Quadratic minimization: from conjugate gradients to an adaptive heavy-ball method with Polyak step-sizes

author names withheld

Under Review for OPT 2022

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 in 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, a arguably necessary first step for developing such methods in more general setups.

Keywords: Optimization; Heavy-ball; Adaptive; Polyak step-size; Quadratic

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 (a few instances of such problems are presented in Appendix B). In the context of large-scale optimization (i.e., $d\gg 1$), we are often interested in using first-order iterative methods for solving eq. (1). There are many known and celebrated iterative methods for solving such quadratic problems, including conjugate gradients, heavy-ball methods (a.k.a., Polyak momentum), Chebyshev methods, and gradient descent. 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, fixing step-size using past observations about the problem at hand. In the context of (sub)gradient descent, 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 Polyak [21] for nonsmooth convex minimization; it is also discussed in Boyd et al. [4] and a few variants are proposed by, e.g., [1, 10, 12] including for stochastic minimization. In terms of speed, this strategy (and variants) enjoy similar theoretical convergence properties as those for gradient descent. This methods appears to perform quite well in applications where $f_\star$ can be efficiently estimated—see, e.g., [11] for an adaptation of the method for estimating it online (although [11] contains a number of mistakes, the approach can be corrected to achieve the claimed target). Therefore, a remaining open question in this context is whether the performances of this method can be improved by incorporating momentum in it. A first answer to this question was provided by Barré et al. [1], 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

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

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

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

with parameters

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

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

In eq. (4), m_t corresponds to the momentum coefficient and h_t to a step-size. With the tuning of section 1, this step-size is twice the Polyak step-size in eq. (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 eq. (3), the heavy-ball method eq. (4) inherits nice advantageous properties of conjugate gradient-type methods, including:

(i) finite-time convergence: the problem eq. (1) is solved exactly after at most d iterations,
(ii) instance optimality: for *all* $H \succcurlyeq 0$, no first-order method satisfying $x_{t+1} \in x_0 + \operatorname{span}\{\nabla f(x_0), \ldots, \nabla f(x_t)\}$ 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 needs to be nuanced in practice due to finite precision arithmetic. The equivalence between eq. (3) and eq. (4) is formally stated in the following theorem.

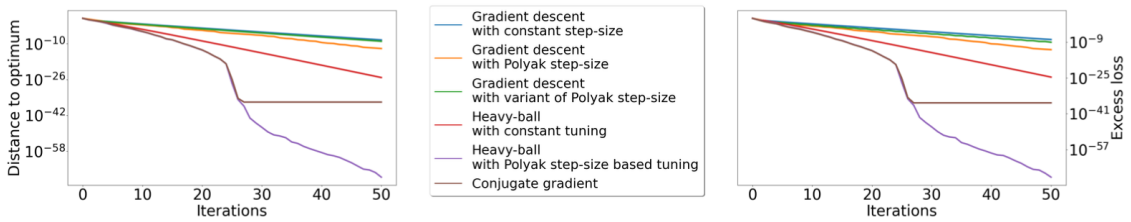
Theorem 1.1 *Let $(x_t)_{t\in\mathbb{N}}$ be the sequence defined by the recursion eq. (3), namely such that for any t, x_{t+1} is the Euclidean projection of $x_\star$ onto the affine subspace $x_0 + \operatorname{span}\{\nabla f(x_0), \nabla f(x_1), \ldots, \nabla f(x_t)\}$. Then $(x_t)_{t\in\mathbb{N}}$ is the sequence generated by Algorithm 1.*

Algorithm 1 Adaptive Heavy-Ball algorithm

Input T and $f : x \mapsto f(x) \triangleq \frac{1}{2}\langle x - x_\star,\, H(x - x_\star)\rangle + f_\star$
Initialize $x_0 \in \mathbb{R}^d$, $m_0 = 0$
for $t = 0 \cdots T-1$ **do**

$$h_t = \frac{2(f(x_t)-f_\star)}{\|\nabla f(x_t)\|^2}$$
$$x_{t+1} = x_t - (1+m_t) \times h_t \nabla f(x_t) + m_t(x_t - x_{t-1})$$
$$m_{t+1} = \frac{-(f(x_{t+1})-f_\star)\langle \nabla f(x_{t+1}), \nabla f(x_t)\rangle}{(f(x_t)-f_\star)\|\nabla f(x_{t+1})\|^2 + (f(x_{t+1})-f_\star)\langle \nabla f(x_{t+1}), \nabla f(x_t)\rangle}$$

end
Result: x_T



(a) Comparison of distances to optimum

(b) Comparison of excess losses

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. **Gradient descent with constant step-size** is the Gradient descent method tuned with the constant step-size $\gamma = \frac{2}{L+\mu}$ while **Gradient descent with Polyak step-size** and **Gradient descent with variant of Polyak step-size** refer respectively to the Gradient descent method tuned with the step-size $\gamma_t = \frac{(f(x_t)-f_\star)}{\|\nabla f(x_t)\|^2}$ as well as the step-size $\gamma_t = \frac{2(f(x_t)-f_\star)}{\|\nabla f(x_t)\|^2}$. **Heavy-ball with constant tuning** is the Heavy-ball method tuned with constant parameters $\gamma_t = \left(\frac{2}{\sqrt{L}+\sqrt{\mu}}\right)^2$ and $m_t = \left(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^2$ while **Heavy-ball with Polyak step-size based tuning** refers to Algorithm 1.

Theorem 1.1 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

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

are equivalently written in terms of an adaptive heavy-ball iteration. In particular, eq. (3) corresponds to eq. (Q-minimization) with $Q(x) = 1$. Similarly, classical conjugate gradient method corresponds to eq. (Q-minimization) with $Q(x) = x$ (this fact is quite famous, see, e.g., [18]). 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 eq. (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; $\operatorname{Sp} H$ 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 eq. (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 in 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 optimum $\|x_T - x_\star\|$, 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 in finding the guarantees of a method that hold for each and every functions of a given class, as for instance the class of L-smooth μ-strongly convex quadratic functions described as quadratic functions verifying $\mu I \preceq H \preceq LI$ for given $0 < \mu \leq L$. The **Gradient descent** method characterized by the update

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

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, Golub and Varga [8] 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 ($m_t = (\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}})^2 \frac{1+(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}})^{2(t-1)}}{1+(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}})^{2(t+1)}}$, $\gamma_t = \frac{2}{L+\mu}(1+m_t)$), is **worst-case optimal** on this class of function, achieving the guarantee $\|x_t - x_\star\| = O((\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}})^t)$ (often referred to as "acceleration"). Methods based on this two-term recursion are called "Heavy-ball" or "Polyak momentum" [20]. In particular, the stationary regime of the **Chebyshev method** is the Heavy-ball method tuned with $m_t = (\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}})^2$ and $\gamma_t = \frac{2}{L+\mu}(1+m_t) = (\frac{2}{\sqrt{L}+\sqrt{\mu}})^2$ and achieves the worst-case guarantee $\|x_t - x_\star\| = O(t(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}})^t)$, close to the optimal one achieved by the **Chebyshev method**. Note that eq. (8) is another formulation of eq. (4) where $\gamma_t = (1+m_t) \times h_t$. In all the aforementioned tuning, h_t has the same value: $h_t = \frac{2}{L+\mu}$ (see Section 3 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 $(\gamma_i^{(t)})_{i \in [\![0,t-1]\!]}$ as suggested in [14, 15]. Note that the recursion eq. (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 implicitely used in Golub and Varga [8] and formally stated, e.g., in [9, Proposition 4.1]).

Lemma 1.2 . *Let $f \in \mathcal{C}_\Lambda$. The iterates x_t satisfy*

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

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., d'Aspremont et al. [6, Chapter 1] or the blog post by Pedregosa [19] 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 7]. More recently, Goujaud et al. [9] used it to derive a method which 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 [3].

Adaptive methods. In Lemma 1.2, 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 depends 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 $\|x_{t+1} - x_\star\|^2 \leq \|x_t - x_\star\|^2 - 2\gamma_t(f - f_\star) + \gamma_t^2\|\nabla f(x_t)\|^2$. [21] 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 [10, 12].

Other variants of the latter have been proposed. For instance [1, 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 perform very well in practice (See Figure 1 and [1, 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 method of the form of eq. (9) have been found with predetermined parameters, it would be better to find a method under the form of eq. (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 gradients 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 of the form of eq. (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 eq. (8) with only parameters γ_t and m_t changing from one method to another. Finally, we show (see Theorem 1.1) 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 is using 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 [21]. 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. 5, 12] argue that this dependency is not a practical issue for many problems which we can assume verify $f_\star = 0$ (see Appendix B). A few variants of the Polyak step-size strategy were proposed by Barré et al. [1], including a version incorporating a Nesterov-type momentum [16], 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 [17]. In [12], the authors study the stochastic Polyak step-size, whereas [5] applies it to Mirror descent.

Many alternative adaptive methods have been proposed in the past. Among them, let us mention [2] which introduced the so-called Barzilai-Borwein step-size rule, and the more recent [13] which developed a step-size policy that adapts to the local geometry with convergence guarantees beyond quadratic minimization.

2. Main theorem

This section states and proves Theorem 1.1. In short, given a certain function f (characterized by H and $x_\star$ here) and a starting point x_0, we search for an iterative procedure, possibly adaptive, verifying the polynomial-based expression eq. (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).

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

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

Theorem 2.1 (Main) *The unique solution to eq.* (Q-minimization) *is given by the heavy-ball procedure*

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

where

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

Remark that setting $Q(X) = X$ leads to a nice expression of the conjugate gradients method. 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 an Hessian vector product) to proceed. This is also a priori the case for all other choices of $Q(\cdot)$. In the case of $Q(X) = 1$, which corresponds to minimizing the distance to the optimum (see eq. (3)), we can use an alternate writing making use of $f_\star$:

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

3. Concluding remarks and discussion

We highlight a few remarkable points:

1. **Possible extensions beyond quadratics.** While our result only applies to quadratic functions, we believe it is a necessary first step toward extending momentum-based Polyak step-sizes on more general smooth strongly convex functions.

2. **Relationship and comparison to conjugate gradients.** The conjugate gradient method is an instance-optimal algorithm minimizing function values at each iteration. Similarly, the main method discussed in this work is instance-optimal for minimizing the distance to the optimum. Perhaps interestingly, the two methods appeared to behave similarly in numerical experiments.

   The main difference between the two methods is the knowledge of $f_\star$ (which is required for our proposed method), thereby limiting its practical use when the latter is not known. On the other hand, the method can be tested beyond quadratics without any cost, which we leave for future work, while the conjugate gradient method is approximated using an hessian vector product when applied in a non quadratic setting.

3. **Step-size parametrization.** While the γ_t plays a different role in eq. (8) and eq. (7), they both usually are called "step-size" by default. But we noticed that both in 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 step-size of the Gradient descent method. In section 1, we notice that again, h_t's value is the one that minimizing the Gradient descent method after a single step. For this reason, we believe that the natural parametrization of the Heavy-ball methods should be $x_{t+1} = x_t - (1 + m_t) \times h_t \nabla f(x_t) + m_t(x_t - x_{t-1})$ and that h_t should be referred to as the step-size. We believe that the fact h_t plays this role instead of γ_t is an important remark. 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.

References

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

[2] Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.

[3] Raphaël Berthier, Francis Bach, and Pierre Gaillard. Accelerated gossip in networks of given dimension using jacobi polynomial iterations. *SIAM Journal on Mathematics of Data Science*, 2(1):24–47, 2020.

[4] Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2003.

[5] Ryan D'Orazio, Nicolas Loizou, Issam Laradji, and Ioannis Mitliagkas. Stochastic mirror descent: Convergence analysis and adaptive variants via the mirror stochastic Polyak stepsize. *arXiv preprint arXiv:2110.15412*, 2021.

[6] Alexandre d'Aspremont, Damien Scieur, Adrien Taylor, et al. Acceleration methods. *Foundations and Trends® in Optimization*, 5(1-2):1–245, 2021.

[7] Bernd Fischer. *Polynomial based iteration methods for symmetric linear systems*. SIAM, 2011.

[8] Gene H Golub and Richard S Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order richardson iterative methods. *Numerische Mathematik*, 3(1):157–168, 1961.

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

[10] Robert M Gower, Mathieu Blondel, Nidham Gazagnadou, and Fabian Pedregosa. Cutting some slack for SGD with Adaptive Polyak Stepsizes. *arXiv preprint arXiv:2202.12328*, 2022.

[11] Elad Hazan and Sham Kakade. Revisiting the Polyak step size. *arXiv preprint arXiv:1905.00313*, 2019.

[12] Nicolas Loizou, Sharan Vaswani, Issam Hadj 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.

[13] Yura Malitsky and Konstantin Mishchenko. Adaptive gradient descent without descent. In *International Conference on Machine Learning (ICML)*, pages 6702–6712. PMLR, 2020.

[14] Arkadi S. Nemirovski. Information-based complexity of linear operator equations. *Journal of Complexity*, 8(2):153–175, 1992.

[15] Arkadi S. Nemirovski. Information-based complexity of convex programming. *Lecture notes,* http://www2.isye.gatech.edu/~nemirovs/Lec_EMCO.pdf, 1994.

[16] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[17] Yurii Nesterov. *Introductory lectures on convex optimization: a basic course*. Applied optimization. Kluwer Academic Publishers, 2004.

[18] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer, 1999.

[19] Fabian Pedregosa. Residual polynomials and the chebyshev method, 2020. URL http://fa.bianp.net/blog/2020/polyopt/.

[20] Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR computational mathematics and mathematical physics*, 4(5):1–17, 1964.

[21] Boris T. Polyak. Introduction to optimization. 1987. *Optimization Software, Inc, New York*, 1987.

Appendix A. Proof of main theorem

In this section, we detail the proof of the main theorem.
Proof.

Designing methods from the polynomial point of view. As suggested by Lemma 1.2, we look for an iterative method which can be expressed in the form $x_t - x_\star = P_t(H)(x_0 - x_\star)$, where P_t is a tth 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 $\lambda \in \mathrm{Sp}(H)$ its eigenvalues and by v_λ th associated orthonormal basis of eigenvectors, leading to $H = \sum_{\lambda \in \mathrm{Sp}(H)} \lambda v_\lambda v_\lambda^T$. The quantity to be minimized can now be written as:

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

with λ_{Q} the discrete measure $\sum_{\lambda \in \mathrm{Sp}(H)} Q(\lambda)\langle x_0 - x_\star, v_\lambda\rangle^2\, \delta_\lambda$ (we sometimes use the shorthand notation $\int P_t^2 \,\mathrm{d}\lambda_{\mathrm{Q}}$ for eq. (17) in what follows). It is clear that eq. (17) is 0 if and only if $P_t(\lambda) = 0$ for all $\lambda \in \mathrm{Sp}(H)$. As a consequence, we conclude that (i) choosing the right sequence of polynomials leads to convergence in exactly $|\,\mathrm{Sp}(H)|$ 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 $\mathbb{R}_{|\,\mathrm{Sp}(H)|-1}[X]$. 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 $t \leq |\,\mathrm{Sp}(H)| - 1$ 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 $t \geq |\,\mathrm{Sp}(H)|$, 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 eq. (18) (see, e.g., Berthier et al. [3] and the references therein).

From minimal norm to orthogonality. The solution to eq. (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 eq. (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 eq. (18) iff

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

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

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

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

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

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 eq. (21), one can write

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

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

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

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

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

with

$$\begin{cases} \tilde{a}_t &= \frac{\langle XP_t(X), P_t(X)\rangle_{XQ}}{\langle P_t, P_t\rangle_{XQ}}, \\ \tilde{b}_t &= \frac{\langle XP_t(X), P_{t-1}(X)\rangle_{XQ}}{\langle P_{t-1}, P_{t-1}\rangle_{XQ}}. \end{cases} \tag{23}$$

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

$$\begin{aligned} x_{t+1} - x_\star &= \frac{\tilde{a}_t(x_t - x_\star) - H(x_t - x_\star) + \tilde{b}_t(x_{t-1} - x_\star)}{\tilde{a}_t + \tilde{b}_t}, \\ &= x_t - x_\star - \frac{1}{\tilde{a}_t + \tilde{b}_t}\nabla f(x_t) + \frac{-\tilde{b}_t}{\tilde{a}_t + \tilde{b}_t}(x_t - x_{t-1}), \end{aligned}$$

thereby reaching the desired

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

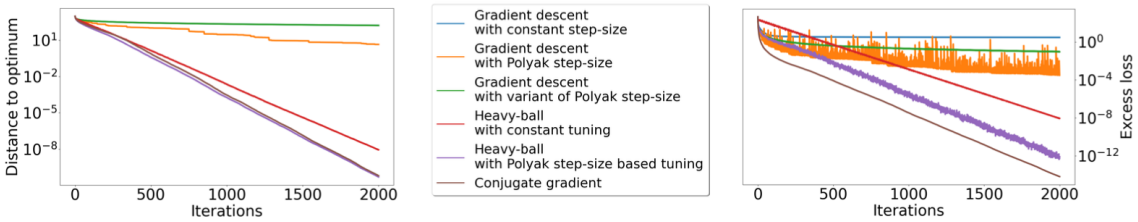
with

$$h_t = \frac{1}{\tilde{a}_t + \tilde{b}_t}, \quad \text{and} \quad m_t = \frac{-\tilde{b}_t}{\tilde{a}_t + \tilde{b}_t} \tag{25}$$

From eq. (24), we recognize an Heavy-ball method with some variable step-size h_t and momentum term m_t, thereby concluding the proof. ■

Appendix B. Additional experiments

In this section, we compare the gradient descent, the heavy-ball and the conjugate gradient method 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 difference between Figure 1 and Figure 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.



(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. **Gradient descent with constant step-size** is the Gradient descent method tuned with the constant step-size $\gamma = \frac{2}{L+\mu}$ while **Gradient descent with Polyak step-size** and **Gradient descent with variant of Polyak step-size** refer respectively to the Gradient descent method tuned with the step-size $\gamma_t = \frac{(f(x_t)-f_\star)}{\|\nabla f(x_t)\|^2}$ as well as the step-size $\gamma_t = \frac{2(f(x_t)-f_\star)}{\|\nabla f(x_t)\|^2}$. **Heavy-ball with constant tuning** is the Heavy-ball method tuned with constant parameters $\gamma = \left(\frac{2}{\sqrt{L}+\sqrt{\mu}}\right)^2$ and $m = \left(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^2$ while **Heavy-ball with Polyak step-size based tuning** refers to Algorithm 1.