OJMO

Open Journal of Mathematical Optimization

Baptiste Goujaud, Adrien Taylor & Aymeric Dieuleveut

Short Paper - Quadratic minimization: from conjugate gradient to an adaptive Polyak's momentum method with Polyak step-sizes

Volume 5 (2024), article no. 9 (10 pages)

https://doi.org/10.5802/ojmo.36

Article submitted on February 16, 2024, revised on September 3, 2024,
accepted on September 4, 2024.

CENTRE MERSENNE

Open Journal of Mathematical Optimization is a member of the Centre Mersenne for Open Scientific Publishing
http://www.centre-mersenne.org/
e-ISSN: 2777-5860

Short Paper - Quadratic minimization: from conjugate gradient to an adaptive Polyak's momentum method with Polyak step-sizes

Baptiste Goujaud
CMAP, Ecole Polytechnique, Institut Polytechnique de Paris
baptiste.goujaud@polytechnique.edu

Adrien Taylor
INRIA, Ecole Normale Supérieure, PSL Research University, Paris
adrien.taylor@inria.fr

Aymeric Dieuleveut
CMAP, Ecole Polytechnique, Institut Polytechnique de Paris
aymeric.dieuleveut@polytechnique.edu

Abstract

In this work, we propose an adaptive variation on the classical Heavy-ball method for convex quadratic minimization. The adaptivity crucially relies on so-called "Polyak step-sizes", which consists of using the knowledge of the optimal value of the optimization problem at hand instead of problem parameters such as a few eigenvalues of the Hessian of the problem. This method happens to also be equivalent to a variation of the classical conjugate gradient method, and thereby inherits many of its attractive features, including its finite-time convergence, instance optimality, and its worst-case convergence rates.

The classical gradient method with Polyak step-sizes is known to behave very well in situations in which it can be used, and the question of whether incorporating momentum in this method is possible and can improve the method itself appeared to be open. We provide a definitive answer to this question for minimizing convex quadratic functions, an arguably necessary first step for developing such methods in more general setups.

Digital Object Identifier 10.5802/ojmo.36

Keywords Optimization, Quadratic, Conjugate Gradient, Heavy-ball, Polyak step-sizes, Optimality.

Acknowledgments We would like to thank Raphael Berthier for his insightful feedback. The work of B. Goujaud and A. Dieuleveut is partially supported by ANR-19-CHIA-0002-01/chaire SCAI, and Hi!Paris. A. Taylor acknowledges support from the European Research Council (grant SEQUOIA 724063). This work was partly funded by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

1 Introduction

Consider the convex quadratic minimization problem in the form

$$\min_{x\in\mathbb{R}^d} \left\{ f(x) \triangleq \frac{1}{2}\langle x,\, Hx\rangle + \langle h,\, x\rangle \triangleq \frac{1}{2}\langle x - x_\star,\, H(x-x_\star)\rangle + f_\star \right\} \tag{1}$$

where $H \succcurlyeq 0$ is a symmetric positive semi-definite matrix, and we denote $f_\star$ the minimum value of f. In the context of large-scale optimization (i.e. $d \gg 1$), we are often interested in using first-order iterative methods for solving (1). There are many known and celebrated iterative methods for solving such quadratic problems, such as the Gradient descent and the Heavy-ball methods (a.k.a., Polyak momentum methods), including the Chebyshev and the conjugate gradient methods (see [22, Chapter 3]). Each of those methods having different specifications, the choice of the method largely depends on the application at hand. In particular, a typical drawback of momentum-based methods is that they generally require the knowledge of some problem parameters (such as extreme values of the spectrum of H). This problem is typically not as critical for simpler Gradient descent schemes with no momentum, although it generally still requires some knowledge on problem parameters if we want to avoid using linesearch-based strategies. This limitation motivates the search for adaptive strategies,

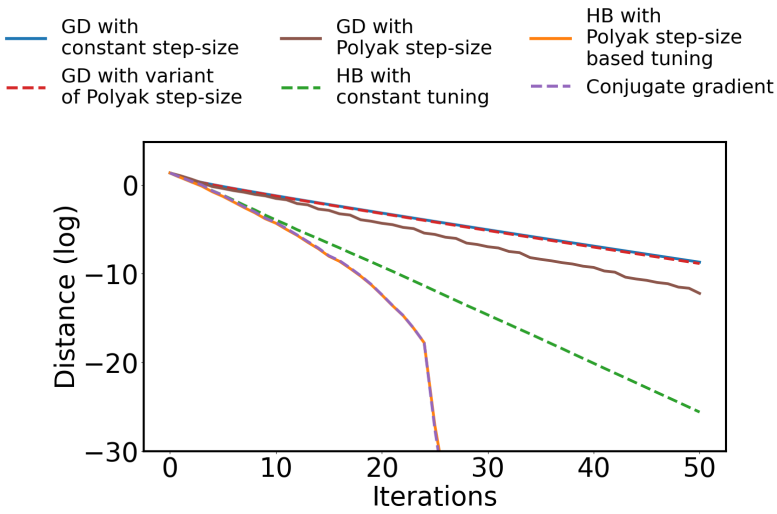


Figure 1 Comparison in semi-log scale over 50 iterations of different first-order methods applied on a 25-dimensional quadratic objective with condition number 10. *GD with constant step-size*, *GD with Polyak step-size* and *GD with variant of Polyak step-size* refer to the GD method tuned respectively with the step-size $\gamma = 2/(L+\mu)$, $\gamma_t = (f(x_t) - f_\star)/\|\nabla f(x_t)\|^2$ and $\gamma_t = 2(f(x_t) - f_\star)/\|\nabla f(x_t)\|^2$. *HB with constant tuning* is the HB method tuned with constant parameters $\gamma_t = (2/(\sqrt{L}+\sqrt{\mu}))^2$ and $m_t = ((\sqrt{L}-\sqrt{\mu})(\sqrt{L}+\sqrt{\mu}))^2$ while *HB with Polyak step-size based tuning* refers to Algorithm 1.

fixing the step-sizes using past observations about the problem at hand. In the context of (sub)gradient method, a famous adaptive strategy is the so-called Polyak step-size, which relies on the knowledge of the optimal value $f_\star$:

$$x_{t+1} = x_t - \gamma_t \nabla f(x_t), \quad \text{with } \gamma_t = \frac{f(x_t) - f_\star}{\|\nabla f(x_t)\|^2}. \tag{2}$$

Polyak steps were originally proposed in [22] for nonsmooth convex minimization; a few variants are proposed by, e.g., [2, 12, 15] including for stochastic minimization. Regarding speed, this strategy (and variants) enjoy similar theoretical convergence properties as those for Gradient descent. This method appears to perform quite well in applications where $f_\star$ can be efficiently estimated (see, e.g., [14] for an adaptation of the method for estimating it online). Therefore, a remaining open question in this context is whether the performance of this method can be improved by incorporating momentum in it. A first answer to this question was provided by [2], although it is not clear that it can match the same convergence properties as optimal first-order methods. In this work, we answer this question for the class of quadratic problems. In short, it turns out that the following conjugate gradient-like iterative procedure

$$x_{t+1} = \arg\min_x \left\{ \|x - x_\star\|^2 \text{ s.t. } x \in x_0 + \operatorname{span}\{\nabla f(x_0), \nabla f(x_1), \dots, \nabla f(x_t)\} \right\}, \tag{3}$$

can be rewritten exactly as a Heavy-ball type method whose parameters are chosen adaptively using the value of $f_\star$. This might come as a surprise, as the iteration (3) might seem impractical due to its formulation relying on the knowledge of $x_\star$. More precisely, (3) is exactly equivalent to:

$$x_{t+1} = x_t - (1 + m_t) h_t \nabla f(x_t) + m_t (x_t - x_{t-1}), \tag{4}$$

with parameters

$$\forall\, t \geqslant 0, \qquad h_t \triangleq \frac{2(f(x_t) - f_\star)}{\|\nabla f(x_t)\|^2}, \tag{5}$$

$$m_0 \triangleq 0 \;\text{ and }\; \forall\, t \geqslant 1, \qquad m_t \triangleq \frac{-(f(x_t) - f_\star)\langle \nabla f(x_t), \nabla f(x_{t-1})\rangle}{(f(x_{t-1}) - f_\star)\|\nabla f(x_t)\|^2 + (f(x_t) - f_\star)\langle \nabla f(x_t), \nabla f(x_{t-1})\rangle}. \tag{6}$$

In (4), m_t corresponds to the momentum coefficient and h_t to a step-size (see Theorem 4 for a discussion about this choice of parametrization). With the tuning of (5), this step-size is twice the Polyak step-size in (2). This Heavy-ball momentum method with Polyak step-sizes is summarized in Algorithm 1 and illustrated in Figure 1. Due to its equivalence with (3), the Heavy-ball method (4) inherits nice advantageous properties of

Algorithm 1 Adaptive Heavy-ball algorithm

Input T, $f_\star$ and routines to evaluate $f : x \mapsto f(x) \triangleq \frac{1}{2}\langle x - x_\star,\, H(x - x_\star)\rangle + f_\star$ and $\nabla f : x \mapsto \nabla f(x) = H(x - x_\star)$.
Initialize $x_0 \in \mathbb{R}^d$, $m_0 = 0$
for $t = 0 \cdots T-1$ **do**
$h_t = \frac{2(f(x_t)-f_\star)}{\|\nabla f(x_t)\|^2}$
$x_{t+1} = x_t - (1+m_t)h_t\nabla f(x_t) + m_t(x_t - x_{t-1})$
$m_{t+1} = \frac{-(f(x_{t+1})-f_\star)\langle\nabla f(x_{t+1}),\nabla f(x_t)\rangle}{(f(x_t)-f_\star)\|\nabla f(x_{t+1})\|^2+(f(x_{t+1})-f_\star)\langle\nabla f(x_{t+1}),\nabla f(x_t)\rangle}$
end
Result: x_T

conjugate gradient-type methods, including:

- **i.** finite-time convergence: the problem (1) is solved exactly after at most d iterations,
- **ii.** instance optimality: for *all* $H \succcurlyeq 0$, no first-order method satisfying $x_{t+1} \in x_0 + \text{span}\{\nabla f(x_0), \dots, \nabla f(x_t)\}$ results in a smaller $\|x_t - x_\star\|$,
- **iii.** it inherits optimal worst-case convergence rates on quadratic functions.

Of course, a few of those points need to be nuanced in practice due to finite precision arithmetic. The equivalence between (3) and (4) is formally stated in the following theorem.

Theorem 1. *Let $(x_t)_{t\in\mathbb{N}}$ be the sequence defined by the recursion* (3)*, namely such that for any t, x_{t+1} is the Euclidean projection of $x_\star$ onto the affine subspace $x_0 + \text{span}\{\nabla f(x_0), \nabla f(x_1), \dots, \nabla f(x_t)\}$. Then $(x_t)_{t\in\mathbb{N}}$ is the sequence generated by Algorithm 1.*

Theorem 1 turns out to be a particular case of a more general result stating that the iterates of any conjugate gradient-type method described with a polynomial Q as

$$x_{t+1} = \arg\min_x \left\{\langle x - x_\star,\, Q(H)(x - x_\star)\rangle \text{ s.t. } x \in x_0 + \text{span}\{\nabla f(x_0), \dots, \nabla f(x_t)\}\right\}, \qquad \text{(Q-minimization)}$$

are equivalently written in terms of an adaptive Heavy-ball iteration. In particular, (3) corresponds to equation (Q-minimization) with $Q(x) = 1$. Similarly, classical conjugate gradient method corresponds to equation (Q-minimization) with $Q(x) = x$ (this fact is quite famous, see, e.g., [19]). We were surprised not to find this general result written as is in the literature, and we therefore provide it in Section 2. The key point of this work is that the equivalent Heavy-ball reformulation of (3) can be written in terms of $f_\star$, thereby obtaining a momentum-based Polyak step-size.

Notations

We denote $\preccurlyeq$ the order between symmetric matrices; SpH the spectrum of the matrix H, namely its set of eigenvalues; $\mathbb{R}_d[X]$ the set of polynomials with degree at most d.

1.1 Preliminary material

Worst-case optimality

Solving (1) is a very important problem and several methods have been proposed to achieve this goal. They are compared with each other through notions of performance. This consists of evaluating the precision of an algorithm over the functions of a given class after a given number T of iterations. The main framework is *worst-case analysis* and the precision is the value of a given metric, e.g. the distance of the last iterate to the optimizer $\|x_T - x_\star\|$, the function value of the last iterate $f(x_T) - f(x_\star)$, or its gradient norm $\|\nabla f(x_T)\|$. The *worst-case analysis* framework consists of finding the guarantees of a method that hold for each and every function of a given class. For instance, the class of L-smooth μ-strongly convex quadratic functions described as quadratic functions verifying $\mu I \preccurlyeq H \preccurlyeq LI$ for given $0 < \mu \leqslant L$. The *Gradient descent (GD)* method characterized by the update

$$x_{t+1} = x_t - \gamma_t\nabla f(x_t) \qquad (7)$$

therefore verifies $\|x_t - x_\star\| = O((\frac{L-\mu}{L+\mu})^t)$ on all such functions for $\gamma_t = \frac{2}{L+\mu}$. Thanks to a relationship with polynomial analysis, [9] proved that the *Chebyshev method*, described as

$$x_{t+1} = x_t - \gamma_t \nabla f(x_t) + m_t(x_t - x_{t-1}), \tag{8}$$

for a well chosen tuning of the parameters γ_t and m_t ($m_t = \big(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\big)^2 \frac{1+((\sqrt{L}-\sqrt{\mu})/(\sqrt{L}+\sqrt{\mu}))^{2(t-1)}}{1+((\sqrt{L}-\sqrt{\mu})/(\sqrt{L}+\sqrt{\mu}))^{2(t+1)}}$, $\gamma_t = \frac{2}{L+\mu}(1+m_t)$), is *worst-case optimal* on this class of function, achieving the guarantee $\|x_t - x_\star\| = O\big(\big(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\big)^t\big)$ (often referred to as "acceleration"). Methods based on this two-term recursion are called "Heavy-ball" or "Polyak momentum" [21]. In particular, the stationary regime of the Chebyshev method is the *Heavy-ball (HB)* method tuned with $m_t = \big(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\big)^2$ and $\gamma_t = \frac{2}{L+\mu}(1+m_t) = \big(\frac{2}{\sqrt{L}+\sqrt{\mu}}\big)^2$ and achieves the worst-case guarantee $\|x_t - x_\star\| = O\big(t\big(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\big)^t\big)$, close to the optimal one achieved by the Chebyshev method. Note that (8) is another formulation of (4) where $\gamma_t = (1+m_t)h_t$. In all the aforementioned tuning, h_t has the same value: $h_t = \frac{2}{L+\mu}$ (see Theorem 4 for more detailed discussion on this).

Span of gradients and Krylov subspaces

All methods described above can be defined using a recursion:

$$x_t = x_0 - \sum_{i=0}^{t-1} \gamma_i^{(t)} \nabla f(x_i) \tag{9}$$

for some sequence $(\gamma_i^{(t)})_{i\in[\![0,t-1]\!]}$. Note that the recursion (9) can also be explicitly written as $x_t = x_0 - H\sum_{i=0}^{t-1}\gamma_i^{(t)}x_i$, and therefore, $x_t - x_0 \in H\mathrm{span}(\{x_i\}_{i\in[\![0,t-1]\!]})$. We deduce by recursion that $x_t - x_0 \in H\mathcal{K}_t(H, x_0)$ where $\mathcal{K}_t(H, x_0) \triangleq \mathrm{span}(\{H^i x_0\}_{i\in[\![0,t-1]\!]})$ is called *order-t Krylov subspace generated by H and x_0*. This creates a link between first-order algorithms and polynomials, summarized in the following lemma (which is implicitly used in [9] and formally stated, e.g., in [10, Proposition 4.1]).

Lemma 2. *Let f be quadratic convex* (1)*. The iterates x_t satisfy*

$$x_t \in x_0 + \mathrm{span}\{\nabla f(x_0), \dots, \nabla f(x_{t-1})\}\,, \tag{10}$$

where x_0 is the initial approximation of $x_\star$, if and only if there exists a sequence of polynomials $(P_t)_{t\in\mathbb{N}}$, each of degree at most 1 more than the highest degree of all previous polynomials and P_0 of degree 0 (hence the degree of P_t is at most t), such that

$$\forall\, t, \quad x_t - x_\star = P_t(H)(x_0 - x_\star), \quad P_t(0) = 1\,. \tag{11}$$

Similar to the way we use this technique below, this lemma has already been extensively used to design methods; see, e.g., [6, Chapter 1] or the blog post by [20] for gentle introductions to this technique. For instance, we can use this technique for optimizing the step-size of the gradient method, or to derive the Chebyshev method, which optimizes the worst-case on the class of smooth and strongly convex quadratic functions (see [8]). More recently, [10] used it to derive a method that can take advantage of a possible gap in the spectrum of H. This approach has also been used for other applications such as accelerated gossip algorithms [4].

Adaptive methods

In Theorem 2, while $P_t(H)$ is a polynomial evaluated on the matrix H, its scalar coefficients might or might not depend on H. If they depend on H, we say that the associated method is adaptive. Non-adaptive methods suffer from two main drawbacks: (i) they use the same parameters for all the functions within the class of problems, not taking advantage of the observed quantities; (ii) the underlying parameters must scale with the function class parameters, and therefore depend on the values of L and μ, which are generally difficult to estimate (and actually do not correspond to first-order information, as they rely on the Hessian of the function at hand). Ultimately, adaptive methods aim at solving those issues by choosing parameters (step-size, momentum, etc.) on the fly.

Polyak steps

It is straightforward that a Gradient descent update verifies on any convex function f that $\|x_{t+1} - x_\star\|^2 \leqslant \|x_t - x_\star\|^2 - 2\gamma_t(f - f_\star) + \gamma_t^2\|\nabla f(x_t)\|^2$. [22] argues that, based on this inequality, the best-guaranteed progression is then achieved for $\gamma_t = \frac{f(x_t)-f_\star}{\|\nabla f(x_t)\|^2}$. This choice is called "Polyak step-size" and has been studied intensively even recently [12, 15]. Other variants of the latter have been proposed. For instance [2, Variant 1] suggested the step-size $\gamma_t = 2\frac{f(x_t)-f_\star}{\|\nabla f(x_t)\|^2}$. This also optimizes the exact progress of one Gradient descent update over quadratics realizing a projection of $x_t - x_\star$ over the orthogonal subspace of $\nabla f(x_t)$. Therefore, the Polyak step-size strategy applied to the Gradient descent method achieves the same worst-case guarantee of the well-tuned fixed step-size Gradient descent method, while not relying on Hessian information. Moreover, due to its adaptivity to each function, and since generic functions do not look like worst-cases, the Polyak step-size strategy applied to the Gradient descent method performs very well in practice (See Figure 1 and [2, Figure 1]), sometimes even beating the well-tuned non-adaptive Heavy-ball method even if the worst-case guarantees are sorted in a different order.

Instance-optimality

While optimal worst-case methods of the form of (9) have been found with predetermined parameters, it would be better to find a method under the form of (9) that is optimal (for some performance metric), not only on worst-case analysis, but on each function individually, taking advantage of the adaptivity of the parameters. The well-known conjugate gradient method achieves this goal when the performance metric is the function value of the last iterate. The MinRes method attacks the problem minimizing the gradient norm of the last iterate.

Contributions

In this work, we derive iterative methods in the form of (9) (which iterates lie in the span of previously observed gradients) that are instance-optimal for a variety of performance metrics. All those methods updates are variations of the Heavy-ball two-term recursion (8) with only parameters γ_t and m_t changing from one method to another. Finally, we show (see Theorem 1) that for a well-chosen yet classical performance metric, this associated method Algorithm 1 is not relying on second-order information at all (not even L and μ). Instead, it uses a classical variant of the Polyak step-size $\frac{2(f(x_t)-f_\star)}{\|\nabla f(x_t)\|^2}$, providing an answer to the question "Can we accelerate methods with Polyak step-size?".

1.2 Related works

Polyak step-sizes were proposed in [22]. Despite the dependency on $f_\star$, the Polyak step-size is more studied theoretically and used in practice due to its efficiency when applied to real-world problems. Recent works (e.g. [7, 15]) argue that this dependency is not a practical issue for many problems which we can assume verify $f_\star = 0$ (see Section 3). A few variants of the Polyak step-size strategy were proposed by [2], including a version incorporating a Nesterov-type momentum [17], achieving a worst-case guarantee of $\|x_t - x_\star\|^2 = O((1 - 2(\mu/L)^{2/3})^{2t})$ over the class of (non-necessarily quadratic) L-smooth μ-strongly convex functions, thereby improving over previous works on adaptive first-order methods. However, the proposed method does not allow to remove the dependency on L and does not achieve the black-box complexity of smooth strongly convex minimization [18]. In [15], the authors study the stochastic Polyak step-size, whereas [7] applies it to Mirror descent.

Many alternative adaptive methods have been proposed in the past. Among them, let us mention [3] which introduced the so-called Barzilai–Borwein step-size rule, and the more recent [16] which developed a step-size policy that adapts to the local geometry with convergence guarantees beyond quadratic minimization.

Finally, let us mention that, while working on quadratic functions is a prerequisite for the proposed method to work on a more general class of functions, the reverse is not true. Recent papers (see [1, 11]) exhibit the failure of famous algorithms to perform on the class of smooth strongly convex functions as good as on quadratic ones.

2 Main theorem

This section states and proves Theorem 1. In short, given a certain function f (characterized by H, $x_\star$ and $f_\star$ here) and a starting point x_0, we search for an iterative procedure, possibly adaptive, verifying the polynomial-based expression (11) such that x_t converges as fast as possible to $x_\star$ for some predefined performance

metric. Most classical ways to measure the performance of such optimization schemes include the distance to optimum $\|x_t - x_\star\|^2$, the function accuracy gap $f(x_t) - f_\star$, the squared gradient norm $\|\nabla f(x_t)\|^2$, and linear combinations of the former. Let us abstract those notions by denoting the performance measure of choice by $\langle x_t - x_\star, Q(H)(x_t - x_\star)\rangle$ (with Q a predefined polynomial that is positive on $\mathbb{R}_{>0}$). Then, we consider the iterative scheme given by (Q-minimization).

$$x_{t+1} = \arg\min_x \left\{\langle x - x_\star, Q(H)(x - x_\star)\rangle \text{ s.t. } x \in x_0 + \operatorname{span}\{\nabla f(x_0), \dots, \nabla f(x_t)\}\right\}. \tag{Q-minimization}$$

The next theorem provides an explicit instance-optimal method to solve equation (Q-minimization).

Theorem 3 (Main Theorem)**.** *The unique solution to equation* (Q-minimization) *is given by the Heavy-ball procedure*

$$x_{t+1} = x_t - (1 + m_t)h_t\nabla f(x_t) + m_t(x_t - x_{t-1}) \tag{12}$$

where

$$\begin{cases} h_t = \dfrac{\langle x_t - x_\star, HQ(H)(x_t - x_\star)\rangle}{\langle x_t - x_\star, H^2Q(H)(x_t - x_\star)\rangle}; \\ m_t = \dfrac{-b_th_t}{1 + b_th_t}, \quad \textit{with } b_t = \dfrac{\langle x_t - x_\star, H^2Q(H)(x_{t-1} - x_\star)\rangle}{\langle x_{t-1} - x_\star, HQ(H)(x_{t-1} - x_\star)\rangle}. \end{cases} \tag{13}$$

Remark that setting $Q(X) = X$ leads to a nice expression of the conjugate gradient method (see [22, Section 3.2.2]). Indeed, setting $Q(X) = X$ corresponds to optimally minimizing the excess loss $f(x_t) - f_\star$.

As already known, the conjugate gradient method requires the knowledge of H (or a Hessian vector product) in addition to first-order information to proceed. This is also a priori the case for most of all other choices of $Q(\,\cdot\,)$. In the particular case of $Q(X) = 1$, which corresponds to minimizing the distance to the optimum (see (3)), we can use an alternate writing making use of $f_\star$:

$$\begin{cases} h_t = \dfrac{2(f(x_t) - f_\star)}{\|\nabla f(x_t)\|^2} \\ m_t = \dfrac{-b_th_t}{1 + b_th_t}, \quad \text{with } b_t = \dfrac{\langle \nabla f(x_t), \nabla f(x_{t-1})\rangle}{2(f(x_{t-1}) - f_\star)}. \end{cases} \tag{14}$$

Proof.

Designing methods from the polynomial point of view. As suggested by Theorem 2, we look for an iterative method that can be expressed in the form $x_t - x_\star = P_t(H)(x_0 - x_\star)$, where P_t is a t^{th} degree polynomial with $P_t(0) = 1$. Furthermore, as we look for an instance-optimal method, the latter polynomial must be instance-specific, and the coefficients of P_t should depend on H (and should describe the iterative procedure (Q-minimization)).

Recalling that H is real symmetric matrix, we denote by $\lambda \in \operatorname{Sp}(H)$ its eigenvalues and by v_λ the associated orthonormal basis of eigenvectors, leading to $H = \sum_{\lambda\in\operatorname{Sp}(H)} \lambda v_\lambda v_\lambda^T$. The quantity to be minimized can now be written as:

$$\begin{aligned} \langle x_t - x_\star, Q(H)(x_t - x_\star)\rangle &= \langle x_0 - x_\star, P_t(H)^TQ(H)P_t(H)(x_0 - x_\star)\rangle & (15)\\ &= \sum_{\lambda\in\operatorname{Sp}(H)} Q(\lambda)P_t(\lambda)^2\langle x_0 - x_\star, v_\lambda\rangle^2 & (16)\\ &= \int_{\lambda\in\mathbb{R}^+} P_t(\lambda)^2 \,\mathrm{d}\lambda_{\mathrm{Q}}(\lambda) & (17) \end{aligned}$$

with λ_{Q} the discrete measure $\sum_{\lambda\in\operatorname{Sp}(H)} Q(\lambda)\langle x_0 - x_\star, v_\lambda\rangle^2\,\delta_\lambda$ (we sometimes use the shorthand notation $\int P_t^2\,\mathrm{d}\lambda_{\mathrm{Q}}$ for (17) in what follows). It is clear that (17) is 0 if and only if $P_t(\lambda) = 0$ for all $\lambda \in \operatorname{Sp}(H)$. As a consequence, we conclude that (i) choosing the right sequence of polynomials leads to convergence in exactly $|\operatorname{Sp}(H)|$ iterations, and (ii) $\langle P^{(1)}, P^{(2)}\rangle_Q \triangleq \int P^{(1)}P^{(2)}\,\mathrm{d}\lambda_{\mathrm{Q}}$ is an inner product on $\mathbb{R}_{|\operatorname{Sp}(H)|-1}[X]$. We therefore want to solve

$$\begin{cases} \underset{P_t\in\mathbb{R}_t[X]}{\text{minimize}} & \|P_t\|_Q^2 \\ \text{subject to} & P_t(0) = 1 \end{cases} \tag{18}$$

for any $t \leqslant |\mathrm{Sp}(H)| - 1$ where $\|P\|_Q^2 \triangleq \langle P, P\rangle_Q = \int P^2 \, \mathrm{d}\lambda_{\mathrm{Q}}$ denotes the underlying norm of the inner product $\langle \cdot, \cdot \rangle_Q$. For $t \geqslant |\mathrm{Sp}(H)|$, we consider instead P_t as a multiple of the polynomial $\prod_{\lambda \in \mathrm{Sp}(H)}(X - \lambda)$ in X. The next steps are somewhat standard and follow a classical pattern for solving (18) (see, e.g. [4] and the references therein).

From minimal norm to orthogonality. The solution to (18) is the projection of the polynomial 0 over the affine space $\{P \in \mathbb{R}_t[X] \mid P(0) = 1\}$ with respect to the inner product $\langle \cdot, \cdot \rangle_Q$. A necessary and sufficient condition for P to be the solution of problem (18) is therefore to verify $\langle 0 - P, \Delta P\rangle_Q = 0$ for any ΔP in the vectorial subspace $\{P \in \mathbb{R}_t[X] \mid P(0) = 0\} = X\mathbb{R}_{t-1}[X]$. Hence P_t solves problem (18) iff

$$\langle P_t, XR\rangle_Q = 0, \quad \forall\, R \in \mathbb{R}_{t-1}[X]. \tag{19}$$

Note however, that for any $(P, R) \in \mathbb{R}[X]^2$,

$$\langle P, XR\rangle_Q = \int_{\lambda \in \mathbb{R}^+} P(\lambda) \cdot \lambda R(\lambda) \, \mathrm{d}\lambda_{\mathrm{Q}}(\lambda) = \int_{\lambda \in \mathbb{R}^+} P(\lambda) \cdot R(\lambda) \, \mathrm{d}\lambda_{\mathrm{XQ}}(\lambda) \triangleq \langle P, R\rangle_{XQ}$$

with $\mathrm{d}\lambda_{\mathrm{XQ}}(\lambda) \triangleq \lambda \mathrm{d}\lambda_{\mathrm{Q}}(\lambda) = \sum_{\lambda \in \mathrm{Sp}(H)} \lambda Q(\lambda) \langle x_0 - x_\star, v_\lambda\rangle^2 \, \delta_\lambda$. Using the latter inner product, the condition for P_t to be the solution to problem (18) becomes:

$$P_t \in \mathbb{R}_{t-1}[X]^{\perp_{XQ}}. \tag{20}$$

Hence, $(P_t)_{t \in \mathbb{N}}$ is a family of orthogonal polynomials for the inner product $\langle \cdot, \cdot \rangle_{XQ}$.

From orthogonality to recursion. We now focus on finding an explicit expression for the polynomials P_t. As for all families of orthogonal polynomials, $(P_t)_{t \in \mathbb{N}}$ can be obtained through a two-term recursion of the form:

$$P_{t+1}(X) = (a_t X + b_t)P_t(X) + c_t P_{t-1}(X), \quad \text{for some } (a_t, b_t, c_t) \in \mathbb{R}^3, \tag{21}$$

which is easy to verify by induction. Our goal is to find a_t, b_t and c_t. First, notice that $(a_t X + b_t)P_t(X) + c_t P_{t-1}(X)$ is orthogonal to $\mathbb{R}_{t-2}[X]$ independently of the values of a_t, b_t and c_t. Those three coefficients can be found via the following three conditions: (i) $\langle P_{t+1}, P_t\rangle_{XQ} = 0$, (ii) $\langle P_{t+1}, P_{t-1}\rangle_{XQ} = 0$, and (iii) $P_{t+1}(0) = 1$.

More precisely, it is clear that $a_t \neq 0$ for P_{t+1} to be of degree $t + 1$. Therefore, one can factorize by a_t. Reparametrizing (21), one can write

$$P_{t+1}(X) = \frac{(\widetilde{a}_t - X)P_t(X) + \widetilde{b}_t P_{t-1}(X)}{\widetilde{c}_t}, \text{ with } (\widetilde{a}_t, \widetilde{b}_t, \widetilde{c}_t) \in \mathbb{R}^3.$$

Moreover, evaluation at $X = 0$ gives $\frac{\tilde{a}_t + \tilde{b}_t}{\tilde{c}_t} = 1$, thereby enforcing $\widetilde{c}_t = \widetilde{a}_t + \widetilde{b}_t$. It remains to verify the two orthogonality conditions (independent of $\widetilde{c}_t$):

$$\begin{aligned}
\widetilde{a}_t \langle P_t, P_t\rangle_{XQ} \quad &+ \widetilde{b}_t \langle P_{t-1}, P_t\rangle_{XQ} \quad = \langle XP_t(X), P_t(X)\rangle_{XQ}, \\
\widetilde{a}_t \langle P_t, P_{t-1}\rangle_{XQ} &+ \widetilde{b}_t \langle P_{t-1}, P_{t-1}\rangle_{XQ} = \langle XP_t(X), P_{t-1}(X)\rangle_{XQ}.
\end{aligned}$$

Note that this system of equations is decoupled since $\langle P_{t-1}, P_t\rangle_{XQ} = 0$, and we finally arrive to

$$P_{t+1}(X) = \frac{(\widetilde{a}_t - X)P_t(X) + \widetilde{b}_t P_{t-1}(X)}{\widetilde{a}_t + \widetilde{b}_t}, \tag{22}$$

with

$$\begin{cases}
\widetilde{a}_t = \dfrac{\langle XP_t(X), P_t(X)\rangle_{XQ}}{\langle P_t, P_t\rangle_{XQ}} = \dfrac{\langle x_t - x_\star,\, H^2 Q(H)(x_t - x_\star)\rangle}{\langle x_t - x_\star,\, HQ(H)(x_t - x_\star)\rangle}, \\
\widetilde{b}_t = \dfrac{\langle XP_t(X), P_{t-1}(X)\rangle_{XQ}}{\langle P_{t-1}, P_{t-1}\rangle_{XQ}} = \dfrac{\langle x_t - x_\star,\, H^2 Q(H)(x_{t-1} - x_\star)\rangle}{\langle x_{t-1} - x_\star,\, HQ(H)(x_{t-1} - x_\star)\rangle}.
\end{cases} \tag{23}$$

From a polynomial recursion to an iterative optimization method. For reaching the final desired result, we simply multiply (22) (evaluated in H) by $x_0 - x_\star$:

$$x_{t+1} - x_\star = \frac{\widetilde{a}_t(x_t - x_\star) - H(x_t - x_\star) + \widetilde{b}_t(x_{t-1} - x_\star)}{\widetilde{a}_t + \widetilde{b}_t} = x_t - x_\star - \frac{1}{\widetilde{a}_t + \widetilde{b}_t}\nabla f(x_t) + \frac{-\widetilde{b}_t}{\widetilde{a}_t + \widetilde{b}_t}(x_t - x_{t-1}),$$

thereby reaching the desired

$$x_{t+1} = x_t - (1+m_t)h_t\nabla f(x_t) + m_t(x_t - x_{t-1}) \tag{24}$$

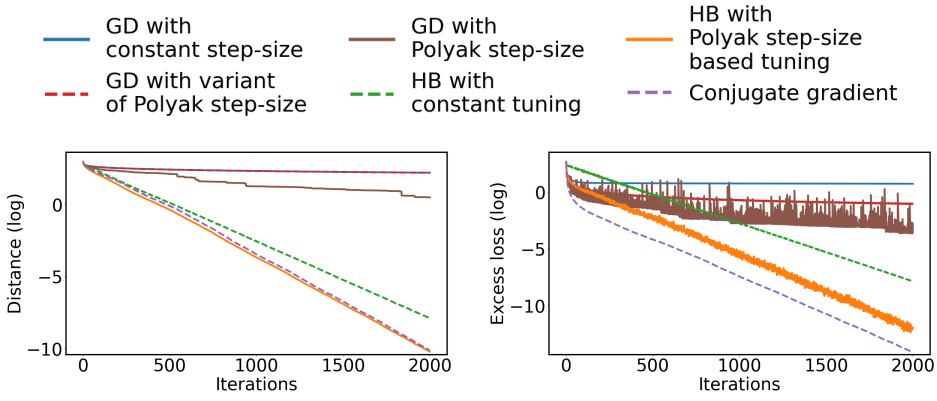
$$\text{with} \quad (1+m_t)h_t = \frac{1}{\widetilde{a}_t + \widetilde{b}_t} \quad \text{and} \quad m_t = \frac{-\widetilde{b}_t}{\widetilde{a}_t + \widetilde{b}_t}, \qquad \text{hence,} \qquad h_t = \frac{1}{\widetilde{a}_t} \quad \text{and} \quad m_t = \frac{-\widetilde{b}_t h_t}{1 + \widetilde{b}_t h_t}. \tag{25}$$

From (24) and (25), we recognize a Heavy-ball method with some variable step-size h_t and momentum term m_t corresponding to the theorem statement, thereby concluding the proof. ◀

▶ Remark 4 (Step-size parametrization). While the γ_t plays a different role in (7) and (8), they are both usually called "step-size" by default. But we noticed that both in the Chebyshev method and the Heavy-ball method (optimally tuned), $h_t = \frac{\gamma_t}{1+m_t}$ is exactly $\frac{2}{L+\mu}$, value of the optimal step-size for Gradient descent, and that the Heavy-ball method converges on quadratic functions exactly when $h_t = \frac{\gamma_t}{1+m_t} \leqslant \frac{2}{L}$, as for Gradient descent. In (5), we notice again that the value of h_t is the optimal step-size for a single step of Gradient descent. For this reason, we believe that the natural parametrization of the Heavy-ball methods should be $x_{t+1} = x_t - (1+m_t)h_t\nabla f(x_t) + m_t(x_t - x_{t-1})$ and that h_t should be referred to as the "natural" step-size. Indeed, when one thinks of the Heavy-ball method with Polyak step-sizes, they would set γ_t to the Polyak step-size, not $h_t = \frac{\gamma_t}{1+m_t}$. We therefore provide a novel view on what should be tested.

3 Numerical experiments

In this section, we compare Gradient descent, Heavy-ball, and conjugate gradient methods in an adaptive setting or not. Figure 2 shows the performance of all these methods on a quadratic objective with known minimal value $f_\star$. The hessian of this quadratic objective has been generated from a sequence of eigenvalues with geometric increase, and a random orthogonal transformation. The difference between Figures 1 and 2 is the dimension of the problem as well as the condition number of the objective function. Due to finite precision arithmetic, the finite-time convergence is not visible when the condition number is too large. However, both figures show that our method and the conjugate gradient algorithm behave similarly and faster than the other methods. The code can be found on the following GitHub repository: `https://github.com/bgoujaud/Heavy-ball_polyak_steps`.



(a) Comparison of distances to optimum **(b)** Comparison of excess losses

Figure 2 Comparison in semi-log scale over 2000 iterations of different first-order methods applied on a 1000-dimensional quadratic objective with condition number 10^5. *GD with constant step-size*, *GD with Polyak step-size* and *GD with variant of Polyak step-size* refer to the GD method tuned respectively with the step-size $\gamma = 2/(L+\mu)$, $\gamma_t = (f(x_t) - f_\star)/\|\nabla f(x_t)\|^2$ and $\gamma_t = 2(f(x_t) - f_\star)/\|\nabla f(x_t)\|^2$. *HB with constant tuning* is the HB method tuned with constant parameters $\gamma_t = (2/(\sqrt{L} + \sqrt{\mu}))^2$ and $m_t = ((\sqrt{L} - \sqrt{\mu})(\sqrt{L} + \sqrt{\mu}))^2$ while *HB with Polyak step-size based tuning* refers to Algorithm 1.

4 Concluding remarks and discussion

Polyak step-sizes are known for their general good working performances when the optimal value for the optimization problem at hand is known. Whether Polyak step-sizes can be used together with momentum for obtaining accelerated first-order methods appears to be an open question [2], which we answer in the simpler

case of convex quadratic minimization. In this context, we argue that not only this tuning works well, but also it pops up naturally when investigating instance-optimal first-order iterative methods. Furthermore, we believe it is a necessary step for being able to understand more general optimization settings beyond quadratics. As our method does not seem to work well beyond quadratics, we leave further investigations on this topic for future work.

Among our competitors, we note that the celebrated conjugate gradient (CG) method is another instance-optimal algorithm for quadratics. Whereas our method minimizes the distance to the solution at each iteration, CG is instance-optimal for minimizing function values at each iteration. Perhaps interestingly, the two methods appeared to behave similarly in our numerical experiments. That being said, the main practical differences between the two methods are that CG Heavy-ball-like formulation naturally relies on higher order information while Polyak step-sizes do require knowledge of $f_\star$. In typical optimization problems, this value is not known. However, there are a few settings where this value is actually well-known, typically when $f_\star = 0$ generically (in machine learning, this setting is known as the "interpolation" regime; an alternative could be to use Polyak-steps as a competitor to MinRes). Finally, let us mention that a few generalizations of CG, often referred to as nonlinear conjugate gradient, were studied in the literature (see, e.g., [5, 13, 19]). A compelling direction for future research would involve expanding our proposed method to a class of non-quadratic objectives.

References

1 Jason M. Altschuler and Pablo A. Parrilo. Acceleration by stepsize hedging I: Multi-step descent and the silver stepsize schedule. `https://arxiv.org/abs/2309.07879`, 2023.

2 Mathieu Barré, Adrien Taylor, and Alexandre d'Aspremont. Complexity guarantees for Polyak steps with momentum. In *Conference on Learning Theory*, pages 452–478. PMLR, 2020.

3 Jonathan Barzilai and Jonathan M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8(1):141–148, 1988.

4 Raphaël Berthier, Francis Bach, and Pierre Gaillard. Accelerated gossip in networks of given dimension using Jacobi polynomial iterations. *SIAM J. Math. Data Sci.*, 2(1):24–47, 2020.

5 Joseph-Frédéric Bonnans, Jean-Charles Gilbert, Claude Lemaréchal, and Claudia A. Sagastizábal. *Numerical optimization: theoretical and practical aspects*. Universitext. Springer, 2006.

6 Alexandre d'Aspremont, Damien Scieur, and Adrien Taylor. Acceleration Methods. *Found. Trends Optim.*, 5(1-2):1–245, 2021.

7 Ryan D'Orazio, Nicolas Loizou, Issam Laradji, and Ioannis Mitliagkas. Stochastic mirror descent: Convergence analysis and adaptive variants via the mirror stochastic Polyak stepsize. *Transactions on Machine Learning Research (TMLR)*, 2021.

8 Bernd Fischer. *Polynomial based iteration methods for symmetric linear systems*. Society for Industrial and Applied Mathematics, 2011.

9 Gene H. Golub and Richard S. Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods. *Numer. Math.*, 3(1):157–168, 1961.

10 Baptiste Goujaud, Damien Scieur, Aymeric Dieuleveut, Adrien Taylor, and Fabian Pedregosa. Super-acceleration with cyclical step-sizes. In *International Conference on Artificial Intelligence and Statistics*, pages 3028–3065. PMLR, 2022.

11 Baptiste Goujaud, Adrien Taylor, and Aymeric Dieuleveut. Provable non-accelerations of the heavy-ball method. `https://arxiv.org/abs/2307.11291`, 2023.

12 Robert M. Gower, Mathieu Blondel, Nidham Gazagnadou, and Fabian Pedregosa. Cutting Some Slack for SGD with Adaptive Polyak Stepsizes. `https://arxiv.org/abs/2202.12328`, 2022.

13 William W. Hager and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.*, 2(1):35–58, 2006.

14 Elad Hazan and Sham Kakade. Revisiting the Polyak step size. `https://arxiv.org/abs/1905.00313`, 2019.

15 Nicolas Loizou, Sharan Vaswani, Issam Laradji, and Simon Lacoste-Julien. Stochastic Polyak step-size for SGD: An adaptive learning rate for fast convergence. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1306–1314. PMLR, 2021.

16 Yura Malitsky and Konstantin Mishchenko. Adaptive Gradient Descent without Descent. In *International Conference on Machine Learning (ICML)*, pages 6702–6712. PMLR, JMLR.org, 2020.

17 Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Sov. Math., Dokl.*, 27(2):372–376, 1983.

18 Yurii Nesterov. *Introductory Lectures on Convex Optimization*, volume 87 of *Applied Optimization*. Springer, 2003.

19 Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer, 1999.

20 Fabian Pedregosa. A Hitchhiker's Guide to Momentum. `http://fa.bianp.net/blog/2021/hitchhiker/`, 2021.

21 Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *U.S.S.R. Comput. Math. Math. Phys.*, 4(5):1–17, 1964.

22 Boris T. Polyak. *Introduction to optimization*. Optimization Software New York, 1987.