We provide a lower bound showing that the $O(1/k)$ convergence rate of the NoLips method (a.k.a. Bregman Gradient or Mirror Descent) is optimal for the class of functions satisfying the $h$-smoothness assumption. This assumption, also known as relative smoothness, appeared in the recent developments around the Bregman Gradient method, where acceleration remained an open issue.

On the way, we show how to constructively obtain the corresponding worst-case functions by extending the computer-assisted performance estimation framework of Drori and Teboulle (Mathematical Programming, 2014) to Bregman first-order methods, and to handle the classes of differentiable and strictly convex functions.

1. INTRODUCTION

We consider the constrained minimization problem

$$\min_{x \in C} f(x) \quad (P)$$

where $f$ is a convex continuously differentiable function and $C$ is a closed convex subset of $\mathbb{R}^n$. In large-scale settings, first-order methods are particularly popular due to their simplicity and their low cost per iteration.

The (projected) gradient descent (PG) is a classical method for solving (P), and consists in successively minimizing quadratic approximations of $f$, with

$$x_{k+1} = \arg\min_{u \in C} f(x_k) + \langle \nabla f(x_k), u - x_k \rangle + \frac{1}{2\lambda} \|u - x_k\|^2,$$  \quad (PG)

where $\|\cdot\|$ is the Euclidean norm. Although standard, there is often no good reason for making such approximations, beyond our capability of solving this intermediate optimization problem. In other words, this traditional approximation typically does not reflect neither the geometry of $f$ nor that of $C$. A powerful generalization of PG consists in performing instead a Bregman gradient step

$$x_{k+1} = \arg\min_{u \in C} f(x_k) + \langle \nabla f(x_k), u - x_k \rangle + \frac{1}{\lambda} D_h(u, x_k),$$  \quad (BG)

where the Euclidean distance has been replaced by the Bregman distance $D_h(x, y) := h(x) - h(y) - \langle \nabla h(y), x - y \rangle$ induced by some strictly convex and continuously differentiable kernel function $h$. A well-chosen $h$ allows designing first-order algorithms adapted to the geometry of the constraint set and/or the objective function. Of course, a conflicting goal is to choose $h$ such that each iteration (BG) can be solved efficiently in practice, discarding choices such as $h = f$ (which would boil down to solve the original problem at each iteration).

Recently, Bauschke et al. [3] introduced a natural condition for analyzing this scheme, which assumes that the inner objective in the iteration (BG) is an upper bound on $f$. This ensures that performing an iteration decreases the value of the function. This assumption, which we refer to as $h$-smoothness (precisely defined in Def. 2 below), generalizes the standard $L$-smoothness assumption implied by the Lipschitz continuity of $\nabla f$. The Bregman gradient algorithm, also called NoLips in the setting of [3], is thus a natural extension of gradient descent (PG) to objective functions whose geometry is better modeled by a non-quadratic kernel $h$. 

Date: November 28, 2019.

*Last two authors listed in alphabetical order.
 Practical examples of $h$-smoothness arise in Poisson inverse problems [3], quadratic inverse problems [8], rank minimization [12] and regularized higher-order tensor methods [31].

Can we accelerate NoLips? In the Euclidean setting where $h(x) = \frac{1}{2} \| \cdot \|^2$, accelerated projected gradient methods exhibit faster convergence than the vanilla projected gradient algorithm. These methods, which can be traced back to Nesterov [29], are proven to be optimal for $L$-smooth functions and have found a number of successful applications, in e.g. imaging [6]. A natural question is therefore to understand whether the NoLips algorithm can be accelerated in the $h$-smooth setting. This question has been raised in several works, including that of Bauschke, Bolte and Teboulle [3, Section 6], that of Lu, Freund and Nesterov [24, Section 3.4], and the survey of Teboulle [37, Section 6]. Positive answers have already been provided under somewhat strict additional regularity assumptions (see e.g., [1, 39, 20] and discussions in the sequel), while the general case was apparently still open, and relevant in practical applications. In this work, we produce a lower complexity bound proving that NoLips is optimal for the general $h$-smooth setting, and therefore that generic acceleration is out of reach.

In order to do so, we adopt the standard black-box model used for studying complexity of first-order methods [28]. We consider that both $f$ and $h$ are described by first-order oracles, so as to obtain generic complexity results, and we look for worst-case couples of functions $(f, h)$ satisfying the $h$-smoothness assumption. A central idea in our approach is the fact that, when studying the worst-case behavior of Bregman methods in the $h$-smooth setting, $f$ and $h$ can get arbitrarily close to some limiting pathological nonsmooth functions.

The worst-case functions used for proving the lower bound were found using the recent computer-assisted analysis technique, called performance estimation problems (PEPs), and pioneered by [16]. This technique consists in computing the worst-case convergence rate of a given algorithm by solving a numerical optimization problem. We rely on the approach of [34] and show how the PEP methodology can be adapted to the setting of Bregman methods and $h$-smooth functions. Besides discovering worst-case functions for NoLips, solving PEPs is of great interest for conjecturing (and, with some additional work, proving) new results on different settings or algorithms, as we illustrate in the sequel.

1.1. Contributions and paper organization. The main contribution of this work is twofold. First, we provide a lower bound showing that it is impossible to generically accelerate Bregman gradient methods under the appropriate oracle model. More precisely, we show that the $O(1/k)$ rate on function values of NoLips is optimal in the $h$-smooth setting, using a family of worst-case functions that were discovered by solving a Performance Estimation Problem (PEP).

On the way, we develop PEP techniques for Bregman settings, and extend the analysis of [34] for handling classes of differentiable and strictly convex functions. While we present the analysis on the basic NoLips algorithm for readability purposes, our results and methodology can be applied to various Bregman methods, such as inertial variants [1], or the Bregman proximal point scheme for convex minimization and monotone inclusions [18, 10].

The paper is organized as follows. After introducing the setup in Section 2, we prove the optimality of NoLips in Section 3. We expose the framework of computer-aided analysis of Bregman methods in Section 4, including several applications in Section 4.5. We point out that Sections 3 and 4 are both of independent interest and can be read separately.

1.2. Related work.

Bregman methods. The idea of using non-Euclidean geometries induced by Legendre kernels can be traced back to the work of Nemirovskii and Yudin [28]. For nonsmooth objectives, it gave birth to the mirror descent algorithm [7, 5, 21], which generalizes the subgradient method to non-quadratic geometries. It has been proven to be particularly efficient for minimization on the unit simplex, where choosing the entropy kernel turns out to be much more effective and scalable than the Euclidean norm. This approach has been
very successful in online learning; see [9, Chap. 5] and references therein. The use of Bregman distances has also been thoroughly studied for interior proximal methods [11, 36, 18, 1].

The introduction of the $h$-smoothness assumption in [3] has provided a way to adapt the Legendre kernel to the geometry of the objective function $f$ and thus extend the domain of application of the Bregman Gradient method. Subsequent work has focused on nonconvex extensions [8], linear convergence rates under additional assumptions [24, 2], and inertial variants [20, 27].

**Black-box model and lower complexity bounds.** The first-order black-box model, developed initially in the works of Nemirovskii [28] and later Nesterov [30] has allowed to prove optimal complexity for several classes of problems in first-order optimization [13]. The very related work of Guzman and Nemirovskii [19] studies lower bounds of first-order methods for smooth convex minimization (with a particular focus on smoothness being measured $l_p$-norms). The smoothing technique we use in the sequel is reminiscent of their technique. To the best of our knowledge, it does not contain the lower bound obtained in the sequel as a particular case.

**Performance estimation problems.** The PEP methodology, proposed initially by [16], was already used to discover optimal methods and corresponding lower bounds in other settings: for smooth convex minimization [16, 22, 13, 15], nonsmooth convex minimization [17, 15], and stochastic optimization [14].

### 1.3. Notations.

We use $\overline{C}$ to denote the closure of a set $C$, int $C$ for its interior and $\partial C$ for its boundary. We denote $(e_1, \ldots, e_n)$ the canonical basis of $\mathbb{R}^n$, and for $p \in \{0, \ldots, n\}$ we write $E_p = \text{Span}(e_1, \ldots, e_p)$ the set of vectors supported by the first $p$ coordinates. $S_n$ denotes the set of symmetric matrices of size $n$. If (P) is an optimization problem, then val(P) stands for its (possibly infinite) value.

Subscripts on a vector denotes the iteration counter, while a superscript such as $x^{(i)}$ denotes the $i$-th coordinate. The set $I = \{0, 1, \ldots, N, \ast\}$ is often used to index the first $N$ iterates of an optimization algorithm as well as the optimal point:

$$\{x_i\}_{i \in I} = \{x_0, x_1, \ldots, x_N, x_*\}.$$

We use the standard notation $\langle \cdot, \cdot \rangle$ for the Euclidean inner product, and $\| \cdot \|$ for the corresponding Euclidean norm. For a vector $x \in \mathbb{R}^n$, we write $\|x\|_{\infty} = \max_{i=1,\ldots,n} |x^{(i)}|$ its $l_\infty$ norm. The other notations are standard from convex analysis; see e.g. [32, 4].

### 2. Algorithmic Setup

In this section, we introduce the base ingredients and technical assumptions on $f$ and $h$ that are used within Bregman first-order methods. In particular, it is necessary to assume $h$ to be Legendre in order to have well-defined iterations of the form (BG).

#### 2.1. Legendre functions.

Let $C$ be a closed convex subset of $\mathbb{R}^n$. The first step in defining Bregman methods is the choice of a Legendre function $h$, or kernel, on $C$. In particular, when $C = \mathbb{R}^n$, the technical definition below reduces to requiring $h$ to be continuously differentiable and strictly convex.

**Definition 1** (Legendre function). [32, Chap. 26] A function $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is called a Legendre function with zone $C$ if

1. $h$ is closed convex proper (c.c.p.),
2. $\text{dom } h = C$,
3. $h$ is continuously differentiable and strictly convex on int dom $h \neq \emptyset$,
4. $\|\nabla h(x_k)\| \to \infty$ for every sequence $\{x_k\}_{k \geq 0} \subset \text{int dom } h$ converging to a boundary point of dom $h$ as $k \to \infty$.

A Legendre function $h$ induces a Bregman distance $D_h$ defined as

$$D_h(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle \quad \forall x \in \text{dom } h, y \in \text{int dom } h.$$
Note that $D_h$ is not a distance in the classical sense, however it enjoys a separation property; due to the strict convexity of $h$ we have $D_h(x, y) \geq 0 \forall x \in \text{dom } h, y \in \text{int dom } h$, and it is equal to zero iff $x = y$.

**Examples.** We list some of the most classical examples of Legendre functions:

- The **Euclidean kernel** $h(x) = \frac{1}{2} \|x\|^2$ with domain $\mathbb{R}^n$, and for which $D_h(x, y) = \frac{1}{2} \|x - y\|^2$ is the Euclidean distance,
- The **Boltzmann-Shannon entropy** $h(x) = \sum_i x^{(i)} \log x^{(i)}$ extended to 0 by setting $0 \log 0 = 0$, whose domain is thus $\mathbb{R}_+^n$,
- The **Burg entropy** $h(x) = \sum_i -\log x^{(i)}$ with domain $\mathbb{R}^n_+$,
- The **quartic kernel** $h(x) = \frac{1}{4} \|x\|^4 + \frac{1}{2} \|x\|^2$ with domain $\mathbb{R}^n$ [8].

We refer the reader to [3, 24] for more examples. It should be emphasized that, while a Legendre function $h$ is required to be differentiable on the interior of its domain, it is not differentiable on the boundary. For instance, the Boltzmann-Shannon entropy is continuous but not differentiable at 0.

**Conjugate of a Legendre function.** We also recall that, if $h$ is a Legendre function, its convex conjugate $h^*$ defined as

$$h^*(y) = \sup_{u \in \mathbb{R}^n} \langle u, y \rangle - h(y)$$

is also Legendre [32, Thm 26.5], and that its gradient is the inverse of $\nabla h$, that is $\nabla h^* = (\nabla h)^{-1}$.

### 2.2. The Bregman Gradient/NoLips algorithm.

We recall the framework of the NoLips algorithm described in [3] for solving the minimization problem (P). As we are interested in studying the complexity, we focus here on the simple Bregman gradient method. Our lower bound will be *a fortiori* valid for the Bregman *proximal* gradient algorithm designed for solving composite problems [3, Eq. (12)].

Let us first state our standing assumptions.

**Assumption 1.**

1. $h$ is a Legendre function with zone $C$,
2. $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a closed convex proper function such that $\text{dom } h \subset \text{dom } f$ and which is continuously differentiable on $\text{int dom } h$,
3. For every $\lambda > 0$, $x \in \text{int dom } h$ and $p \in \mathbb{R}^n$, the problem
   $$\min_{u \in \mathbb{R}^n} \langle p, u - x \rangle + \frac{1}{\lambda} D_h(u, x)$$
   has a unique minimizer, which lies in $\text{int dom } h$,
4. The problem is bounded from below, i.e. $f_* := \inf \{ f(x) : x \in C \} > -\infty$,
5. There exists at least one minimizer $x_* \in \text{argmin}_C f$ such that $x_* \in \text{dom } h$.

Condition (iii) is standard and ensures that the algorithm is well-posed. It is satisfied if, for instance, $h$ is strongly convex or supercoercive [3, Lemma 2]. In Condition (v), we make the requirement that there is a solution $x_*$ to (P) that lies in $\text{dom } h$. This is a nontrivial assumption and we must distinguish two cases:

- if $\text{dom } h$ is closed, as for the Euclidean kernel and the Boltzmann-Shannon entropy, then $C = \text{dom } h$ and the condition is necessarily satisfied for every minimizer.
- If $\text{dom } h$ is open, like for the Burg entropy, Condition (v) may fail as the minimizers $x_*$ can lie on the boundary of $\text{dom } h$, where $h$ is infinite.

In addition to these assumptions, the central property we need in order to apply the Bregman gradient method is the so-called $h$-smoothness, first introduced in [3], also known as relative smoothness [24].

**Definition 2 (h-smoothness).** Let $h$ be a Legendre function with zone $C$, and $f$ a function such that $\text{dom } h \subset \text{dom } f$. We say that $f$ is $h$-smooth if there exists a constant $L > 0$ such that

$$L h - f$$

is convex on $\text{dom } h$. (LC)
**h-smoothness** allows to build a simple global majorant of \( f \); indeed, (LC) implies that

\[
f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + LD_h(x, y) \quad \forall x \in \text{dom } h, \ y \in \text{int dom } h,
\]

and successively minimizing this upper approximation will give birth to the NoLips algorithm.

The **h-smoothness** assumption generalizes the usual smoothness assumption; in particular, when taking the Euclidean kernel \( h(x) = \frac{1}{2} \|x\|^2 \), (LC) reduces to standard smoothness implied by the Lipschitz continuity of \( \nabla f \). To avoid ambiguity, we will refer to this standard Euclidean smoothness as **L-smoothness**.

**Remark.** A particular case of **h-smoothness** appears when \( f \) has a Lipschitz continuous gradient with constant \( \tilde{L} \) and the kernel \( h \) is \( \sigma \)-strongly convex (see e.g., [1, 39]), provided that the norm is Euclidean. Indeed, in this case we have

\[
\begin{align*}
\{ \nabla f \text{ is Lipschitz continuous with constant } \tilde{L} \} & \quad \implies \quad \left\{ \frac{\tilde{L}}{2} \| \cdot \|^2 - f \text{ is convex} \right. \\
& \quad \implies \quad \left. \frac{\tilde{L}}{\sigma} (h - \frac{\sigma}{2} \| \cdot \|^2) + (\frac{\tilde{L}}{2} \| \cdot \|^2 - f) \text{ is convex} \right. \\
& \quad \implies \quad \left. \frac{\tilde{L}}{\sigma} h - f \text{ is convex} \right.
\end{align*}
\]

which shows that \( f \) is **h-smooth** with constant \( \frac{\tilde{L}}{\sigma} \). We use the following convenient notation to characterize functions that satisfy the assumptions for NoLips:

**Definition 3.** We say that the couple of functions \( f, h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\} \) is admissible for NoLips, and write \( (f, h) \in B_L(C) \) if

(i) \( f \) and \( h \) satisfy Assumption 1,

(ii) \( Lh - f \) is convex on \( C \).

Finally, let us denote by \( B_L \) the union of \( B_L(C) \) for all closed convex sets \( C \):

\[
B_L = \bigcup_{n \geq 1} \bigcup_{C \subset \mathbb{R}^n \text{ closed convex}} B_L(C)
\]

With this framework, we can define the Bregman Gradient (BG)/NoLips algorithm for minimizing \( f \). For simplicity, we restrict ourselves to constant step size choice.

**Algorithm 1 Bregman Gradient (BG) / NoLips** [3]

**Input:** \( (f, h) \in B_L(C), x_0 \in \text{int dom } h \), step size \( \lambda \in (0, 1/L] \).

**for** \( k = 0,1, \ldots \) **do**

\[
x_{k+1} = \arg\min_{u \in \mathbb{R}^n} \langle \nabla f(x_k), u - x_k \rangle + \frac{1}{\lambda} D_h(u, x_k)
\]

**end for**

Using the first-order optimality condition, the update (1) can also be written as

\[
x_{k+1} = \nabla h^* \left[ \nabla h(x_k) - \lambda \nabla f(x_k) \right]
\]

involving the gradient \( \nabla h^* \) which we call the **mirror map**.
3. Convergence Rate and Optimality of NoLips

In this section, we begin by recalling the \(O(1/k)\) convergence rate bound for the NoLips algorithm in the setting where \((f, h) \in B_L(C)\). We then proceed to prove that NoLips is an \textit{optimal} algorithm for the class \(B_L(C)\), by showing that this rate is also a \textit{lower bound} for a generic class of Bregman gradient algorithms that we define below. The key elements for proving the lower bound were discovered through the solution to a Performance Estimation Problem (PEP), which will be detailed in Section 4.

3.1. Upper bound. We first state the \(O(1/k)\) convergence rate for NoLips. Comparing to previous work [3], it is slightly different, as it is improved by a factor of 2 and does not involve the so-called \textit{symmetry coefficient}.

**Theorem 1** (NoLips convergence rate). Let \(L > 0, C\) be a closed convex subset of \(\mathbb{R}^n\) and \((f, h) \in B_L(C)\) functions admissible for NoLips. Then the sequence \(\{x_k\}_{k \geq 0}\) generated by Algorithm 1 with constant step size \(\lambda \in (0, 1/L]\) satisfies for \(k \geq 0\)

\[
f(x_k) - f_* \leq \frac{D_h(x_0, x_k)}{\lambda k}
\]

for \(f_* = \min_C f\) and any \(x_* \in \text{argmin}_C f \cap \text{dom} h\).

The proof, whose analytical form has been inferred from solving a PEP, is provided in Section 4.5.1. This result extends the \(O(1/k)\) rate of Euclidean gradient descent for \(L\)-smooth functions to the \(h\)-smooth setting. However, unlike in the Euclidean case, we will show in the next section that this rate is actually neither improvable for NoLips, nor for other Bregman first-order methods satisfying a set of reasonable assumptions.

3.2. A lower bound for \(h\)-smooth Bregman optimization. It is natural to ask whether, under the same assumptions as those of Theorem 1, an \textit{accelerated} Bregman algorithm can be obtained, with a better convergence rate than \(O(1/k)\).

This has already been achieved under additional regularity assumptions, as follows

- in the Euclidean setting, when \(h(x) = \frac{1}{2}\|x\|^2\) and \(f\) has a Lipschitz continuous gradient, the seminal accelerated gradient method of Nesterov [29] enjoys a \(O(1/k^2)\) convergence rate, which is optimal for this class of functions [30].
- When \(h\) is a strongly convex Legendre kernel with closed domain and \(f\) has a Lipschitz continuous gradient, the Improved Interior Gradient Algorithm (IGA) [1] also admits a \(O(1/k^2)\) convergence rate, by using the same momentum technique as Nesterov-type methods.
- Recently, [20] proposed an accelerated Bregman proximal gradient algorithm with rate \(O(1/k^\gamma)\), where \(\gamma \in [1, 2]\) is determined by some crucial \textit{triangle scaling property} of the Bregman distance, whose genericity is unclear.

However, the existence of an accelerated algorithm for the \(h\)-smooth setting is still an open question, and many applications [3] do not satisfy the supplementary assumptions made in the works mentioned above. In this section, we prove that, up to a constant factor of 2, the bound (3) is not improvable, making NoLips an \textit{optimal} algorithm in the black box setting for \((f, h) \in B_L\).

More precisely, we will show in Theorem 2 that for every \(\epsilon \in (0, 1)\) and number of oracle calls \(N\), there is a pair of functions \((f, h) \in B_L(\mathbb{R}^{2N+1})\) such that for any \textit{Bregman gradient algorithm}, the output \(x_N\) returned after performing at most \(N\) oracle calls satisfies

\[
f(x_N) - \min_{\mathbb{R}^{2N+1}} f \geq \frac{L D_h(x_0, x_*)}{2N + 1} \cdot (1 - \epsilon).
\]

But first, we need to clarify what we call a \textit{Bregman gradient algorithm} and define the \textit{oracle calls}. 


3.2.1. *Defining a class of Bregman gradient methods.* We adopt the first-order black-box model, where information about a function can be gained by calling an *oracle* returning the value and gradient of $f$ at a given point. In the Bregman setting, we assume that we also have access to the first-order oracles of the Legendre function $h$ and its conjugate $h^*$.

**Assumption 2.** Let functions $f, h$ be in $B_L(C)$ and $T \geq 0$. An algorithm $A$ is called a Bregman gradient algorithm if it generates at each time step $t = 0, \ldots, T$ a set of vectors $V_t$ from the following process:

1. Set $V_0 = \{x_0\}$, where $x_0 \in \text{int dom } h$ is some initialization point.
2. For $t = 1, \ldots, T - 1$, choose some query point $y_t \in \text{Span}(V_t)$ and perform one of the two following operations:
   - either call the **primal oracle** $(\nabla f, \nabla h)$ at $y_t$ and update
     $$V_{t+1} = V_t \cup \{\nabla f(y_t), \nabla h(y_t)\}.$$  
   - Or call the **mirror oracle** $\nabla h^*$ at $y_t$ as
     $$\nabla h^*(y_t) = \arg\min_{u \in C} h(u) - \langle y_t, u \rangle$$  
     and update
     $$V_{t+1} = V_t \cup \{\nabla h^*(y_t)\}.$$  
3. Output some vector $\overline{x} \in \text{Span}(V_T)$.

This model implicitly assumes that $y_t$ is chosen in the domain of the oracle so as to guarantee the existence of the next iterate.

Such structural assumptions on the class of algorithms are classical from complexity analyses of Euclidean first order methods and are used to prove e.g. the optimality of accelerated first order methods [30]. Assumption 2 is a natural extension to the Bregman setting, allowing additional uses of the oracles associated to the Legendre function $h$. This model can often be relaxed through the use of more involved information theoretic arguments, see e.g., [28, 19, 13].

Here, we focus on Assumption 2 as it is general enough to encompass all Bregman-type methods that use only the oracles $\nabla f, \nabla h$, which we call the **primal oracles**, the map $\nabla h^*$, which we call the **mirror oracle**, and linear operations. One can verify that known Bregman gradient methods, including NoLips and inertial variants such as IGA [1] or the recent algorithm in [20], fit in this model.

Note that $V_t$ can contain both points (in the “primal” space) and directions (in the “dual” space), which might allow some unnatural operations (such as scaling a point), but this enables us to write a model that is simple and very general. Observe also that, as NoLips performs one primal oracle call and one mirror call per iteration, an iteration of NoLips corresponds actually to *two time steps* of the formal procedure in Assumption 1. This is why, in order to avoid ambiguity, we will state our lower bound as a function of the number of oracle calls.

3.2.2. *Proof of the lower bound.*

**Proof intuition.** To find a pair of functions $(f, h)$ which is a difficult instance for all Bregman methods, we use two main ideas. The first is the well-known technique used by Nesterov [30] for proving that $O(1/k^2)$ is the optimal complexity for $L$-smooth convex minimization. He defines a “worst function in the world” that allows any gradient method to discover only one dimension per iteration, hence hiding the minimizer from the algorithm in the last dimensions explored.

The second idea is more specific to our setting, and relies on the fact that the set of admissible functions for NoLips $B_L(C)$ is not closed. In particular, a limit of differentiable functions need not be differentiable. This is why, in our case, we actually have a worst-case sequence of differentiable functions parameterized...
by some parameter $\mu$, whose limit when $\mu \to 0$ is a nonsmooth pathological function. Also, it explains why the lower bound (4) we give is not attained, but rather approached to an arbitrary precision $\epsilon$.

Choosing the objective function. Let us fix a dimension $n \geq 1$ and a positive constant $\eta > 0$. Define the convex function $\hat{f}$ for $x \in \mathbb{R}^n$ by

$$\hat{f}(x) = \max_{i=1, \ldots, n} |x^{(i)} - 1 - \frac{\eta}{i}| = \|x - x_*\|_{\infty}$$

which has an optimal value $\hat{f}_* = 0$ attained at

$$x_* := (1 + \eta, 1 + \frac{\eta}{2}, \ldots, 1 + \frac{\eta}{n}).$$

The behavior of $\hat{f}$ as a pathological function comes from the fact that if at least one of the coordinates of $x$ is zero, then $\hat{f}(x) - \hat{f}_* \geq 1$. Let us first prove a technical lemma about the subdifferential of $\hat{f}$.

Lemma 1. Let $x \in \mathbb{R}^n$ and $v \in \partial \hat{f}(x)$ a subgradient of $\hat{f}$ at $x$. Then

(i) $\|v\|_{\infty} \leq 1$.

(ii) Let $i \in \{1 \ldots n\}$. If $v^{(i)} \neq 0$ then $|x^{(i)} - x_*^{(i)}| = \|x - x_*\|_{\infty}$.

Proof. Write $\hat{f}$ as $\hat{f}(x) = \max_{1 \leq i \leq n} \hat{f}_i(x)$ with $\hat{f}_i(x) = |x^{(i)} - x_*^{(i)}|$. Then, by [30, Lemma 3.1.10], we have

$$\partial \hat{f}(x) = \text{Conv}\{\partial \hat{f}_i(x)|i \in I(x)\}$$

where $I(x) = \{i \in \{1 \ldots n\} | \hat{f}_i(x) = \hat{f}(x)\}$. Hence, (i) follows immediately from the well-known property that the subgradients of the absolute value lie in $[-1, 1]$. (ii) is a consequence of the fact that if $v^{(i)} \neq 0$, then $i \in I(x)$, which means that $|x^{(i)} - x_*^{(i)}| = \|x - x_*\|_{\infty}$. ■

Note that $\hat{f}$ is nonsmooth hence does not fit in our assumptions. We approach it with a smooth function by considering its Moreau proximal envelope $f_{\mu}$ given by

$$f_{\mu}(x) = \min_{u \in \mathbb{R}^n} \hat{f}(u) + \frac{1}{2\mu}\|x - u\|^2$$

(5)

where $\mu \in (0, 1)$ is a small parameter. $f_{\mu}$ is a smoothed version of $\hat{f}$, which will behave similarly to $\hat{f}$ when we choose $\mu$ small enough. Figure 1 illustrates this phenomenon in dimension 2.

For general properties of the Moreau proximal envelope, we refer the reader to [25]. We state the properties that we will need in our analysis.

Lemma 2. $f_{\mu}$ is a differentiable convex function, whose minimum is the same as that of $\hat{f}$. Its gradient at a point $x \in \mathbb{R}^n$ is given by $\nabla f_{\mu}(x) = \mu^{-1}\left(x - \text{prox}_{\hat{f}}^\mu(x)\right)$ where

$$\text{prox}_{\hat{f}}^\mu(x) = \arg\min_{u \in \mathbb{R}^n} \hat{f}(u) + \frac{1}{2\mu}\|x - u\|^2$$

is the Moreau proximal map. Moreover, $\nabla f_{\mu}$ is Lipschitz continuous with constant $1/\mu$.

We now prove the central property of $f_{\mu}$, which states that if the last $n - p$ coordinates of $x$ are small enough, then the gradient $\nabla f_{\mu}(x)$ is supported by the first $p + 1$ coordinates. Recall that we denote $(e_1, \ldots, e_n)$ the canonical basis of $\mathbb{R}^n$ and write, for $p \in \{1 \ldots n\}$, $E_p = \text{Span}(e_1, \ldots, e_p)$ and $E_0 = \{(0, \ldots, 0)\}$.

Lemma 3. Assume that $\mu \in (0, 1)$ and $\eta > 4\mu n^2$. Let $p \in \{0 \ldots n - 1\}$. For any vector $x \in \mathbb{R}^n$ such that

$$\max_{i=p+1, \ldots, n} |x^{(i)}| \leq \mu$$

we have that $\nabla f_{\mu}(x) \in E_{p+1}$. In addition, we have $\|\nabla f_{\mu}(x)\|_{\infty} \leq 1$. 

8
Figure 1. Level lines of function $\hat{f}$ and its smoothed Moreau envelope $f_\mu$ for $n = 2, \mu = 0.2$ and $\eta = 1/2$. Lemma 3 states that if $\mu$ is small enough compared to $\eta$, the behavior of $\hat{f}$ and $f_\mu$ at $x_0 = 0$ is the same. Indeed, the size of the smoothed region where the corners are “rounded” decreases when $\mu$ goes to 0.

**Proof.** Take $x \in \mathbb{R}^n$ such that $\max_{i=p+1, \ldots, n} |x_i| \leq \mu$. By Lemma 2, $\nabla f_\mu$ is given by

$$\nabla f_\mu(x) = \frac{1}{\mu}(x - \text{prox}_f^{\mu}(x))$$

Write $y = \text{prox}_f^{\mu}(x)$. Then, the optimality condition defining the proximal map writes

$$y - x + \mu v = 0$$

where $v \in \partial \hat{f}(y)$, and therefore combining (6) and (7) implies

$$\nabla f_\mu(x) = v \in \partial \hat{f}(y).$$

Assume by contradiction that $\nabla f_\mu(x)$ is not in $E_{p+1}$, meaning that there exists an index $l \in \{p + 2 \ldots n\}$ such that $v^{(l)} \neq 0$. It follows from Lemma 1 that $|(y - x^*)^{(l)}| = \|y - x^*\|_\infty$. Hence we have in particular that $|y^{(l)} - x^*_x^{(l)}| = \|y^{(p+1)} - x^*_x^{(p+1)}\|$. Using Condition (7) to replace $y$ we get

$$|x^*_x^{(l)} + \mu v^{(l)} - x^{(l)}| \geq |x^*_x^{(p+1)} + \mu v^{(p+1)} - x^{(p+1)}|,$$

and recalling the definition of $x^*_x$ we have

$$|1 + \frac{\eta}{l} + \mu v^{(l)} - x^{(l)}| \geq |1 + \frac{\eta}{p+1} + \mu v^{(p+1)} - x^{(p+1)}|.$$

By Lemma 1, $\|v\|_\infty \leq 1$, so for all $i$ we have $1 + \mu v^{(i)} \geq 1 - \mu \|v\|_\infty \geq 0$. In addition, we assumed that $\max_{i=p+1, \ldots, n} |x^{(i)}| \leq \mu < \frac{\eta}{1 + \mu}$ which implies $\frac{\eta}{l} - x^{(i)} \geq 0$ for all $i \geq p + 1$. Therefore, both terms inside the absolute values are nonnegative, it follows that we can drop the absolute values and write

$$\mu(v^{(l)} - v^{(p+1)}) \geq \frac{\eta}{p+1} - \frac{\eta}{l} + x^{(l)} - x^{(p+1)}$$

$$\geq \eta \cdot \frac{l - (p + 1)}{p+1} - 2\mu$$

$$\geq \frac{\eta}{l(p+1)} - 2\mu$$

$$\geq \frac{\eta}{n^2} - 2\mu$$

(9)
therefore
\[ v^{(l)} - v^{(p+1)} \geq \frac{\eta}{\mu n^2} - 2 > 2 \]
because we assumed \( \eta > 4\mu n^2 \). This is a contradiction since \( (v^{(l)} - v^{(p+1)}) \leq 2\|v\|_\infty \leq 2 \). Finally, the second part of the Lemma is a consequence of (8) and the inequality \( \|v\|_\infty \leq 1. \)

We will also need the following Lemma which relates the values of \( \hat{f} \) and \( f_\mu \).

**Lemma 4.** Let \( \mu > 0 \) and \( x \in \mathbb{R}^n \). Then \( f_\mu(x) \geq \hat{f}(x) - \mu \).

**Proof.** Write \( y = \text{prox}^{\hat{f}}_\mu(x) \). By definition of \( f_\mu \) and the proximal map we have
\[
f_\mu(x) = \hat{f}(y) + \frac{1}{2\mu} \|y - x\|^2
\geq \hat{f}(y)
= \|y - x_*\|_\infty
\geq \|x - x_*\|_\infty - \|y - x\|_\infty.
\]

Recall that the optimality condition defining the proximal map writes
\[
\mu^{-1}(x - y) \in \partial \hat{f}(y)
\]
and, since all subgradients of \( \hat{f} \) have coordinates smaller than one (Lemma 1), we have \( \|x - y\|_\infty \leq \mu \). It follows that \( f_\mu(x) \geq \|x - x_*\|_\infty - \|y - x\|_\infty \geq \|x - x_*\|_\infty - \mu = \hat{f}(x) - \mu. \)

**Choosing the kernel.** As for the objective function \( f_\mu \), we will also choose a family of kernels \( h_\mu \), whose properties will be close to the ones of a nonsmooth function as \( \mu \to 0 \).

Let us first define a unidimensional convex function \( \phi_\mu : \mathbb{R} \to \mathbb{R} \) by
\[
\phi_\mu(t) = \begin{cases} 
 t - \mu/2 & \text{if } t \geq \mu \\
 1/2t^2 & \text{elsewhere} 
\end{cases}
\]
which is convex, differentiable and continuous. Now let \( d_\mu : \mathbb{R}^n \to \mathbb{R} \) be defined for \( x \in \mathbb{R}^n \) by
\[
d_\mu(x) = \frac{\mu}{2} \|x\|^2 + \sum_{i=1}^{n} \phi_\mu(x^{(i)}) 
\tag{10}
\]
\( d_\mu \) is a differentiable strictly convex function, whose gradient satisfies, for \( x \in \mathbb{R}^n \) and \( i \in \{1 \ldots n\} \),
\[
\nabla d_\mu(x^{(i)}) = \mu x^{(i)} + \min(1, x^{(i)}/\mu).
\]

From the expression above, we can deduce two crucial properties that we will need in the sequel: for \( x \in \mathbb{R}^n \) and \( i \in \{1 \ldots n\} \) we have
\[
\nabla d_\mu(x^{(i)}) = 0 \quad \text{if and only if} \quad x^{(i)} = 0, \tag{11}
\]
\[
|\nabla d_\mu(x^{(i)})| \leq 1 \quad \text{implies} \quad |x^{(i)}| \leq \mu. \tag{12}
\]

Now, let \( L > 0 \). We define the Legendre kernel \( h_\mu \) for \( x \in \mathbb{R}^n \) as
\[
h_\mu(x) = \frac{1}{L} (f_\mu(x) + d_\mu(x)). \tag{13}
\]
By construction, \( L h_\mu - f_\mu \) is convex, so the \( h \)-smoothness property holds. It is easy to see that Assumption 1 is satisfied as \( h_\mu \) is strongly convex, so we have \( (f_\mu, h_\mu) \in \mathcal{B}_L(\mathbb{R}^n) \).
Proving the zero-preserving property of the oracles. Now that the functions are defined, we are ready to prove that all the oracles involved in the Bregman algorithm allow to discover only one dimension per oracle call.

**Proposition 1** (Zero-preserving property of $\nabla f_\mu, \nabla h_\mu, \nabla h_\mu^*$). Assume that $\mu \in (0, 1)$ and $\eta > 4\mu n^2$. Let $p \in \{0 \ldots n - 1\}$, and $x \in \mathbb{R}^n \cap E_p$ a vector supported by the $p$ first coordinates. Then

$$\nabla f_\mu(x), \nabla h_\mu(x), \nabla h_\mu^*(x) \in E_{p+1}.$$ **Proof.** Let $x \in E_p$. Then $x$ satisfies the assumption of Lemma 3 which proves that $\nabla f_\mu(x) \in E_{p+1}$. By Property (11) of $d_\mu$, we also have that $\nabla d_\mu(x) \in E_p$, which allows us to conclude that

$$\nabla h_\mu(x) = L^{-1}(\nabla f_\mu(x) + \nabla d_\mu(x)) \in E_{p+1}.$$ It remains to prove the result for $\nabla h_\mu^*(x)$. Write $z = \nabla h_\mu^*(x)$, which amounts to say that $\nabla h_\mu(z) = x$, that is

$$\nabla f_\mu(z) + \nabla d_\mu(z) = Lx.$$ using (13). We have $x \in E_p$, hence the $l - th$ coordinate of $x$ is zero and

$$\nabla f_\mu(z)^{(l)} + \nabla d_\mu(z)^{(l)} = 0,$$ for $l \in \{p + 1 \ldots n\}$. Using the second part of Lemma 3 we have that $\|\nabla f_\mu(z)\|_\infty \leq 1$; it follows that

$$|\nabla d_\mu(z)^{(l)}| \leq 1$$ which implies that $|z^{(l)}| \leq \mu$, by property (12) of $d_\mu$. Since this holds for any $l \geq p + 1$, we have established

$$\max_{i=p+1,\ldots,n} |z^{(i)}| \leq \mu.$$ Using Lemma 3 again applied to $z$, we have that $\nabla f_\mu(z) \in E_{p+1}$. Remembering that $\nabla h_\mu(z) = x \in E_p$, by construction, we get

$$\nabla d_\mu(z) = L\nabla h_\mu(z) - \nabla f_\mu(z) \in E_{p+1}.$$ By Property (11) of $d_\mu$, we conclude that $z \in E_{p+1}$, which proves the result. ■

We can now use Proposition 1 inductively to state a lower bound on the performance of any Bregman gradient algorithm applied to $(f_\mu, h_\mu)$.

**Proposition 2.** Let $N \geq 1$ and choose the dimension $n = 2N + 1$. Let $\mu \in (0, 1)$ and $\eta > 4\mu n^2$. Consider the functions $f_\mu, h_\mu : \mathbb{R}^n \to \mathbb{R}$ defined in (5) and (13) respectively. Then, for any Bregman gradient method satisfying Assumption 2 applied to $(f_\mu, h_\mu)$ and initialized at $x_0 = (0, \ldots, 0)$, the output $\vec{x}$ returned after performing at most $N$ calls to each one of the primal and mirror oracles satisfies

$$f_\mu(\vec{x}) - \min_{\mathbb{R}^n} f_\mu \geq \frac{LD_{h_\mu}(x_*, x_0)}{2N + 1} \cdot \frac{1 - \mu}{1 + \mu + \eta + \frac{\mu}{2}(1 + \eta)^2}.$$ **Proof.** The zero-preserving property and the structure of Bregman gradient algorithms described in Assumption 2 implies that the set of vectors $\mathcal{V}_t$ at time $t$ is supported by the first $t$ coordinates, i.e.,

$$\mathcal{V}_t \subset E_t.$$ Indeed, since we initialized $\mathcal{V}_0 = \{x_0\} \subset E_0$, this follows by induction: if at time $t$, we have $\mathcal{V}_t \subset E_t$, then the query point $y_t$ lies also in $E_t$ and thus Proposition 1 states that the oracle output belongs to $E_{t+1}$.

Now, because the algorithm has called at most $N$ times each oracle, it has performed at most $2N$ steps and thus the output point satisfies $\vec{x} \in E_{2N}$, which means that $\vec{x}^{(2N+1)} = 0$. 


We use Lemma 4 to relate $f_\mu(\pi)$ and $\hat{f}(\pi)$. Recalling that $\min_{\mathbb{R}^n} f_\mu = \hat{f}_\mu = 0$, we get
\[
\begin{align*}
f_\mu(\pi) - \min_{\mathbb{R}^n} f_\mu &= f_\mu(\pi) \\
&\geq \hat{f}(\pi) - \mu \\
&\geq |\pi^{(2N+1)} - \pi^{(2N+1)}_\ast| - \mu \\
&= 1 + \frac{\eta}{2N + 1} - \mu \\
&\geq 1 - \mu
\end{align*}
\] (14)
where we used the definition of $\hat{f}$ and the fact that $\pi^{(2N+1)} = 0$.

Let us now upper bound the initial diameter. Remembering that $Lh_\mu = f_\mu + d_\mu$ in (13), we have
\[
LD_{h_\mu}(x_\ast, x_0) = D_{f_\mu}(x_\ast, x_0) + D_{d_\mu}(x_\ast, x_0),
\]
by definition of the Bregman distance. To deal with the first term, we recall that $f_\mu(x_\ast) = 0$ and write
\[
\begin{align*}
D_{f_\mu}(x_\ast, x_0) &= f_\mu(x_\ast) - f_\mu(x_0) - \langle \nabla f_\mu(x_0), x_\ast - x_0 \rangle \\
&= -f_\mu(x_0) - \langle \nabla f_\mu(x_0), x_\ast - x_0 \rangle \\
&\leq -\hat{f}(x_0) + \mu - \langle \nabla f_\mu(x_0), x_\ast - x_0 \rangle \\
&= -1 - \eta + \mu - \langle \nabla f_\mu(x_0), x_\ast - x_0 \rangle
\end{align*}
\]
where we used again Lemma 4 at $x_0 = (0, \ldots, 0)$. Now, Lemma 3 applies to $x_0$ with $p = 0$ and allows to state that $\nabla f_\mu(x_0) \in E_1$ and that $\left\| \nabla f_\mu(x_0) \right\|_\infty \leq 1$. Therefore
\[
|\langle \nabla f_\mu(x_0), x_\ast - x_0 \rangle| = |\nabla f_\mu(x_0)(x_\ast^{(1)} - x_0^{(1)})| \leq |x_\ast^{(1)} - x_0^{(1)}| = 1 + \eta.
\]
Hence we have the following upper bound
\[
D_{f_\mu}(x_\ast, x_0) \leq -1 - \eta + \mu + |\langle \nabla f_\mu(x_0), x_\ast - x_0 \rangle| \leq \mu.
\] (15)

Now, the second term can be directly computed from the definition (10) of $d_\mu$, recalling that $x_\ast^{(i)} \geq 1 \geq \mu$ for $i \in \{0, \ldots n\}$,
\[
\begin{align*}
D_{d_\mu}(x_\ast, x_0) &= d_\mu(x_\ast) - d_\mu(x_0) - \langle \nabla d_\mu(x_0), x_\ast - x_0 \rangle \\
&= d_\mu(x_\ast) \\
&= \sum_{k=1}^{2N+1} \left[ \frac{\mu}{2} (1 + \frac{\eta}{k})^2 + 1 + \frac{\eta}{k} - \mu \right] \\
&\leq (2N + 1) \left[ \frac{\mu}{2} (1 + \eta)^2 + \eta + 1 \right].
\end{align*}
\] (16)
Combining (15) and (16) gives
\[
LD_{h_\mu}(x_\ast, x_0) = D_{f_\mu}(x_\ast, x_0) + D_{d_\mu}(x_\ast, x_0) \\
\leq \mu + (2N + 1) \frac{\mu}{2} (1 + \eta)^2 + \eta + 1 \leq (2N + 1) \left[ \mu + \frac{\mu}{2} (1 + \eta)^2 + \eta + 1 \right].
\]
This bound, along with (14), yields
\[
f_\mu(\pi) - \min_{\mathbb{R}^n} f_\mu \geq 1 - \mu \geq \frac{LD_{h_\mu}(x_\ast, x_0)}{2N + 1} \cdot \frac{1 - \mu}{1 + \mu + \eta + \frac{\mu}{2} (1 + \eta)^2}
\]

hence the desired result. ■
Since constants $\mu, \eta$ can be taken arbitrarily small, we now use Proposition 1 to show that the bound can be approached to any precision and thus prove our main result.

**Theorem 2** (Lower complexity bound for $B_L$). Let $N \geq 1$, a precision $\epsilon \in (0, 1)$ and a starting point $x_0 \in \mathbb{R}^{2N+1}$. Then, there exist functions $(f, h) \in B_L(\mathbb{R}^{2N+1})$ such that for any Bregman gradient method $A$ satisfying Assumption 2 and initialized at $x_0$, the output $\overline{x}$ returned after performing at most $N$ calls to each one of the primal and mirror oracles satisfies

$$f(\overline{x}) - \min_{\mathbb{R}^{2N+1}} f \geq \frac{LD_h(x_*, x_0)}{2N + 1} \cdot (1 - \epsilon).$$

**Proof.** Consider a number $N$ of oracle calls and a target precision $\epsilon \in (0, 1)$. Choose the functions $f_\mu, h_\mu$ defined respectively in Equations (5) and (13) on $\mathbb{R}^n$ with $n = 2N + 1$. These functions satisfy Assumption 1, since their domain is $\mathbb{R}^n$, they are convex, differentiable, and $h_\mu$ is strongly convex. Moreover, $h$-smoothness holds because $Lh_\mu - f_\mu = d_\mu$ is convex by construction. Hence $(f_\mu, h_\mu) \in B_L(\mathbb{R}^n)$.

Because the class of functions $B_L(\mathbb{R}^n)$ is invariant by translation, we can assume without loss of generality that the algorithm is initialized at $x_0 = (0, \ldots, 0)$. Recall that the only conditions our analysis imposed on the parameters $\eta, \mu$ are that $\mu \in (0, 1)$ and $\eta > 4\mu n^2$.

We can therefore choose $\eta = \epsilon/4$ and $\mu = \eta/(5n^2) = \epsilon/(20n^2)$. Under these conditions, Proposition 2 applies and gives that for any point $\overline{x}$ returned by a Bregman gradient algorithm that is initialized at $x_0$ and which performs at most $N$ calls to each oracle we have

$$f_\mu(\overline{x}) - \min_{\mathbb{R}^{2N+1}} f_\mu \geq \frac{LD_{h_\mu}(x_*, x_0)}{2N + 1} \cdot \frac{1 - \mu}{1 + \mu + \eta + \frac{\mu}{2}(1 + \eta)^2}.$$

The last term can be bounded from below, using our choice of $\mu, \eta$, and the fact that $\eta < 1$, as

$$\frac{1 - \mu}{1 + \eta + \mu + \frac{\mu}{2}(1 + \eta)^2} \geq \frac{1 - \mu}{1 + \eta + 3\mu} = \frac{1 - \frac{\epsilon}{20n^2}}{1 + \frac{\epsilon}{4} + \frac{3\epsilon}{20n^2}} \geq \frac{1 - \epsilon/2}{1 + \epsilon/2} \geq 1 - \epsilon$$

yielding the desired result. $\blacksquare$

**Remark.** One could refine the result above in the case where the primal and mirror oracles are not used the same number of times. Indeed, if the primal oracles are called $N_1$ times and the mirror oracle is called $N_2$ times, then the same reasoning shows that the lower bound remains true by replacing $2N$ with $N_1 + N_2$.

Also, our lower bound involves the $h$-smoothness constant $L$ instead of the step size $\lambda$ in (3), but it is equivalent (up to a factor 2) when choosing $\lambda = 1/L$, which is actually the best possible step size choice.

### 4. Computer-aided performance analyses of Bregman first-order methods

In this section, we extend the computer-aided performance estimation framework in [16, 33] to the setting of Bregman methods. In short, these results show how to compute the worst-case convergence rate of a given algorithm by solving a numerical optimization problem, called performance estimation problem (PEP). Solving a PEP offers several benefits, including:

1. Computing (numerically) the exact worst-case complexity of an algorithm on a given class of problems after a fixed number of iterations.
2. Studying the corresponding worst-case functions.
(3) Inferring an analytical proof for upper bounding this complexity through a dual PEP, whose feasible points provide combination of inequalities.

Here, we focus on inferring worst-case functions. In particular, this is how we designed the lower bound provided in Section 3.2. However, solving the PEP is also useful for proving new convergence rates (see Section 4.5.2), or for getting quick numerical insights about the convergence guarantees of an algorithm, like for instance on the inertial algorithm IGA [1] (Section 4.5.3).

To use PEPs on Bregman methods, we extend the analysis in [16, 33] to deal with differentiable and/or strictly convex functions. Previous works on the topic modelled differentiability through an $L$-smoothness parameter for instance on the inertial algorithm IGA [1] (Section 4.5.3), or for getting quick numerical insights about the convergence guarantees of an algorithm, like for instance on the inertial algorithm IGA [1] (Section 4.5.3).

Section 4.5.2), or for getting quick numerical insights about the convergence guarantees of an algorithm, with the following form.

This section is organized as follows. In Section 4.1, we introduce the PEP framework. Sections 4.2-4.4 extend PEPs to the Bregman setting. We provide in Section 4.5 several applications, including the procedure used to find the worst-case functions involved in the proof of the general lower bound in Section 3.2.

4.1. Worst-case scenarios through optimization. We now formulate the task of finding the worst-case performance of Algorithm 1 as an optimization problem. We focus on the analysis of NoLips for simplicity. However, the same ideas are directly applicable to other Bregman-type algorithms like IGA [1] (see Section 4.5.3) or Bregman proximal point [18].

Recall that we write $B_L(C)$ the set of function pairs $(f, h)$ satisfying Assumption 1, such that $Lh - f$ is convex on a convex set $C$. For simplicity, we first focus on the case where functions have full domain, i.e., $C = \mathbb{R}^n$ for some $n \geq 1$. In this setting, the set $B_L(\mathbb{R}^n)$ can be rewritten as

$$B_L(\mathbb{R}^n) = \left\{ f, h : \mathbb{R}^n \to \mathbb{R} \middle| \begin{array}{l}
\text{f is convex, differentiable and has at least one minimizer,} \\
\text{h is strictly convex and differentiable,} \\
\forall \lambda > 0, \forall x, p \in \mathbb{R}^n, \text{the function } u \mapsto (p, u - x) + \frac{1}{\lambda} D_h(u, x) \\
\text{has a unique minimizer in } u.
\end{array} \right\},$$

since all constraints in Assumption 1 about the domains of $f$ and $h$ become unnecessary. The general case when $C$ is a convex subset of $\mathbb{R}^n$ can be treated along the same approach. In fact, from the perspective of performance estimation, we can show that every problem in $B_L(\mathbb{R}^n)$ can be reduced to some problem in $B_L(\mathbb{R}^n)$ with equivalent convergence rate (see Appendix A for details).

Performance estimation problem. Throughout this section, we fix a number of iterations $N \geq 1$, a $h$-smoothness parameter $L > 0$, and a step size $\lambda > 0$. In the currently known analyses of NoLips, worst-case guarantees have the following form

$$f(x_N) - f_* \leq \theta(N, L, \lambda) D_h(x_*, x_0),$$

(17)

For instance, Theorem 1 states this result with $\theta(N, L, \lambda) = 1/(\lambda N)$ when $\lambda \in (0, 1/L)$. We then naturally seek the smallest $\theta(N, L, \lambda)$ such that the bound (17) holds for any functions $(f, h) \in B_L(\mathbb{R}^n)$, that is, solve the optimization problem

$$\text{maximize } \left( f(x_N) - f(x_*) \right) / D_h(x_*, x_0)$$

subject to $(f, h) \in B_L(\mathbb{R}^n)$, $x_*$ is a minimizer of $f$, $x_1, \ldots, x_N$ are generated from $x_0$ by Algorithm 1 with step size $\lambda$,

(PEP)

in the variables $f, h, x_0, \ldots, x_N, x_*$ and $n$. We refer to this problem as a performance estimation problem (PEP). We use the convention $0/0 = 0$ so that the objective is well defined when $x_* = x_0$. Optimizing over the dimension $n$ to get dimension-free bounds allows the problem to admit efficient convex reformulations, as
we will see in the sequel. We seek guarantees that are independent of the kernel $h$, so $h$ is also part of the optimization variables.

We begin by simplifying the problem. First, due to the strict convexity of $h$, the NoLips iteration (1) can be equivalently formulated via the first-order optimality condition
\[
\nabla h(x_{i+1}) = \nabla h(x_i) - \lambda \nabla f(x_i) \quad \forall i \in \{0 \ldots N - 1\}
\]
and, since the domain is $\mathbb{R}^n$, the condition that $x_*$ minimizes $f$ reduces to requiring $\nabla f(x_*) = 0$. Second, the problem is homogeneous in $(f, h)$ (i.e., from a feasible couple $(f, h)$, take any constant $c > 0$ and observe that the couple $(cf, ch)$ is also feasible with the same objective value), hence optimizing the objective function $f(x_N) - f(x_*)$ under the additional constraint $D_h(x_*, x_0) = 1$ produces the same optimal value than the problem above.

Finally, we use the same argument as in [16, 34] and observe that the objective of (PEP) and the algorithmic constraints mentioned above depend solely on the values of the first-order oracles of $f$ and $h$ at the points $x_0, \ldots, x_N, x_*$. Denoting $I = \{0, 1, \ldots, N, *\}$ the indices associated to the points involved in the problem we proceed to write these values as
\[
\{(f_i, g_i)\}_{i \in I} = \{(f(x_i), \nabla f(x_i))\}_{i \in I},
\{(h_i, s_i)\}_{i \in I} = \{(h(x_i), \nabla h(x_i))\}_{i \in I}.
\]
With this notation the NoLips iterations rewrite $s_{i+1} = s_i - \lambda g_i$ for $i \in \{0 \ldots N - 1\}$, and the normalization constraint $D_h(x_*, x_0) = 1$ becomes $h_* - h_0 - \langle s_0, x_* - x_0 \rangle = 1$.

Using this discrete representation of $f$ and $h$, we can reformulate (PEP) equivalently as
\[
\begin{align*}
\text{maximize} & \quad f_N - f_* \\
\text{subject to} & \quad f_i = f(x_i), g_i = \nabla f(x_i), \\
& \quad h_i = h(x_i), s_i = \nabla h(x_i), \quad \text{for all } i \in I \text{ and some } (f, h) \in B_L(\mathbb{R}^n), \\
& \quad g_* = 0, \\
& \quad s_{i+1} = s_i - \lambda g_i \quad \text{for } i \in \{1 \ldots N - 1\}, \\
& \quad h_* - h_0 - \langle s_0, x_* - x_0 \rangle = 1,
\end{align*}
\]
in the variables $n, \{(x_i, f_i, g_i, h_i, s_i)\}_{i \in I}$. The equivalence with the initial problem is guaranteed by the first constraints which are called the interpolation conditions.

It turns out that interpolation conditions for the class $B_L(\mathbb{R}^n)$ are delicate to establish. However, there exist two classes $B_{L^*}(\mathbb{R}^n)$ and $\overline{B}_L(\mathbb{R}^n)$ for which they can be derived. The first class is a restriction of $B_L(\mathbb{R}^n)$ where $f$ and $Lh - f$ are both assumed to be strictly convex:
\[
B_{L^*}(\mathbb{R}^n) = B_L(\mathbb{R}^n) \cap \{f, h : \mathbb{R}^n \to \mathbb{R} \mid f \text{ and } Lh - f \text{ are strictly convex}\}
\]
whereas the second class consists in considering a relaxation with possibly nonsmooth functions:
\[
\overline{B}_L(\mathbb{R}^n) = \{f, h : \mathbb{R}^n \to \mathbb{R} \mid f \text{ and } Lh - f \text{ are convex}\}.
\]
We then have
\[
B_{L^*}(\mathbb{R}^n) \subset B_L(\mathbb{R}^n) \subset \overline{B}_L(\mathbb{R}^n).
\]
With theses classes, we can now define two easier problems. The first one is a restriction of (PEP) defined on the class $B_{L^*}(\mathbb{R}^n)$, under the additional constraint that all iterates are distinct:
\[
\begin{align*}
\text{maximize} & \quad f_N - f_* \\
\text{subject to} & \quad f_i = f(x_i), g_i = \nabla f(x_i), \\
& \quad h_i = h(x_i), s_i = \nabla h(x_i), \quad \text{for all } i \in I \text{ and some } (f, h) \in B_{L^*}(\mathbb{R}^n) \\
& \quad g_* = 0, \\
& \quad s_{i+1} = s_i - \lambda g_i \quad \text{for } i \in \{1 \ldots N - 1\}, \\
& \quad h_* - h_0 - \langle s_0, x_* - x_0 \rangle = 1, \\
& \quad x_i \neq x_j \quad \text{for } i \neq j \in I,
\end{align*}
\]

(PEP)
in the variables \( n, \{ (x_i, f_i, g_i, h_i, s_i) \}_{i \in I} \). The second problem is a relaxation of (PEP), where \( (f, h) \in \overline{\mathcal{B}}_{L}(\mathbb{R}^n) \) are possibly nonsmooth and \( g_i, s_i \) are thus subgradients:

\[
\begin{align*}
\text{maximize} & \quad f_N - f_* \\
\text{subject to} & \quad f_i = f(x_i), g_i \in \partial f(x_i), \quad h_i = h(x_i), \quad s_i \in \partial h(x_i), \quad \frac{Ls_i - g_i}{L} \in \partial (Lh - f)(x_i) \quad \text{for all } i \in I \text{ and some } (f, h) \in \overline{\mathcal{B}}_{L}(\mathbb{R}^n), \\
& \quad g_* = 0, \quad s_{i+1} = s_i - \lambda g_i \quad \text{for } i \in \{1 \ldots N - 1\}, \quad h_* - h_0 - \langle s_0, x_* - x_0 \rangle = 1,
\end{align*}
\]

in the variables \( n, \{ (x_i, f_i, g_i, h_i, s_i) \}_{i \in I} \). We added the technical constraint \( Ls_i - g_i \in \partial (Lh - f)(x_i) \), which is redundant for differentiable functions; but that is necessary in order to establish interpolation conditions in the nonsmooth case.

Because of the inclusions between the feasible sets of these problems, we naturally have

\[
\text{val}(\text{PEP}) \leq \text{val}(\text{PEP}) \leq \text{val}(\overline{\text{PEP}}).
\]

We will prove in the sequel that \( \overline{\text{PEP}} \) can be solved via a semidefinite program and that \( \text{val}(\overline{\text{PEP}}) = \text{val}(\text{PEP}) \) (Theorem 4), allowing to reach our claims.

Note that the relaxed problem (PEP) does not correspond to any practical algorithm, as NoLips is not properly defined for nonsmooth functions \( h \). However, we will see in the sequel that feasible points of this problem correspond to accumulation points of (PEP). In other words, instances of NoLips can get arbitrarily close to pathological nonsmooth functions whose behaviors are captured by (PEP).

In the following sections, we show that problems (PEP) and (PEP) can be cast as semidefinite programs (SDP) [38] and solved numerically using standard packages [26, 23]. The main ingredient consists in showing that interpolation constraints can actually be expressed using quadratic inequalities, as detailed in the next section.

4.2. Interpolation involving differentiability and strict convexity. In this section, we show how to reformulate interpolation constraints for (PEP) and (PEP) as quadratic inequalities. We start by recalling interpolation conditions for the class of \( L \)-smooth and \( \mu \)-strongly convex functions.

**Theorem 3** (Smooth strongly convex interpolation, [34]). Let \( I \) be a finite index set, \( \{ (x_i, f_i, g_i) \}_{i \in I} \in (\mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n)^{|I|} \) and \( 0 \leq \mu \leq L \leq +\infty \). The following statements are equivalent:

(i) There exists a proper closed convex function \( f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\} \) such that \( f \) is \( \mu \)-strongly convex, has a \( L \)-Lipschitz continuous gradient and

\[
f_i = f(x_i), \quad g_i \in \partial f(x_i) \quad \forall i \in I.
\]

(ii) For every \( i, j \in I \) we have

\[
f_i - f_j - \langle g_j, x_i - x_j \rangle \geq \frac{1}{2L} \| g_i - g_j \|^2 + \frac{\mu}{2(1+\mu/L)} \| x_i - x_j - \frac{1}{L}(g_i - g_j) \|^2.
\]

In particular, when \( L = +\infty \) (meaning that we require no smoothness) and \( \mu = 0 \), those conditions reduce to the simpler convex interpolation conditions, reminiscent of subgradient inequalities:

\[
f_i - f_j - \langle g_j, x_i - x_j \rangle \geq 0
\]

In our setting, we want to avoid working with smoothness and strong convexity, so we provide interpolation conditions for the class of differentiable strictly convex functions.

**Proposition 3** (Differentiable and strictly convex interpolation). Let \( I \) be a finite index set and \( \{ (x_i, f_i, g_i) \}_{i \in I} \in (\mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n)^{|I|} \). The following statements are equivalent:
(i) There exists a convex function \( f : \mathbb{R}^n \to \mathbb{R} \) such that \( f \) is differentiable, strictly convex and

\[
 f_i = f(x_i), \quad g_i = \nabla f(x_i) \quad \forall i \in I.
\]

(ii) For every \( i, j \in I \) we have

\[
\begin{align*}
 f_i - f_j - \langle g_j, x_i - x_j \rangle & \geq 0, \\
 f_i - f_j - \langle g_j, x_i - x_j \rangle & > 0 \text{ if } x_i \neq x_j \text{ (strict convexity),} \\
 f_i - f_j - \langle g_j, x_i - x_j \rangle & > 0 \text{ if } g_i \neq g_j \text{ (differentiability).}
\end{align*}
\]

Proof. (i) \( \implies \) (ii). Assume that (i) holds, and choose such a function \( f \). The first inequality of (19) follows from convexity of \( f \). The second inequality follows directly from strict convexity when \( x_i \neq x_j \). Now, to prove the third part, consider the case when we have \( \nabla f(x_i) \neq \nabla f(x_j) \) for some indices \( i, j \). Let us prove the result by contradiction, i.e., assume that

\[
 f(x_i) - f(x_j) - \langle \nabla f(x_j), x_i - x_j \rangle = 0.
\]

Let \( u \in \mathbb{R}^n \), convexity implies that

\[
 f(u) \geq f(x_j) + \langle \nabla f(x_j), u - x_j \rangle.
\]

Combining the above inequality with (20) gives

\[
 f(u) \geq f(x_i) + \langle \nabla f(x_j), u - x_i \rangle \quad \forall u \in \mathbb{R}^n
\]

which shows, by definition of a subgradient, that \( \nabla f(x_j) \in \partial f(x_i) \). Since \( f \) is differentiable at \( x_i \), we have by [32, Thm 25.1] that \( \partial f(x_i) = \{ \nabla f(x_i) \} \) which is a contradiction as we assumed \( \nabla f(x_i) \neq \nabla f(x_j) \). Thus the third part of (19) is proved.

(ii) \( \implies \) (i). Assume that (ii) holds. If for all \( i, j \in I \), we have \( g_i = g_j \) and \( x_i = x_j \), then there is only one point and one subgradient to be interpolated, and the result follows immediately from considering a well-chosen definite quadratic function. In the other case, define

\[
 \nu = \min_{i,j \in I, g_i \neq g_j \text{ or } x_i \neq x_j} f_i - f_j - \langle g_j, x_i - x_j \rangle.
\]

Because of (19) and the finiteness of \( I \), we have that \( \nu > 0 \). Now, define \( r \) as

\[
 r = \max_{i,j \in I} \| g_i - g_j \|^2 + \| x_i - x_j \|^2
\]

so that \( r > 0 \). Condition (19) together with the definitions of \( \nu \) and \( r \) yield that for all \( i, j \in I \) we have

\[
 f_i - f_j - \langle g_j, x_i - x_j \rangle \geq \frac{\nu}{r} \left( \| g_i - g_j \|^2 + \| x_i - x_j \|^2 \right).
\]

Now, let us choose two constants \( 0 < \mu < L < +\infty \) such that

\[
 \frac{1}{L - \mu} \leq \frac{\nu}{r}, \quad \frac{\mu}{1 - \mu/L} \leq \frac{\nu}{r}.
\]

as it suffices to take \( L \) large enough and \( \mu \) small enough. We now proceed to show that the interpolation conditions of Theorem 3 hold with the constants \( \mu, L \) defined above. Using the inequality \( \| u - v \|^2 \leq
2\|u\|^2 + 2\|v\|^2 and (21) we get that for all \(i, j,\)
\[
\frac{1}{2L} \|g_i - g_j\|^2 + \frac{\mu}{2(1 - \mu/L)} \|x_i - x_j\|^2 - \frac{1}{L} (g_i - g_j) \leq \left( \frac{1}{2L} + \frac{\mu}{L(L - \mu)} \right) \|g_i - g_j\|^2 + \frac{\mu}{1 - \mu/L} \|x_i - x_j\|^2
\]
\[
= \frac{1}{L - \mu} \|g_i - g_j\|^2 + \frac{\mu}{1 - \mu/L} \|x_i - x_j\|^2
\]
\[
\leq \frac{\nu}{r} \|g_i - g_j\|^2 + \frac{\nu}{r} \|x_i - x_j\|^2
\]
\[
\leq f_i - f_j - \langle g_j, x_i - x_j \rangle.
\]

Under those conditions, Theorem 3 states that there exists a convex function \(f\) that interpolates \(\{(x_i, f_i, g_i)\}_{i \in I}\) which is \(\mu\)-strongly convex and has \(L\)-Lipschitz continuous gradients. A fortiori, since \(\mu > 0\) and \(L < \infty\), \(f\) is differentiable and strictly convex. Finally, \(f\) is finite on \(\mathbb{R}^n\) since it is \(L\)-smooth. ■

**Remark.** It is easy to adapt the result of Proposition 3 for only one of the two conditions (strict convexity or differentiability), which amounts to choose only the corresponding inequalities in (19).

Using these results, we can now formulate interpolation conditions for the problems (PEP) and (PEP) involving the classes \(\overline{B}_L(\mathbb{R}^n)\) and \(\overline{B}_L(\mathbb{R}^n)\) that were defined in Section 4.1.

**Corollary 1 (Interpolation conditions for (PEP)).** Let \(I\) be a finite index set and \(\{(x_i, f_i, g_i, h_i, s_i)\}_{i \in I} \in (\mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n)^{|I|}\). The following statements are equivalent.

(i) There exist functions \((f, h) \in \overline{B}_L(\mathbb{R}^n)\) such that
\[
f_i = f(x_i), \ g_i \in \partial f(x_i), \quad h_i = h(x_i), \ s_i \in \partial h(x_i),
\]
\[
Ls_i - g_i \in \partial (Lh - f)(x_i).
\]

(ii) For all \(i, j \in I\) such that \(i \neq j\) we have
\[
f_i - f_j - \langle g_j, x_i - x_j \rangle \geq 0,
\]
\[
(Lh_i - f_i) - (Lh_j - f_j) - \langle Ls_i - g_j, x_i - x_j \rangle \geq 0. \tag{22}
\]

**Proof.** (i) \(\implies\) (ii) follows immediately from the definition of a subgradient applied to convex functions \(f\) and \(Lh - f\). Now, assume that (ii) holds. By the specialization (18) of Theorem 3, conditions (ii) imply that there exist two convex functions \(f, d : \mathbb{R}^n \to \mathbb{R}\) such that
\[
f_i = f(x_i), \ g_i \in \partial f(x_i), \quad Lh_i - f_i = d(x_i), \quad Ls_i - g_i \in \partial d(x_i).
\]
Now, defining the convex function \(h = (f + d)/L\), we have that \(d = Lh - f\), hence \(Ls_i - g_i \in \partial (Lh - f)(x_i)\). We also get
\[
h_i = h(x_i), \ s_i \in \partial h(x_i)
\]
where we used the fact that \(Ls_i \in \partial f(x_i) + \partial d(x_i) \subset \partial (f + d)(x_i) = L\partial h(x_i)\) (see [32, Thm 23.8] for the subdifferential of a sum of convex functions). Hence (i) holds. ■

**Corollary 2 (Interpolation conditions for (PEP)).** Let \(I\) be a finite index set and \(\{(x_i, f_i, g_i, h_i, s_i)\}_{i \in I} \in (\mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n)^{|I|}\). Assume that \(x_i \neq x_j\) for every \(i \neq j \in I\). The following statements are equivalent.
(i) There exist functions \((f, h) \in \mathcal{B}_L(\mathbb{R}^n)\) such that
\[
\begin{align*}
  f_i &= f(x_i), \quad g_i = \nabla f(x_i), \\
  h_i &= h(x_i), \quad s_i = \nabla h(x_i).
\end{align*}
\]

(ii) For all \(i, j \in I\) such that \(i \neq j\) we have
\[
(f_i - f_j - \langle g_j, x_i - x_j \rangle) > 0, \quad (Lh_i - f_i) - (Lh_j - f_j) - \langle Ls_j - g_j, x_i - x_j \rangle > 0.
\]

Proof. Note that since \(x_i \neq x_j\) for every \(i \neq j\), interpolation conditions of Proposition 3 reduce to requiring the strict inequality in (19) for every \(i \neq j\). As before, define \(d := Lh - f\). Then since \((f, h) \in \mathcal{B}_L(\mathbb{R}^n)\) the functions \(f\) and \(d\) are differentiable strictly convex, hence (i) \(\implies\) (ii) follows simply from strict convexity of these functions.

Conversely, assume (ii). By using Proposition 3 again, we can interpolate differentiable strictly convex functions \(f\) and \(d\) and recover \(h\) with \(h = (f + d)/L\), thus we have naturally \(Lh - f\) convex. The function \(h\) is thus also differentiable and strictly convex. Moreover, it can be seen from the proof of Proposition 3 that the interpolating functions can actually be chosen strongly convex, hence with this choice the well-posedness condition Assumption 1(iii) holds, and we can conclude that \((f, h) \in \mathcal{B}_L(\mathbb{R}^n)\).

4.3. Semidefinite reformulations. Now that we established the interpolation conditions for (PEP) and (PEP), we may use them to obtain semidefinite performance estimation formulations as in \([16, 34]\). This is made possible by observing that interpolation conditions (22)-(23) are quadratic inequalities in the problem variables.

Let \(\{(x_i, f_i, g_i, h_i, s_i)\}_{i \in I}\) be a feasible point of one of the PEPs in dimension \(n\). We write \(G \in \mathcal{S}_{3(N+2)}\) the Gram matrix that contains all dot products between \(x_i, g_i, s_i\) for \(i \in I\), with
\[
G = \begin{pmatrix}
G_{xx} & G_{gx} & G_{gs} \\
G_{gx}^\top & G_{gg} & G_{gs} \\
G_{gs}^\top & G_{gs} & G_{ss}
\end{pmatrix} \succeq 0
\]
whose size is independent of the dimension \(n\), where the blocks are defined as
\[
G_{ij}^{xx} = \langle x_i, x_j \rangle, \quad G_{ij}^{gx} = \langle g_i, x_j \rangle, \quad G_{ij}^{gs} = \langle g_i, s_j \rangle, \quad G_{ij}^{gg} = \langle g_i, g_j \rangle, \quad G_{ij}^{gs} = \langle s_i, x_j \rangle, \quad G_{ij}^{ss} = \langle s_i, s_j \rangle, \quad i, j \in I.
\]
Write also
\[
F = (f_0, \ldots, f_N, f_*), \quad H = (h_0, \ldots, h_N, h_*) \in \mathbb{R}^{N+2},
\]
the vectors representing the function values of \(f, h\) at the iterates. We now observe that all the constraints of (PEP) and (PEP) can be expressed using only \(G, F\) and \(H\).

For instance, interpolation conditions (22) for \(\overline{\mathcal{B}}_L(\mathbb{R}^n)\) rewrite for all \(i, j \in I\) as
\[
(Lh_i - f_i) - (Lh_j - f_j) - L(G^{sx}_{ji} - G^{sx}_{jj}) + G^{gx}_{ji} - G^{gx}_{jj} \geq 0, \quad (Lh_i - f_i) - (Lh_j - f_j) - L(G^{sx}_{ji} - G^{sx}_{jj}) + G^{gx}_{ji} - G^{gx}_{jj} \geq 0.
\]
This allows to reformulate the relaxation (PEP) as a semidefinite program, written
\[
\begin{align*}
\text{maximize} & \quad f_N - f_* \\
\text{subject to} & \quad f_i - f_j - G^{gx}_{ji} + G^{gx}_{jj} \geq 0, \\
& \quad (Lh_i - f_i) - (Lh_j - f_j) - L(G^{sx}_{ji} - G^{sx}_{jj}) + G^{gx}_{ji} - G^{gx}_{jj} \geq 0 \quad \text{for } i, j \in I, \\
& \quad G_{gg}^{ss} = 0, \\
& \quad G_{i+1,j}^{sx} = G_{ij}^{sx} - \lambda G_{ij}^{gx} \quad \text{for } i \in \{0 \ldots N - 1\}, j \in I, \\
& \quad h_s - h_0 - G^{ss}_{0s} + G^{sx}_{00} = 1, \\
& \quad G \succeq 0,
\end{align*}
\]
in the variables \(G \in \mathcal{S}_{3(N+2)}\) and \(F, H \in \mathbb{R}^{N+2}\).
Any feasible point of \((\text{PEP})\) can be cast into an admissible point of \((\text{sdp-PEP})\) by computing the semidefinite Gram matrix \(G\). Conversely, if \(G,F,H\) is an admissible point of \((\text{sdp-PEP})\), then the vectors \(\{(x_i,g_i,s_i)\}_{i \in I}\) can be recovered by performing, for instance, Cholesky decomposition of \(G\). Note that we expressed the algorithmic constraint \(s_{i+1} = s_i - \lambda g_i\) only through scalar products with the \(x_i\)’s in the SDP, since only the projection of the gradients on \(\text{Span}\{\{x_i\}_{i \in I}\}\) is relevant in the PEPs. Because interpolation conditions from Corollary 1 are necessary and sufficient, we conclude that the problems are equivalent, that is

\[
\text{val}(\text{sdp-PEP}) = \text{val}(\text{PEP}).
\]

The rank of \(G\) determines the dimension of the interpolated problem. If we look instead for a solution that has a given dimension \(n\), this would mean imposing a nonconvex rank constraint on \(G\). Our formulation, on the other side, is convex and finds the best convergence bound that is dimension-independent, which is an usual requirement for large-scale settings. Since \(G\) has size \(3(N+2)\), the dimension of the worst-case functions will be at most \(3(N+2)\).

In the same way, the value of \((\text{PEP})\) can be computed as

\[
\text{maximize } f_N - f_s
\]

subject to

\[
\begin{align*}
&f_i - f_j - G_{ij}^{g^x} + G_{jj}^{g^x} > 0, \\
&(Lh_i - f_i) - (Lh_j - f_j) - L(G_{jj}^{g^x} - G_{ij}^{g^x}) + G_{ji}^{g^x} - G_{jj}^{g^x} > 0 \quad \text{for } i \neq j \in I, \\
&G_{ii}^{g^x} = 0, \\
&G_{i+1,j}^{g^x} = G_{ij}^{g^x} - \lambda G_{ij}^{g^x} \quad \text{for } i \in \{0 \ldots N-1\}, j \in I, \\
&h_s - h_0 - G_{0s}^{g^x} + G_{00}^{g^x} = 1, \\
&G_{ii}^{g^x} + G_{jj}^{g^x} - 2G_{ij}^{g^x} > 0 \quad \text{for } i \neq j \in I, \\
&G \succeq 0,
\end{align*}
\]

in the variables \(G \in S_{3(N+2)}\) and \(F,H \in \mathbb{R}^{N+2}\), where we used interpolation conditions for \(B_L(\mathbb{R}^n)\) from Corollary 2, since all points \(\{x_i\}_{i \in I}\) are constrained to be distinct. Therefore, as above we infer that

\[
\text{val}(\text{sdp-PEP}) = \text{val}(\text{PEP}).
\]

Recalling the hierarchy between the problems, we thus have

\[
\text{val}(\text{sdp-PEP}) \leq \text{val}(\text{PEP}) \leq \text{val}(\text{sdp-PEP}).
\]

By comparing the two semidefinite programs stated above, one can notice that the only difference is that \((\text{sdp-PEP})\) imposes some inequalities of \((\text{sdp-PEP})\) to be strict. In the next section, we use topological arguments to prove that the values of the two problems are actually equal. In fact, strict inequalities have little meaning in numerical optimization (the value of \((\text{sdp-PEP})\) is actually a supremum and not a maximum); in our experiments, we will focus on \((\text{sdp-PEP})\) as solvers usually admit only closed feasible sets.

4.4. Tightness of the approach: nonsmooth limit behaviors. We are now ready to prove the main result of this section.

**Theorem 4.** The value of the performance estimation problem \((\text{PEP})\) for NoLips is equal to the value of the nonsmooth relaxation \((\text{PEP})\), which can be computed by solving the semidefinite program \((\text{sdp-PEP})\).

**Proof.** We will show that the closure of the feasible set of \((\text{sdp-PEP})\) is the feasible set of \((\text{sdp-PEP})\). We first need to prove that the strengthened problem \((\text{PEP})\) is feasible, by finding an instance of NoLips where \(f\) and \(Lh - f\) are strictly convex and such that all iterates are distinct. It suffices for instance to consider two unidimensional quadratic functions. Define \(f,h : \mathbb{R} \to \mathbb{R}\) with

\[
f(x) = \frac{\alpha}{2}x^2, \quad h(x) = \frac{1}{2}x^2 \quad \text{where } \alpha = \min \left( \frac{1}{2\lambda}, \frac{L}{2} \right).
\]

20
Then $f$ is strictly convex and so is $Lh - f = \frac{L-\alpha}{2}x^2$ since $L - \alpha \geq \frac{L}{2} > 0$. The optimum is $x_\ast = 0$. Choose $x_0 = \sqrt{2}$ for which we have $D_h(x_\ast, x_0) = x_0^2/2 = 1$. Then, Algorithm 1 is equivalent to gradient descent and the iterates satisfy

$$x_N = (1 - \lambda \alpha)^N x_0.$$ 

Since $\alpha \lambda \leq 1/2 < 1$, all the iterates are distinct and therefore we constructed a feasible point of (PEP). Let us therefore write $(G, F, H)$ a corresponding feasible point of (sdp-PEP), and $(\bar{G}, \bar{F}, \bar{H})$ a feasible point of (sdp-PEP). Define the sequence $\{(G^k, F^k, H^k)\}_{k \geq 1}$ as

$$G^k = \frac{1}{k} G + (1 - \frac{1}{k}) \bar{G},$$
$$F^k = \frac{1}{k} F + (1 - \frac{1}{k}) \bar{F},$$
$$H^k = \frac{1}{k} H + (1 - \frac{1}{k}) \bar{H}.$$ 

Then, for every $k \geq 1$, $(G^k, F^k, H^k)$ is still an feasible point of (sdp-PEP), because of convexity of the constraints and the fact that adding a strict inequality to a weak inequality gives a strict inequality. Moreover, the sequence converges to the point $(\bar{G}, \bar{F}, \bar{H})$ when $k \to +\infty$.

Hence we proved that for any feasible point of (sdp-PEP), there is a sequence of admissible points of (sdp-PEP) that converge to it. Since the objective is linear in the vector $F$ therefore continuous, we deduce that the two problems have the same value:

$$\text{val}(\text{sdp-PEP}) = \text{val}(\text{sdp-PEP}),$$

which means that $\text{val}(\text{PEP}) = \text{val}(\text{sdp-PEP})$. Since $\text{val}(\text{PEP})$ lies in between these two values, we conclude that they are all equal.

Theorem 4 states that the value of the original problem (PEP) can be computed numerically with a semi-definite solver applied to (sdp-PEP). The result itself also helps us gain some theoretical insight: it tells us that the worst case for NoLips might be reached as $(f, h)$ approach possibly pathological limiting nonsmooth functions in $B_L(\mathbb{R}^n)$.

Observe also that we focused on presenting the PEP for the class $B_L(\mathbb{R}^n)$ to avoid technicalities related to the domain of definition. However, we show in Appendix A that the exact same problem (sdp-PEP) also solves the performance estimation problem for NoLips on the general class $B_L(C)$, for any closed convex set $C$.

4.5. Numerical evidences and computer-assisted proofs. We now provide several applications of the performance estimation framework that we developed for Bregman methods.

4.5.1. Solving (PEP) for finding the exact worst-case convergence rate of NoLips. We first start by the most direct application, that is finding exact worst-case performance of NoLips. Theorem 4 states that it can be computed by solving the semidefinite program (sdp-PEP). The link to the MATLAB implementation is provided in Section 5.

To simplify our setting, note that we can assume without loss of generality that the $h$-smoothness constant $L$ is 1, since we can replace $h$ by a scaled version $Lh$. Recall that we know from Theorem 1 that

$$\text{val}(\text{PEP}) \leq \frac{1}{\lambda N}.$$ 

Table 1 shows the result of solving (sdp-PEP) for several values of $N$ up to 100, for a step size $\lambda = 1$. We observe that with high precision, $\text{val}(\text{sdp-PEP})$ is equal to the theoretical bound $1/(\lambda N)$. 

---

Table 1 shows the result of solving (sdp-PEP) for several values of $N$ up to 100, for a step size $\lambda = 1$. We observe that with high precision, $\text{val}(\text{sdp-PEP})$ is equal to the theoretical bound $1/(\lambda N)$. 

---
The proof relies on the fact that, since $\lambda > 1/L$, $\text{val}(\text{PEP}) = +\infty$, hence Algorithm 1 does not converge in general with these step size values. This suggests that the maximal step size value allowed for NoLips is indeed $1/L$, unlike the Euclidean setting where gradient descent can be applied with a step size that goes up to $2/L$.

While results above suggest that $1/(\lambda N)$ is the exact worst-case rate of NoLips, they provide only numerical evidence. We can however use them to deduce formal guarantees, both for proving an upper bound and a lower bound.

**Upper bound guarantee through duality.** As noticed in previous work on PEPs [16, 33], solving the dual of (sdp-PEP) can be used to deduce a proof. Indeed, the dual solution gives a combination of the constraints that, when transposed to analytical form, leads to a formal guarantee. This provides the following proof for the $O(1/k)$ convergence rate of Theorem 1.

**Proof of Theorem 1.** The proof relies on the fact that, since $Lh - f$ is convex we have that $1/\lambda h - f$ is convex for any $\lambda \in (0, 1/L]$, and only consists in performing the following weighted sum of inequalities:

- convexity of $f$, between $x_*$ and $x_i$ ($i = 0, \ldots, k$) with weights $\gamma_{i,i} = 1/k$:

  $$f(x_*) \geq f(x_i) + \langle \nabla f(x_i), x_* - x_i \rangle,$$

- convexity of $f$, between $x_i$ and $x_{i+1}$ ($i = 0, \ldots, k - 1$) with weights $\gamma_{i,i+1} = 1/k$:

  $$f(x_i) \geq f(x_{i+1}) + \langle \nabla f(x_{i+1}), x_i - x_{i+1} \rangle,$$

- convexity of $1/\lambda h - f$, between $x_*$ and $x_k$ with weight $\mu_{*,k} = 1/k$:

  $$\frac{1}{\lambda} h(x_*) - f(x_*) \geq \frac{1}{\lambda} h(x_k) - f(x_k) + \langle \frac{1}{\lambda} \nabla h(x_k) - \nabla f(x_k), x_* - x_k \rangle,$$

- convexity of $1/\lambda h - f$, between $x_{i+1}$ and $x_i$ ($i = 0, \ldots, k - 1$) with weight $\mu_{i+1,i} = \frac{i+1}{k}$:

  $$\frac{1}{\lambda} h(x_{i+1}) - f(x_{i+1}) \geq \frac{1}{\lambda} h(x_i) - f(x_i) + \langle \frac{1}{\lambda} \nabla h(x_i) - \nabla f(x_i), x_{i+1} - x_i \rangle,$$

- convexity of $1/\lambda h - f$, between $x_i$ and $x_{i+1}$ ($i = 0, \ldots, k - 1$) with weight $\mu_{i,i+1} = \frac{i}{k}$:

  $$\frac{1}{\lambda} h(x_i) - f(x_i) \geq \frac{1}{\lambda} h(x_{i+1}) - f(x_{i+1}) + \langle \frac{1}{\lambda} \nabla h(x_{i+1}) - \nabla f(x_{i+1}), x_i - x_{i+1} \rangle.$$

**Table 1.** Numerical value of the performance estimation problem (PEP) with $\lambda = 1$, $L = 1$. *Rel. error* denotes the relative error between val(PEP) and the theoretical bound of $1/N$ given by Theorem 1. *Primal feasibility* corresponds to the maximal absolute value of constraint violation returned by the MOSEK solver.

| N  | val(PEP) | Rel. error | Primal feasibility |
|----|----------|------------|--------------------|
| 1  | 1.000    | 1.8e-11    | 4.3e-10            |
| 2  | 0.500    | 1.8e-8     | 2.8e-9             |
| 3  | 0.333    | 1.8e-8     | 2.8e-9             |
| 4  | 0.250    | 4.9e-8     | 2.3e-8             |
| 5  | 0.200    | 1.8e-10    | 6.4e-11            |
| 10 | 0.100    | 6.4e-11    | 1.3e-11            |
| 20 | 0.050    | 1.1e-8     | 1.9e-10            |
| 50 | 0.020    | 6.5e-6     | 5.0e-7             |
| 100| 0.01     | 7.2e-5     | 1.6e-6             |

**Other values of $\lambda$.** One can wonder how the numerical value evolves when we vary the step size $\lambda$. The experimental observations are the following:

- For any $\lambda \in (0, 1/L]$, val(PEP) is exactly equal to the theoretical bound $1/(\lambda N)$.
- For any $\lambda > 1/L$, val(PEP) = $+\infty$, hence Algorithm 1 does not converge in general with these step size values. This suggests that the maximal step size value allowed for NoLips is indeed $1/L$, unlike the Euclidean setting where gradient descent can be applied with a step size that goes up to $2/L$. 
The weighted sum is written as
\[
0 \geq \sum_{i=0}^{k} \gamma_{*,i} [f(x_i) - f(x_*) + \langle \nabla f(x_i), x_* - x_i \rangle
\]
\[
+ \sum_{i=0}^{k-1} \gamma_{i,i+1} [f(x_{i+1}) - f(x_i) + \langle \nabla f(x_{i+1}), x_i - x_{i+1} \rangle]
\]
\[
+ \mu_{*,k} \left[ \frac{1}{\lambda} h(x_k) - f(x_k) - \left( \frac{1}{\lambda} h(x_*) - f(x_*) \right) + \langle \frac{1}{\lambda} \nabla h(x_k) - \nabla f(x_k), x_* - x_k \rangle \right]
\]
\[
+ \sum_{i=0}^{k-1} \mu_{i,i+1} \left[ \frac{1}{\lambda} h(x_i) - f(x_i) - \left( \frac{1}{\lambda} h(x_{i+1}) - f(x_{i+1}) \right) + \langle \frac{1}{\lambda} \nabla h(x_i) - \nabla f(x_i), x_{i+1} - x_i \rangle \right]
\]
\[
+ \sum_{i=0}^{k-1} \mu_{i,i+1} \left[ \frac{1}{\lambda} h(x_{i+1}) - f(x_{i+1}) - \left( \frac{1}{\lambda} h(x_i) - f(x_i) \right) + \langle \frac{1}{\lambda} \nabla h(x_{i+1}) - \nabla f(x_{i+1}), x_i - x_{i+1} \rangle \right],
\]
By substitution of \( \nabla h(x_{i+1}) = \nabla h(x_i) - \lambda \nabla f(x_i) \) \((i = 0, \ldots, k - 1)\), one can reformulate the weighted sum exactly as (i.e., there is no residual):
\[
0 \geq f(x_k) - f(x_*) - \frac{h(x_*) - h(x_0) - \langle \nabla h(x_0), x_* - x_0 \rangle}{\lambda k},
\]
yielding the desired result. ■

**Lower bound through worst-case functions.** As (PEP) computes the exact worst-case performance of NoLips, experiments above suggest that \( 1/(\lambda N) \) is also a lower bound, meaning that for every \( \epsilon > 0 \), there exist functions \((f, h) \in B_L\) such that the iterates of NoLips satisfy
\[
f(x_N) - f_* \geq \frac{D_h(x_*, x_0)}{\lambda N} - \epsilon.
\]
We detail here how such functions can be constructed from the solution of (sdp-PEP). The numerical solver allow us to find a maximizer \( \overline{G}, \overline{T}, \overline{P} \) (recall that only the relaxed problem has a maximizer as the feasible set is closed), and by factorizing the matrix \( G \) as \( \overline{P}^T P \) we can thus recover the corresponding discrete representation \( \{\overline{x}_i, \overline{y}_i, \overline{T}_i, \overline{h}_i, \overline{s}_i\}_{i \in I} \). This discretization can in turn be interpolated to get the corresponding functions \((\overline{f}, \overline{h}) \in B_{\overline{L}}\). There are multiple ways to perform this interpolation; see [34, Thm. 1] for a constructive approach.

Recall that since functions \((\overline{f}, \overline{h})\) are solution to (PEP), they belong to \( B_{\overline{L}} \) and might thus form a pathological nonsmooth limiting worst-case. They can be approached by valid instances \((f_\mu, h_\mu) \in B_L\) by performing for instance smoothing through Moreau envelopes (as in Section 3.2) and adding a small quadratic to \( h \) to make it strictly convex.

There are however many possible maximizers of (sdp-PEP). If we seek a low-dimensional example that may be easily interpretable, we can search for a maximizer such that the Gram matrix \( G \) has minimal rank. Using rank minimization heuristics, we were able to find one-dimensional worst-case functions. Fix a number of iterations \( N \geq 1 \), assume \( \lambda = 1/L = 1 \) and define \( \overline{f}, \overline{h} : \mathbb{R} \to \mathbb{R} \) as
\[
\overline{f}(x) = |x - 1|,
\]
\[
\overline{h}(x) = \overline{f}(x) + \max(-Nx, 0).
\]
Then clearly \((\overline{f}, \overline{h}) \in B_{\overline{L}}(\mathbb{R})\). Figure 2 shows the functions \((\overline{f}, \overline{h})\) as well as their smoothed versions \((f_\mu, h_\mu) \in B_L(\mathbb{R})\). Note that the pathological behavior also reflects in the iterates: in the limiting instance, all iterates \( \overline{x}_0, \ldots, \overline{x}_N \) are equal. In the smoothed version, iterates are distinct (since \( h_\mu \) is strictly convex), but they get closer and closer as the smoothing parameter \( \mu \) goes to 0.

The smoothed function \( f_\mu \) is a Huber function, which is also the worst-case instance for Euclidean gradient descent on \( L \)-smooth functions described in [34]. This analysis could be formalized to prove the \( 1/k \)
In the same way as before, the formal guarantee has been obtained by examining the dual of the corresponding PEP. The proof relies on the fact that \( f \) is convex and \( h \) is smooth and strongly convex. Let \( \lambda \) denote the step size. By adapting (PEP), we get the following new convergence result for NoLips.

**Proposition 4** (NoLips convergence rate, take II). Let \( L > 0 \), \( C \) be a nonempty closed convex subset of \( \mathbb{R}^n \) and \( (f, h) \in B_L(C) \) functions admissible for NoLips. Then the sequence \( \{x_k\}_{k \geq 0} \) generated by Algorithm 1 with constant step size \( \lambda \in (0, 1/L] \) satisfies for \( k \geq 2 \)

\[
\min_{1 \leq i \leq k} D_h(x_{i-1}, x_i) \leq \frac{2D_h(x_0, x_0)}{k(k-1)}
\]

where \( x_* \in \text{argmin}_C f \cap \text{dom} h \).

**Proof.** In the same way as before, the formal guarantee has been obtained by examining the dual of the corresponding PEP. The proof relies on the fact that \( \frac{1}{\lambda} h - f \) is convex for any \( \lambda \in (0, \frac{1}{L}] \), and only consists in performing the following weighted sum of inequalities:

- **convexity of** \( f \), **between** \( x_* \) **and** \( x_i \) (\( i = 0, \ldots, k \)) **with weights** \( \gamma_{*,i} = \frac{2\lambda}{k(k-1)} \):
  \[
  f(x_*) \geq f(x_i) + \langle \nabla f(x_i), x_* - x_i \rangle,
  \]

- **optimality of** \( x_* \) **for each** \( x_k \) **with weight** \( \gamma_{k,*} = \frac{2\lambda}{k-1} \):
  \[
  f(x_k) \geq f(x_*),
  \]

- **convexity of** \( \frac{1}{\lambda} h - f \), **between** \( x_* \) **and** \( x_k \) **with weight** \( \mu_{*,k} = \frac{2\lambda}{k(k-1)} \):
  \[
  \frac{1}{\lambda} h(x_*) - f(x_*) \geq \frac{1}{\lambda} h(x_k) - f(x_k) + \langle \frac{1}{\lambda} \nabla h(x_k) - \nabla f(x_k), x_* - x_k \rangle,
  \]

- **convexity of** \( \frac{1}{\lambda} h - f \), **between** \( x_{i+1} \) **and** \( x_i \) (\( i = 0, \ldots, k - 1 \)) **with weight** \( \mu_{i+1,i} = \frac{2\lambda(i+1)}{k(k-1)} \):
  \[
  \frac{1}{\lambda} h(x_{i+1}) - f(x_{i+1}) \geq \frac{1}{\lambda} h(x_i) - f(x_i) + \langle \frac{1}{\lambda} \nabla h(x_i) - \nabla f(x_i), x_{i+1} - x_i \rangle,
  \]
proposed by Auslender and Teboulle [1], a.k.a. the Improved Interior Gradient Algorithm (IGA). We recall
For instance, we can also solve the performance estimation problem for the inertial Bregman algorithm
Beyond NoLips: inertial Bregman algorithms.
yielding the desired result.
∇ 
\sum_{i=0}^{k} \gamma_{i} [f(x_{i}) - f(x_{*}) + \langle \nabla f(x_{i}), x_{*} - x_{i} \rangle] 
+ \gamma_{k} [f(x_{*}) - f(x_{k})] 
+ \mu_{k} [\frac{1}{k} h(x_{k}) - f(x_{k}) - (\frac{1}{k} h(x_{*}) - f(x_{*})) + \langle \frac{1}{k} \nabla h(x_{k}) - \nabla f(x_{k}), x_{*} - x_{k} \rangle] 
+ \sum_{i=0}^{k-1} \mu_{i+1} [\frac{1}{i+1} h(x_{i}) - f(x_{i}) - (\frac{1}{i+1} h(x_{i+1}) - f(x_{i+1})) + \langle \frac{1}{i+1} \nabla h(x_{i}) - \nabla f(x_{i}), x_{i+1} - x_{i} \rangle] 
+ \sum_{i=1}^{k} \tau_{i} [\min_{1 \leq j \leq k} \{ D_{h}(x_{j-1}, x_{j}) \} - (h(x_{i-1}) - h(x_{i}) - \langle \nabla h(x_{i}), x_{i-1} - x_{i} \rangle)].

By substitution of \( \nabla h(x_{i+1}) = \nabla h(x_{i}) - \lambda \nabla f(x_{i}) \) (i = 0, ..., k - 1), one can reformulate the weighted sum exactly as (i.e., there is no residual):
0 \geq \min_{1 \leq j \leq k} \{ D_{h}(x_{j-1}, x_{j}) \} - 2 \frac{h(x_{*}) - h(x_{0}) - \langle \nabla h(x_{0}), x_{*} - x_{0} \rangle}{k(k-1)},
yielding the desired result. ■

4.5.3. Beyond NoLips: inertial Bregman algorithms. Our approach is not limited to the NoLips algorithm. For instance, we can also solve the performance estimation problem for the inertial Bregman algorithm proposed by Auslender and Teboulle [1], a.k.a. the Improved Interior Gradient Algorithm (IGA). We recall its simplified formulation in Algorithm 2, in the case where there are no affine constraints.
Algorithm 2 Improved Interior Gradient Algorithm (IGA) [1]

**Input:** Functions \( f, h \), initial point \( x_0 \in \text{int dom } h \), step size \( \lambda \).

Set \( z_0 = x_0 \) and \( t_0 = 1 \).

for \( k = 0, 1, \ldots \) do

\[
y_k = (1 - \frac{1}{t_k}) x_k + \frac{1}{t_k} z_k
\]

\[
z_{k+1} = \arg\min \{ \langle \nabla f(y_k), u - y_k \rangle + \frac{1}{t_k} D_h(u, z_k) \mid u \in \mathbb{R}^n \}
\]

\[
x_{k+1} = (1 - \frac{1}{t_k}) x_k + \frac{1}{t_k} z_{k+1}
\]

\[
t_{k+1} = \frac{1 + \sqrt{1 + 4 t_k^2}}{2}
\]

end for

In the setting where \( f \) has \( \bar{L} \)-Lipschitz continuous gradients and \( h \) is a \( \sigma \)-strongly convex Legendre function, IGA with step size \( \lambda = \frac{\sigma}{\bar{L}} \) enjoys the following convergence rate [1, Thm. 5.2]:

\[
f(x_N) - f^* \leq \frac{4\bar{L}}{\sigma N^2} (D_h(x^*, x_0) + f(x_0) - f^*)
\]

(24)

Our PEP framework can be also applied to this algorithm, in order to find the smallest value of \( \theta(N, \bar{L}, \sigma, \lambda) \) which satisfies

\[
f(x_N) - f^* \leq \theta(N, \bar{L}, \sigma, \lambda) (D_h(x^*, x_0) + f(x_0) - f^*)
\]

for every instance of IGA with the supplementary assumptions made above. In this case, we use the standard interpolation conditions of Theorem 3 for \( L \)-smooth and strongly convex functions. Results are shown in Figure 3. The exact numerical worst-case performance of IGA is slightly below the theoretical bound above, since the proof in [1] makes some approximations.

**IGA in the general \( h \)-smooth case.** We pointed out in Section 2 that the setting in which \( f \) is \( \bar{L} \)-smooth and \( h \) is \( \sigma \)-strongly convex is a particular case of \( h \)-smoothness with constant \( L = \bar{L}/\sigma \). The natural question that was also raised in [37, Section 6] is therefore: does IGA converge for the general class \( \mathcal{B}_L(C) \)? Solving the corresponding PEP yields the following results. For Algorithm 2 with the setting that \((f, h) \in \mathcal{B}_L(C)\) and several choices of step size in \((0, 1/L]\), the solver states the value of the corresponding performance estimation problem is unbounded, i.e., there does not exist any \( \theta \) such that the bound (24) holds for every instance \((f, h) \in \mathcal{B}_L\). Of course, this constitutes numerical evidence and not a formal proof. Nonetheless, due to the tightness result of Theorem 4, there are strong reasons to conjecture that IGA indeed does not converge in the general \( h \)-smooth setting.

### 4.5.4. From worst-case functions for NoLips to a lower bound for general Bregman methods.

We briefly explain how, with the PEP methodology, the worst case functions from Section 3.2 were discovered.

We described in Section 4.5.1 how a one-dimensional worst-case instance \((\bar{f}, \bar{h})\) for NoLips has been discovered from low-rank solutions of \((\text{sdp-PEP})\). However, this instance may not be difficult enough for a more generic Bregman algorithm that can use arbitrary linear combinations of gradients (as in Assumption 2, our definition of the Bregman gradient algorithm), and thus cannot be used to prove a general lower bound.

Our objective now is to find worst-case instances that are difficult for any Bregman gradient algorithm. A desirable property would be that these instances allow to explore only one dimension per oracle call, so that the function hides information in the unexplored dimensions. This similar in spirit with the so-called “worst function in the world” of Nesterov [30]. In order to achieve this goal, we propose to search for functions \( f \) for which all gradients \( \nabla f(x_i) \) would be orthogonal, guaranteeing that one new dimension is explored at
each step. Note that a similar approach has been used in some previous work on PEPs to find lower bounds or optimal methods e.g. in [13, 15]. This amounts to add some orthogonality constraints to (PEP) and solve

\[
\begin{align*}
\text{maximize} & \quad (f(x_N) - f(x_\ast)) / D_h(x_\ast, x_0) \\
\text{subject to} & \quad (f, h) \in B_L(\mathbb{R}^n), \\
& \quad x_\ast \text{ is a minimizer of } f, \\
& \quad x_1, \ldots, x_N \text{ are generated from } x_0 \text{ by Algorithm 1 with step size } \lambda, \\
& \quad \langle \nabla f(x_i), \nabla f(x_j) \rangle = 0 \text{ for } i \neq j \in I, \\
\end{align*}
\]

(PEP-orth)

in the variables \( f, h, x_0, \ldots, x_N, x_\ast, n \).

In the same spirit as before, we were able to find a dimension-\( N \) solution of (PEP-orth). This allows us to interpolate the following worst-case pathological instance in dimension \( N \):

\[
\begin{align*}
\tilde{f}(x) &= \|x - (1, \ldots, 1)\|_\infty, \\
\tilde{h}(x) &= f(x) + \sum_{i=2}^{N} \max(-x^{(i)}, 0).
\end{align*}
\]

Again, these are nonsmooth functions and do not form valid instances of NoLips. However, they can be approached by a sequence of such functions, for instance by applying smoothing with the Moreau envelope, and adding a small quadratic term to make \( h \) strictly convex. Along with a few tweaks, this is how we found the example that was used to prove the general lower bound for \( B_L \) in Section 3.2.

5. Conclusion

Our paper has two main contributions: proving optimality of NoLips for the general \( h \)-smooth setting, and developing numerical performance estimation techniques for Bregman gradient algorithms. We presented the performance estimation problem on the basic NoLips algorithm for simplicity, but our approach can be applied to different settings and various algorithms involving Bregman distances. We provided several applications illustrating how the PEP methology is an efficient tool for conjecturing and analyzing the worst-case behavior of Bregman algorithms.

There is a fundamental concept linking the two parts of the paper, which is that of limiting nonsmooth pathological behavior. When looking for worst-case guarantees over a class of functions that is open such as the class of differentiable convex functions, the performance estimation problem is a supremum and the worst-case maximizing sequence might approach some function that is not in this class, e.g. one that is nonsmooth in our case. This idea, observed by analyzing the equivalence between (PEP) and the nonsmooth relaxation (PEP), was used in the proof of the lower bound in Section 3.2. Moreover, the worst-case sequence of functions was directly inspired by examining particular solutions of (PEP).

It is clear that additional assumptions on functions \( f \) and \( h \) are needed in order to prove better bounds or devise faster algorithms than NoLips. If the usual properties of \( L \)-smoothness and strong convexity are too restrictive and do not hold in many applications, the future challenge is to find weaker assumptions, that define a larger class of functions where improved rates can be obtained. One other possible approach would be to find algorithms that do not fit in Assumption 2, for instance by including second-order oracles of \( h \), in the case where \( h \) is simple enough.

Code. Experiments have been run in MATLAB, using the semidefinite solver MOSEK [26] as well as the modeling toolbox YALMIP [23]. The support for Bregman methods has been added to the Performance Estimation Toolbox (PESTO, [35]) for which we provide some examples. The code can be downloaded from https://github.com/RaduAlexandruDragomir/BregmanPerformanceEstimation
Acknowledgements. The authors would like to thank Dmitrii Ostrovskii for helpful discussions and Edouard Pauwels for useful comments. RD would like to acknowledge support from an AMX fellowship. AT acknowledges support from the European Research Council (grant SEQUOIA 724063). AA is at CNRS & département d’informatique, École normale supérieure, UMR CNRS 8548, 45 rue d’Ulm 75005 Paris, France, INRIA and PSL Research University. AA would like to acknowledge support from the ML & Optimisation joint research initiative with the fonds AXA pour la recherche and Kamet Ventures, as well as a Google focused award. JB acknowledges the support of ANR-3IA ANITI, ANR Chess, Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant numbers FA9550-19-1-7026, FA9550-18-1-0226.
APPENDIX A. EXTENSION OF PERFORMANCE ANALYSIS TO THE CASE WHERE \( C \) IS A GENERAL CLOSED CONVEX SUBSET OF \( \mathbb{R}^n \)

For simplicity of the presentation, we left out in Section 4 the case where the domain \( C \) is a proper subset of \( \mathbb{R}^n \). We show in this section that it actually corresponds to the same minimization problem (sdp-PEP).

Let us formulate the performance estimation problem for Algorithm 1 in the general case. Recall that we denote \( B_L \) the union of \( B_L(C) \) for all closed convex subsets of \( \mathbb{R}^n \) for every \( n \geq 1 \). The performance estimation problem writes

\[
\begin{align*}
\text{maximize} & \quad (f(x_N) - f(x_*))/D_h(x_*, x_0) \\
\text{subject to} & \quad (f, h) \in B_L, \\
& \quad x_\ast \text{ is a minimizer of } f \text{ on } \text{dom } h \text{ such that } x \in \text{dom } h, \\
& \quad x_1, \ldots, x_N \text{ are generated from } x_0 \text{ by Algorithm 1 with step size } \lambda,
\end{align*}
\]

(PEP-C)

in the variables \( f, h, x_0, \ldots, x_N, x_*, n \). Now, as (PEP-C) is a problem that includes (PEP) in the special case where \( C = \mathbb{R}^n \), its value is larger:

\[ \text{val}(\text{PEP}) \leq \text{val}(\text{PEP-C}) \]

We now proceed to show that \( \text{val}(\text{PEP-C}) \) is upper bounded by the same relaxation \( \text{val}(\text{PEP}) \), which will allow to conclude that the values are equal. We recall that the problem (PEP) can be written, using interpolation conditions of Corollary 1, as

\[
\begin{align*}
\text{maximize} & \quad f_N - f_* \\
\text{subject to} & \quad f_i - f_j - \langle g_j, x_i - x_j \rangle \geq 0, \\
& \quad (Lh_i - f_i) - (Lh_j - f_j) - \langle Ls_j - g_j, x_i - x_j \rangle \geq 0 \quad \text{for } i, j \in I, \\
& \quad g_* = 0, \\
& \quad s_{i+1} = s_i - \lambda g_i \quad \text{for } i \in \{1 \ldots N - 1\}, \\
& \quad h_* - h_0 - \langle s_0, x_* - x_0 \rangle = 1,
\end{align*}
\]

(PEP)

in the variables \( n, \{(x_i, f_i, g_i, h_i, s_i)\}_{i \in I} \). We show that every admissible point of (PEP-C) can be cast into an admissible point of (sdp-PEP). This actually amounts to show that, from the point of view of performance estimation, an instance \((f, h) \in B_L(C)\) is actually equivalent to some instance in \( B_L(\mathbb{R}^n) \).

Let \( f, h, x_0, \ldots, x_N, x_* \) be a feasible point of (PEP-C). We distinguish two cases.

Case 1: \( x_* \in \text{int } \text{dom } h \). This is the simplest case, as the necessary conditions are the same as in the situation where \( C = \mathbb{R}^n \). Indeed, then we have \( x_0, \ldots, x_N, x_* \in \text{int } \text{dom } h \), since \( x_0 \) is constrained to be in the interior and the next iterates are in \( \text{int } \text{dom } h \) by Assumption 1. Since \( f \) and \( h \) are differentiable on \( \text{int } \text{dom } h \), convexity of \( f \) and \( Lh - f \) imply that the first two constraints of (PEP) hold for all \( i, j \in I \). Finally, \( g_* = 0 \) follows from the fact that \( x_* \) minimizes \( f \) and that it lies on the interior of the domain.

Hence the discrete representation satisfies the constraints of (sdp-PEP).

Case 2: \( x_* \in \partial \text{dom } h \). In this case, \( f \) and \( h \) are not necessarily differentiable at \( x_* \), but are still differentiable still at \( x_0, \ldots, x_N \) for the same reasons. But we can still, with a small modification at \( x_* \), derive a discrete representation that fits the constraints of (PEP) and whose objective is the same. Indeed, define

\[
\begin{align*}
(g_i, f_i, s_i, h_i) &= (\nabla f(x_i), f(x_i), \nabla h(x_i), h(x_i)) \quad \text{for } i = 0 \ldots N \\
(g_*, f_*, s_*, h_*) &= (0, f(x_*), v, h(x_*))
\end{align*}
\]

where \( v \in \mathbb{R}^n \) is a vector that will be specified later. Then, for \( i \in I \) and \( j \in \{0 \ldots N\} \), convexity of \( f \) and \( Lh - f \) imply that the constraints

\[
\begin{align*}
f_i - f_j - \langle g_j, x_i - x_j \rangle &\geq 0 \\
(Lh_i - f_i) - (Lh_j - f_j) - \langle Ls_j - g_j, x_i - x_j \rangle &\geq 0
\end{align*}
\]
hold. It remains to verify them for \(i \in \{0 \ldots N\}\) and \(j = \ast\). The first one holds because \(x_{\ast}\) minimizes \(f\) on \(\text{dom} \ h\), so with \(g_{\ast} = 0\) we have \(f_i - f_{\ast} \geq 0\). We now show that the second one is satisfied, i.e. that we can choose \(v \in \mathbb{R}^n\) so that

\[
(Lh_i - f_i) - (Lh_{\ast} - f_{\ast}) - \langle Lv, x_i - x_{\ast}\rangle \geq 0 \quad \forall i \in \{0 \ldots N\}
\]

To this extend, we use the fact that \(x_{\ast} \in \partial \text{dom} \ h\) and that \(x_i \in \text{int} \text{ dom} \ h\) for \(i = 0 \ldots N\). This means that \(\{x_{\ast}\} \cap \text{int} \text{ dom} \ h = \emptyset\), and therefore by the hyperplane separation theorem [32, Thm 11.3], there exists a hyperplane that separates the convex sets \(\{x_{\ast}\}\) and \(\text{int} \text{ dom} \ h\) properly, meaning that there exists a vector \(u \in \mathbb{R}^n\) such that

\[
\langle x_i - x_{\ast}, u \rangle < 0 \quad \forall i \in \{0 \ldots N\}
\]

Denote now

\[
\alpha = \min_{i=0\ldots N} (Lh_i - f_i) - (Lh_{\ast} - f_{\ast})
\]

\[
\beta = \min_{i=0\ldots N} -\langle x_i - x_{\ast}, u \rangle > 0
\]

where \(\beta > 0\) because of the separation. Choose now \(s_{\ast} = v\) as \(v = \frac{|\alpha|}{L\beta} u\). Then we have

\[
(Lh_i - f_i) - (Lh_{\ast} - f_{\ast}) - \langle Ls_{\ast}, x_i - x_{\ast}\rangle \geq \alpha + L\frac{|\alpha|}{L\beta} \beta
\]

\[
\geq \alpha + |\alpha| \geq 0.
\]

This achieves to show that we built an instance \(\{(x_i, g_i, f_i, h_i, s_i)\}_{i \in I}\) that is admissible for (PEP).

To conclude, we proved that in both cases, an admissible point of (PEP-C) can be turned into an admissible point of (sdp-PEP) with the same objective value. Hence we have

\[
\text{val}(\text{PEP-C}) \leq \text{val}(\text{sdp-PEP}).
\]

Now, recalling that \(\text{val}(\text{PEP}) \leq \text{val}(\text{PEP-C})\) and that \(\text{val}(\text{sdp-PEP}) = \text{val}(\text{PEP})\) by Theorem 4, we get

\[
\text{val}(\text{PEP-C}) = \text{val}(\text{PEP}).
\]

In other words, solving the performance estimation problem (PEP-C) for functions with any closed convex domain is equivalent to solving the performance estimation problem (PEP) restricted to functions that have full domain.

REFERENCES

[1] Auslender, A., Teboulle, M.: Interior gradient and proximal methods for convex and conic optimization. SIAM Journal on Optimization 16(3), 697–725 (2006)
[2] Bauschke, H.H., Bolte, J., Chen, J., Teboulle, M., Wang, X.: On Linear Convergence of Non-Euclidean Gradient Methods without Strong Convexity and Lipschitz Gradient Continuity. Journal of Optimization Theory and Applications 182(3), 1068–1087 (2019)
[3] Bauschke, H.H., Bolte, J., Teboulle, M.: A Descent Lemma Beyond Lipschitz Gradient Continuity: First-Order Methods Revisited and Applications. Mathematics of Operations Research 42(2), 330–348 (2017)
[4] Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer Publishing Company, Inc. (2011)
[5] Beck, A., Teboulle, M.: Mirror descent and nonlinear projected subgradient methods for convex optimization. Operations Research Letters 31(3), 167–175 (2003)
[6] Beck, A., Teboulle, M.: A Fast Iterative Shrinkage-Thresholding Algorithm. SIAM Journal on Imaging Sciences 2(1), 183–202 (2009)
[7] Ben-tal, A., Margalit, T., Nemirovski, A.: The Ordered Subsets Mirror Descent Optimization Method with Applications to Tomography. SIAM Journal on Optimization 12(1), 79–108 (2001)
[8] Bolte, J., Sabach, S., Teboulle, M., Vaisbourd, Y.: First order methods beyond convexity and lipschitz gradient continuity with applications to quadratic inverse problems. SIAM Journal on Optimization 28(3), 2131–2151 (2018)
[9] Bubeck, S.: Introduction to online optimization. Lecture Notes (2011)
[10] Bui, M.N., Combettes, P.L.: Bregman Forward-Backward Operator Splitting. arXiv preprint arXiv:1908.03878 (2019)
[11] Censor, Y., Zenios, S.A.: Proximal minimization algorithm with D-functions. Journal of Optimization Theory and Applications 73(3), 451–464 (1992)
[12] Dragomir, R.A., D’Aspremont, A., Bolte, J.: Quartic First-Order Methods for Low Rank Minimization. arXiv preprint arXiv:1901.10791v2 (2019)
[13] Drori, Y.: The exact information-based complexity of smooth convex minimization. Journal of Complexity 39, 1–16 (2017)
[14] Drori, Y., Shamir, O.: The Complexity of Finding Stationary Points with Stochastic Gradient Descent. arXiv preprint arXiv:1910.01845 (2019)
[15] Drori, Y., Taylor, A.B.: Efficient first-order methods for convex minimization: a constructive approach. Mathematical Programming (2019)
[16] Drori, Y., Teboulle, M.: Performance of first-order methods for smooth convex minimization: A novel approach. Mathematical Programming 145(1-2), 451–482 (2014)
[17] Drori, Y., Teboulle, M.: An optimal variant of Kelley’s cutting-plane method. Mathematical Programming 160(1-2), 321–351 (2016)
[18] Eckstein, J.: Nonlinear Proximal Point Algorithms Using Bregman Functions, with Applications to Convex Programming. Mathematics of Operations Research 18(1), 202–226 (1993)
[19] Guzmán, C., Nemirovski, A.: On lower complexity bounds for large-scale smooth convex optimization. Journal of Complexity 31(1), 1–14 (2015)
[20] Hanzely, F., Richtarik, P., Xiao, L.: Accelerated Bregman proximal gradient methods for relatively smooth convex optimization. ArXiv preprint arXiv:1808.03045v1 (2018)
[21] Juditsky, A., Nemirovski, A.: First Order Methods for Nonsmooth Convex Large-Scale Optimization , I : General Purpose Methods. In: S.S. Wright, S. Nowozin, S. J. (eds.) Optimization for Machine Learning, pp. 121–147. MIT Press (2010)
[22] Kim, D., Fessler, J.A.: Optimized first-order methods for smooth convex minimization. Mathematical Programming 159(1-2), 81–107 (2016)
[23] Lofberg, J.: YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In: In Proceedings of the CACSD Conference (2004)
[24] Lu, H., Freund, R.M., Nesterov, Y.: Relatively-Smooth Convex Optimization by First-Order Methods, and Applications. SIAM Journal on Optimization 28(1), 333–354 (2018)
[25] Moreau, J.J.: Proximité et dualité dans un espace hilbertien. Bull. Soc. Math. Fr. 93(2), 273–299 (1965)
[26] Mosek, A.: The MOSEK optimization toolbox for MATLAB manual. Version 9.0. (2019). URL http://docs.mosek.com/9.0/toolbox/index.html
[27] Mukkamala, M.C., Ochs, P., Pock, T., Sabach, S.: Convex-Concave Backtracking for Inertial Bregman Proximal Gradient Algorithms in Non-Convex Optimization. arXiv preprint arXiv:1904.03537 (2019)
[28] Nemirovski, A., Yudin, D.B.: Problem complexity and method efficiency in optimization (1983)
[29] Nesterov, Y.: A Method of Solving A Convex Programming Problem With Convergence rate O(1/k^2) (1983)
[30] Nesterov, Y.: Introductory lectures on convex optimization: A basic course, 1 edn. Springer Publishing Company, Inc (2003)
[31] Nesterov, Y.: Implementable Tensor Methods in Unconstrained Convex Optimization. CORE Discussion Paper (2018)

[32] Rockafellar, R.T.: Convex analysis (1970)

[33] Taylor, A., Hendrickx, J., Glineur, F.: Exact worst-case performance of first-order methods for composite convex optimization. SIAM Journal on Optimization 27, 1600–1621 (2015)

[34] Taylor, A.B., Hendrickx, J.M., Glineur, F.: Smooth strongly convex interpolation and exact worst-case performance of first-order methods. Mathematical Programming 161(1-2), 307–345 (2017)

[35] Taylor, A.B., Hendrickx, J.M., Glineur, F.: Performance estimation toolbox (PESTO): Automated worst-case analysis of first-order optimization methods. In: IEEE 56th Annual Conference on Decision and Control (CDC 2017), pp. 1278–1283 (2018)

[36] Teboulle, M.: Entropic Proximal Mappings with Applications to Nonlinear Programming. Mathematics of Operations Research 17(3), 670–690 (1992)

[37] Teboulle, M.: A simplified view of first order methods for optimization. Mathematical Programming 170(1), 67–96 (2018)

[38] Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Review 38(1), 49–45 (1996)

[39] Walid, K., Bayen, A., Bartlett, P.L.: Accelerated Mirror Descent In Continuous and Discrete Time. In: Advances in Neural Information Processing Systems 28, pp. 2845—2853 (2015)

Université Toulouse I Capitole, Toulouse, D.I. Ecole Normale Supérieure, Paris, France.
E-mail address: radu-alexandru.dragomir@inria.fr

INRIA, D.I. Ecole Normale Supérieure, Paris, France
E-mail address: adrien.taylor@inria.fr

CNRS & D.I., UMR 8548, École Normale Supérieure, Paris, France.
E-mail address: aspremon@ens.fr

TSE (Université Toulouse I Capitole), Toulouse, France.
E-mail address: jbolte@ut-capitole.fr