Math., 38(2):183–222, 2022. 1, 1
- [22] H. Liu and X. Tian, *AEGD: adaptive gradient descent with energy*, Numer. Algebra Control Optim., 15(2):315–340, 2025. 1, 1
- [23] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, *On the variance of the adaptive learning rate and beyond*, Proc. Int. Conf. Learn. Rep., 2020. 1.2
- [24] L. Luo, Y. Xiong, Y. Liu, and X. Sun, *Adaptive gradient methods with dynamic bound of learning rate*, Proc. Int. Conf. Learn. Rep., 2019. 1.2
- [25] A. Mallasto, T.D. Haije, and A. Feragen, *A formalization of the natural gradient method for general similarity measures*, in F. Nielsen and F. Barbaresco (eds.), Geometric Science of Information: 4th International Conference, GSI 2019, Toulouse, France, August 27–29, 2019, Proceedings, Springer, 599–607, 2019. 5
- [26] J. Martens, *New insights and perspectives on the natural gradient method*, J. Mach. Learn. Res., 21(1):5776–5851, 2020. 1.2
- [27] Y. Nesterov, *A method of solving a convex programming problem with convergence rate $o(1/k^2)$* , Dokl. Akad. Nauk, 269:543–547, 1983. 1.2
- [28] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Springer Science & Business Media, 87, 2003. 1.2
- [29] Y. Nesterov, A. Gasnikov, S. Guminov, and P. Dvurechensky, *Primal–dual accelerated gradient methods with small-dimensional relaxation oracle*, Optim. Meth. Softw., 36(4):773–810, 2021. 1.2
- [30] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer, 2006. 1, 1, 1.2, 3
- [31] L. Nurbekyan, W. Lei, and Y. Yang, *Efficient natural gradient descent methods for large-scale PDE-based optimization problems*, SIAM J. Sci. Comput., 45(4):A1621–A1655, 2023. 1, 1, 1.2, 4.1, 4.1, 5, 5.1, 5, 6.2, 7
- [32] S. Osher, B. Wang, P. Yin, X. Luo, F. Barekat, M. Pham, and A. Lin, *Laplacian smoothing gradient descent*, Res. Math. Sci., 9(3):55, 2022. 1

- [33] B.T. Polyak, *Some methods of speeding up the convergence of iteration methods*, USSR Comput. Math. Math. Phys., **4**(5):1–17, 1964. 1.2
- [34] B.T. Polyak, *Introduction to Optimization*, Optimization Software Inc., New York, 1987. 1
- [35] S.J. Reddi, S. Kale, and S. Kumar, *On the convergence of Adam and beyond*, Proc. Int. Conf. Learn. Rep., 2019. 1.2
- [36] G. Sagnol and M. Stahlberg, *PICOS: A Python interface to conic optimization solvers*, J. Open Source Softw., **7**(70):3915, 2022. 2
- [37] Z. Shen, Z. Wang, A. Ribeiro, and H. Hassani, *Sinkhorn natural gradient for generative models*, Adv. Neural Inf. Process. Syst., **33**:1646–1656, 2020. 1
- [38] T. Tieleman and G. Hinton, *Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude*, COURSERA: Neural networks for machine learning, **4**:26–31, 2012. 1.2
- [39] L. Vandenberghe, S. Boyd, and S.-P. Wu, *Determinant maximization with linear matrix inequality constraints*, SIAM J. Matr. Anal. Appl., **19**(2):499–533, 1998. 6.1.3, 6.1.3
- [40] C. Villani, *Topics in Optimal Transportation*, Amer. Math. Soc., **58**, 2021. 5
- [41] X. Wu, S.S. Du, and R. Ward, *Global convergence of adaptive gradient methods for an over-parameterized neural network*, arXiv preprint, arXiv:1902.07111, 2019. 1.2
- [42] L. Ying, *Natural gradient for combined loss using wavelets*, J. Sci. Comput., **86**(2):1–10, 2021. 1
- [43] H. Zhang and W.W. Hager, *A nonmonotone line search technique and its application to unconstrained optimization*, SIAM J. Optim., **14**(4):1043–1056, 2004. 1.2
- [44] J. Zhuang, T. Tang, Y. Ding, S.C. Tatikonda, N. Dvornek, X. Papademetris, and J. Duncan, *Adabelief optimizer: Adapting stepsizes by the belief in observed gradients*, Adv. Neural Inf. Process. Syst., **33**:18795–18806, 2020. 1.2