Super-Acceleration with Cyclical Step-sizes

Baptiste Goujaud
CMAP, École Polytechnique
Institut Polytechnique de Paris

Damien Scieur
Samsung SAIL Montreal

Aymeric Dieuleveut
CMAP, École Polytechnique
Institut Polytechnique de Paris

Adrien Taylor
INRIA, École Normale Supérieure
CNRS, PSL Research University, Paris

Fabian Pedregosa
Google Research

Abstract

We develop a convergence-rate analysis of momentum with cyclical step-sizes. We show that under some assumption on the spectral gap of Hessians in machine learning, cyclical step-sizes are provably faster than constant step-sizes. More precisely, we develop a convergence rate analysis for quadratic objectives that provides optimal parameters and shows that cyclical learning rates can improve upon traditional lower complexity bounds. We further propose a systematic approach to design optimal first order methods for quadratic minimization with a given spectral structure. Finally, we provide a local convergence rate analysis beyond quadratic minimization for the proposed methods and illustrate our findings through benchmarks on least squares and logistic regression problems.

1 Introduction

One of the most iconic methods in first order optimization is gradient descent with momentum, also known as the heavy ball method (Polyak, 1964). This method enjoys widespread popularity both in its original formulation and in a stochastic variant that replaces the gradient by a stochastic estimate, a method that is behind many of the recent breakthroughs in deep learning (Sutskever et al., 2013).

A variant of the stochastic heavy ball where the step-sizes are chosen in *cyclical* order has recently come to the forefront of machine learning research, showing state-of-the art results on different deep learning benchmarks (Loshchilov and Hutter, 2017; Smith, 2017). Inspired by this empirical success, we aim to study the convergence of the heavy ball algorithm where step-sizes $h_0, h_1, \ldots$ are not fixed or decreasing but instead chosen in cyclical order, as in Algorithm 1.

Algorithm 1:
Cyclical heavy ball $\mathrm{HB}_K(h_0, \ldots, h_{K-1}; m)$

Input: Initialization x_0, momentum $m \in (0, 1)$, step-sizes $\{h_0, \ldots, h_{K-1}\}$
$x_1 = x_0 - \frac{h_0}{1+m}\nabla f(x_0)$
for $t = 1, 2, \ldots$ **do**
  $x_{t+1} = x_t - h_{\mathrm{mod}(t,K)}\nabla f(x_t) + m(x_t - x_{t-1})$
end

The heavy ball method with constant step-sizes enjoys a mature theory, where it is known for example to achieve optimal black-box worst-case complexity of quadratic convex optimization (Nemirovsky, 1992). In stark contrast, little is known about the the convergence of the above variant with cyclical step-sizes. Our main motivating question is

> Do cyclical step-sizes improve convergence of heavy ball?

Our **main contribution** provides a positive answer to this question and, more importantly, *quantifies* the speedup under different assumptions. In particular, we show that for quadratic problems, whenever Hessian's spectrum belongs to two or more disjoint intervals, the heavy ball method with cyclical step-sizes achieves a faster worst-case convergence rate. Recent works have shown that this assumption on the spectrum is quite natural and occurs in many machine learning

Proceedings of the 25$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

problems, including deep neural networks (Sagun et al., 2017; Papyan, 2018; Ghorbani et al., 2019; Papyan, 2019). The concurrent work of Oymak (2021) analyzes gradient descent (without momentum, see extended comparison in Appendix G) under this assumption. More precisely, we list our main contributions below.

- In sections 3 and 4, we provide a **tight convergence rate analysis** of the cyclical heavy ball method (Theorems 3.1 and 3.2 for two step-sizes, and Theorem 4.8 for the general case). This analysis highlights a regime under which this method achieves a faster worst-case rate than the accelerated rate of heavy ball, a phenomenon we refer to as *super-acceleration*. Theorem 5.1 extends the (local) convergence rate analysis results to non-quadratic objectives.
- As a byproduct of the convergence-rate analysis, we obtain an explicit expression for the **optimal parameters** in in the case of cycles of length two (Algorithm 2) and an implicit expression in terms of a system of K equations in the general case.
- Section 6 presents **numerical benchmarks** illustrating the improved convergence of the cyclical approach on 4 problems involving quadratic and logistic losses on both synthetic and a handwritten digits recognition dataset.
- Finally, we conclude in Section 7 with a discussion of this work's **limitations**.

2 Notation and Problem Setting

Throughout the paper (except in Section 5), we consider the problem of minimizing a quadratic function:

$$\min_{x\in\mathbb{R}^d} f(x)\,, \text{ with } \quad f\in\mathcal{C}_\Lambda, \tag{OPT}$$

where $\mathcal{C}_\Lambda$ is the class of quadratic functions with Hessian matrix H and whose Hessian spectrum $\mathrm{Sp}(H)$ is localized in $\Lambda \subseteq [\mu, L] \subseteq \mathbb{R}_{>0}$:

$$\mathcal{C}_\Lambda \triangleq \left\{ f(x) = (x-x_*)^\top \tfrac{H}{2}(x-x_*) + f_*,\, \mathrm{Sp}(H)\subseteq\Lambda \right\}$$

The condition $\Lambda \subseteq [\mu, L]$ implies all quadratic functions under consideration are L-smooth and μ-strongly convex. For this function class, we define κ, the (inverse) condition number, and ρ, the ratio between the center of Λ and its radius, as

$$\kappa \triangleq \frac{\mu}{L}, \qquad \rho \triangleq \frac{L+\mu}{L-\mu} \quad = \left(\frac{1+\kappa}{1-\kappa}\right). \tag{1}$$

Finally, for a method solving (OPT) that generates a sequence of iterates $\{x_t\}$, we define its worst-case rate r_t and its asymptotic rate factor τ as

$$r_t \triangleq \sup_{x_0\in\mathbb{R}^d,\, f\in\mathcal{C}_\Lambda} \frac{\|x_t - x_*\|}{\|x_0 - x_*\|},\; 1-\tau \triangleq \limsup_{t\to\infty} \sqrt[t]{r_t}\,. \tag{2}$$

3 Super-acceleration with Cyclical Step-sizes

In this section we develop one of our main contributions, a convergence rate analysis of the cyclical heavy ball method with cycles of length 2. This analysis crucially depends on the location of the Hessian's eigenvalues; we assume that these are contained in a set Λ that is the union of 2 intervals *of the same size*

$$\Lambda = [\mu_1,\, L_1] \cup [\mu_2,\, L_2]\,,\; L_1 - \mu_1 = L_2 - \mu_2\,. \tag{3}$$

By symmetry, this set is alternatively described by

$$\mu \triangleq \mu_1, \quad L \triangleq L_2 \quad \text{and} \quad R \triangleq \frac{\mu_2 - L_1}{L_2 - \mu_1}\,, \tag{4}$$

where R is the relative length of the gap $\mu_2 - L_1$ with respect to the diameter $L_2 - \mu_1$ (see Figure 1). This parametrization is convenient since the relative gap plays a crucial role in our convergence analysis. Our results allow $R = 0$, therefore recovering the classical setting of Hessian eigenvalues contained in an interval.

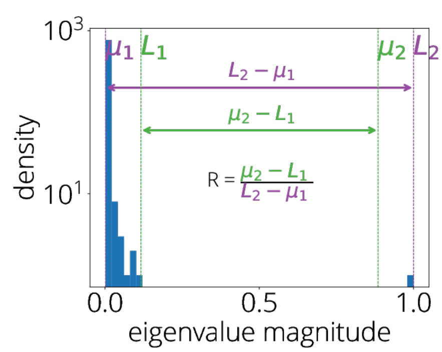

Figure 1: Hessian eigenvalue histogram for a quadratic objective on MNIST. The outlier eigenvalue at L_2 generates a non-zero relative gap $R = 0.77$. In this case, the 2-cycle heavy ball method has a faster asymptotic rate than the single-cycle one (see Section 3.2).

Through a correspondence between optimization methods and polynomials (see Section 4), we can derive a worst-case analysis for the cyclical heavy ball method. The outcome of this analysis is in the following theorem, that provides the asymptotic convergence rate of Algorithm 1 for cycles of length two. All proofs of results in this section can be found in Appendix D.3.

Theorem 3.1 (Rate factor of $\mathrm{HB}_2(h_0, h_1; m)$)**.** *Let $f \in \mathcal{C}_\Lambda$ and consider the cyclical heavy ball method with*

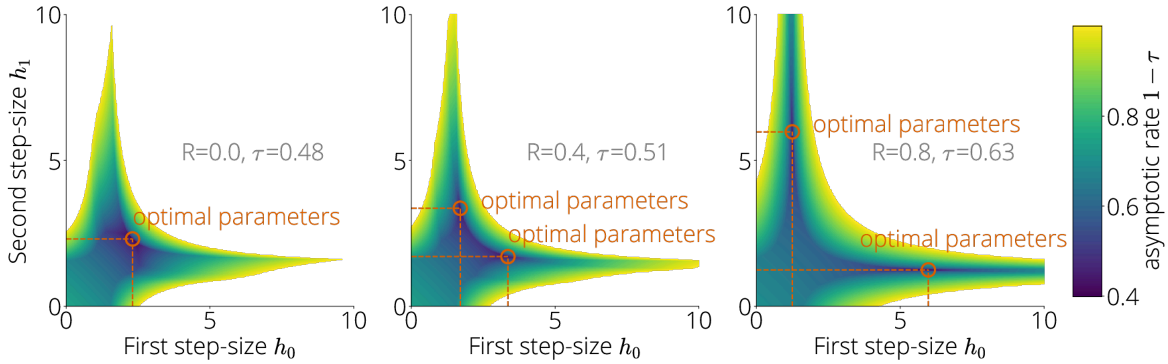

Figure 2: **Asymptotic rate of cyclical $(K = 2)$ heavy ball** in terms of its step-sizes h_0, h_1 across 3 different values of the relative gap R. In the **left** plot, the relative gap is zero, and so the step-sizes with smallest rate coincide $(h_0 = h_1)$. For non-zero values of R (**center and right**), the optimal method instead alternates between two *different* step-sizes. In all plots the momentum parameter m is set according to Algorithm 2.

step-sizes h_0, h_1 and momentum parameter m. The asymptotic rate factor of Algorithm 1 with cycles of length two is

$$1-\tau = \begin{cases} \sqrt{m} & \text{if } \sigma_* \leq 1, \\ \sqrt{m}\left(\sigma_* + \sqrt{\sigma_*^2 - 1}\right)^{\frac{1}{2}} & \text{if } \sigma_* \in \left(1, \frac{1+m^2}{2m}\right), \\ \geq 1 \text{ (no convergence)} & \text{if } \sigma_* \geq \frac{1+m^2}{2m}, \end{cases}$$

$$\text{with} \quad \sigma_* = \max_{\lambda \in \left\{\mu_1, L_1, \mu_2, L_2, (1+m)\frac{h_0+h_1}{2h_0h_1}\right\} \cap \Lambda} |\sigma_2(\lambda)|$$

$$\text{and} \quad \sigma_2(\lambda) = 2\left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)\left(\frac{1+m-\lambda h_1}{2\sqrt{m}}\right) - 1\,.$$

3.1 Optimal algorithm

The previous theorem gives the convergence rate for all triplets (h_0, h_1, m). This allows us for instance to map out the associated convergence rate for every pair of step-sizes. As we illustrate in Figure 2, as we increase the relative gap (R), the optimal step-sizes become further apart.

Another application of the previous theorem is to find the parameters that minimize the asymptotic convergence rate. Although the process rather tedious and relegated to Appendix D.3, the resulting momentum (m) and step-size parameters (h_0, h_1) are remarkably simple, and given by the expressions

$$m = \left(\frac{\sqrt{\rho^2 - R^2} - \sqrt{\rho^2 - 1}}{\sqrt{1 - R^2}}\right)^2 \tag{5}$$

$$h_0 = \frac{1+m}{L_1} \qquad h_1 = \frac{1+m}{\mu_2}\,. \tag{6}$$

Being one of our main contributions, this algorithm is also described in pseudocode in Algorithm 2. By construction, this method has an *asymptotically optimal* convergence rate which we detail in the next Corollary:

Algorithm 2: Cyclical $(K = 2)$ heavy ball with optimal parameters

---

Input: Initial iterate x_0, $\mu_1 < L_1 \leq \mu_2 < L_2$ (where $L_1 - \mu_1 = L_2 - \mu_2$, $\mu_1 < L_2$)
Set: $\rho = \frac{L_2+\mu_1}{L_2-\mu_1}$, $R = \frac{\mu_2 - L_1}{L_2 - \mu_1}$,
$m = \left(\frac{\sqrt{\rho^2-R^2}-\sqrt{\rho^2-1}}{\sqrt{1-R^2}}\right)^2$
$x_1 = x_0 - \frac{1}{L_1}\nabla f(x_0)$

for $t = 1, 2, \ldots$ **do**
  $h_t = \frac{1+m}{L_1}$ (if t is even), $h_t = \frac{1+m}{\mu_2}$ (if t is odd)
  $x_{t+1} = x_t - h_t \nabla f(x_t) + m(x_t - x_{t-1})$
end

Corollary 3.2. *The non-asymptotic and asymptotic worst-case rates $r_t^{Alg.\ 2}$ and $1 - \tau^{Alg.\ 2}$ of Algorithm 2 over $\mathcal{C}_\Lambda$ for even iteration number t are*

$$r_t^{Alg.\ 2} = \left(\frac{\sqrt{\rho^2-R^2}-\sqrt{\rho^2-1}}{\sqrt{1-R^2}}\right)^t \left(1 + t\sqrt{\frac{\rho^2-1}{\rho^2-R^2}}\right),$$

$$1 - \tau^{Alg.\ 2} = \frac{\sqrt{\rho^2 - R^2} - \sqrt{\rho^2 - 1}}{\sqrt{1 - R^2}}\,.$$

Note that this result also holds if we swap the 2 step-sizes in Algorithm 2.

Eigengap and accelerated cyclical step-sizes While Corollary 3.2 focuses on the optimal tuning of Algorithm 2, Theorem D.1 provides general convergence analysis for non-optimal parameters. In the case of existence of an eigengap, a range of cyclical step-sizes leads to an accelerated rate of convergence (compared to the optimal constant step-size strategy) and therefore, an inexact parameters search can lead to such an acceleration.

3.2 Comparison with Polyak Heavy Ball

In the absence of eigenvalue gap ($R = 0$ and $\Lambda = [\mu, L]$), Algorithm 2 reduces to Polyak heavy ball (PHB) (Polyak, 1964), whose worst-case rate is detailed in Appendix B. Since the asymptotic rate of Algorithm 2 is monotonically decreasing in R, the convergence rate of the cyclical variant is always better than PHB. Furthermore, in the ill-conditioned regime (small κ), the comparison is particularly simple: the optimal 2-cycle algorithm has a $\sqrt{1-R^2}$ relative improvement over PHB, as provided by the next proposition. A more thorough comparison for different support sets Λ is discussed in Table 1.

Proposition 3.3. *Let $R \in [0, 1)$. The rate factors of respectively Algorithm 2 and PHB verify*

$$1 - \tau^{Alg.\ 2} \underset{\kappa \to 0}{=} 1 - \tfrac{2\sqrt{\kappa}}{\sqrt{1-R^2}} + o(\sqrt{\kappa}), \tag{7}$$
$$1 - \tau^{PHB} \underset{\kappa \to 0}{=} 1 - 2\sqrt{\kappa} + o(\sqrt{\kappa}).$$

4 A constructive Approach: Minimax Polynomials

This section presents a generic framework that allows designing optimal momentum and step-size cycles for given sets Λ and cycle length K.

We first recall classical results that link optimal first order methods on quadratics and Chebyshev polynomials. Then, we generalize the approach by showing that optimal methods can be viewed as combinations of Chebyshev polynomials, and minimax polynomials σ_K^Λ of degree K over the set Λ. Finally, we show how to recover the step-size schedule from σ_K^Λ and present the general algorithm (Algorithm 3).

4.1 First Order Methods on Quadratics and Polynomials

A key property that we will use extensively in the analysis is the following link between first order methods and polynomials.

Proposition 4.1. *Let $f \in \mathcal{C}_\Lambda$. The iterates x_t satisfy*

$$x_{t+1} \in x_0 + \operatorname{span}\{\nabla f(x_0), \ldots, \nabla f(x_t)\}, \tag{8}$$

where x_0 is the initial approximation of x_*, 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

$$\forall\, t \quad x_t - x_* = P_t(H)(x_0 - x_*), \quad P_t(0) = 1. \tag{9}$$

Example 4.2 (Gradient descent)**.** Consider the gradient descent algorithm with fixed step-size h, applied to problem (OPT). Then, after unrolling the update, we have

$$\begin{aligned} x_{t+1} - x_* &= x_t - x_* - h\nabla f(x_t) \\ &= x_t - x_* - hH(x_t - x_*) \\ &= (I - hH)^{t+1}(x_0 - x_*). \end{aligned} \tag{10}$$

In this case, the polynomial associated to gradient descent is $P_t(\lambda) = (1 - h\lambda)^t$.

The above proposition can be used to obtain worst-case rates for first order methods by bounding their associated polynomials. Indeed, using the Cauchy-Schwartz inequality in (9) leads to

$$\|x_t - x_*\| \le \sup_{\lambda\in\Lambda} |P_t(\lambda)|\, \|x_0 - x_*\|$$
$$\implies \quad r_t = \sup_{\lambda\in\Lambda} |P_t(\lambda)|, \quad \text{where } P_t(0) = 1. \tag{11}$$

Therefore, finding the algorithm with the fastest worst-case rate can be equivalently framed as the problem of finding the polynomial with smallest value on the eigenvalue support Λ, subject to the normalization condition $P_t(0) = 1$. Such polynomials are referred to as **minimax**. Throughout the paper, we use this polynomial-based approach to find methods with optimal rates.

An important property of minimax polynomials is their *equioscillation* on Λ (see Theorem C.1 and its proof for a formal statement).

Definition 4.3. (Equioscillation) A polynomial P_t of degree t equioscillates on Λ if it verifies $P_t(0) = 1$ and there exist $\lambda_0 < \lambda_1 < \ldots < \lambda_t \in \Lambda$ such that

$$P_t(\lambda_i) = (-1)^i \max_{\lambda\in\Lambda} |P_t(\Lambda)|. \tag{12}$$

Example 4.4 (Λ is an interval)**.** The t-th order Chebyshev polynomials of the first kind T_t satisfy the *equioscillation* property on $[-1, 1]$. It follows that minimax polynomials on $\Lambda = [\mu, L]$ can be obtained by composing the Chebyshev polynomial T_t with the linear transformation σ_1^Λ:

$$\frac{T_t\left(\sigma_1^\Lambda(\lambda)\right)}{T_t\left(\sigma_1^\Lambda(0)\right)} = \underset{P\in\mathbb{R}_t[X], P(0)=1}{\arg\min} \ \sup_{\lambda\in\Lambda} |P(\lambda)|, \tag{13}$$
$$\text{with} \ \ \sigma_1^\Lambda(\lambda) = \frac{L+\mu}{L-\mu} - \frac{2}{L-\mu}\lambda,$$

where σ_1^Λ maps the interval $[\mu, L]$ to $[-1, 1]$. The optimization method associated with this minimax polynomial is the Chebyshev semi-terative method (Flanders and Shortley, 1950; Golub and Varga, 1961), described

Relative gap R	Set Λ	Rate factor τ	Speedup τ/τ^{PHB}
$R \in [0,1)$	$[\mu, \mu + \frac{1-R}{2}(L-\mu)] \cup [L - \frac{1-R}{2}(L-\mu), L]$	$\frac{2\sqrt{\kappa}}{\sqrt{1-R^2}}$	$(1-R^2)^{-\frac{1}{2}}$
$R = 1 - \sqrt{\kappa}/2$	$[\mu, \mu + \frac{\sqrt{\mu L}}{4}] \cup [L - \frac{\sqrt{\mu L}}{4}, L]$	$2\sqrt[4]{\kappa}$	$\kappa^{-\frac{1}{4}}$
$R = 1 - 2\gamma\kappa$	$[\mu, (1+\gamma)\mu] \cup [L - \gamma\mu, L]$	indep. of κ	$O(\kappa^{-\frac{1}{2}})$

Table 1: Case study of the convergence of Algorithm 2 as a function of R, in the regime $\kappa \to 0$. The **first line** corresponds to the regime where R is independent of κ, and we observe a constant gain w.r.t. PHB. The **second line** considers a setting in which R depends on $\sqrt{\kappa}$, that is, the two intervals in Λ are relatively small. The asymptotic rate reads $(1 - 2\sqrt[4]{\kappa})^t$, improving over the $(1-2\sqrt{\kappa})^t$ rate of Polyak Heavy ball, unimprovable when $R = 0$. Finally, in the **third line**, R depends on κ, the two intervals in Λ are so small that the convergence becomes $O(1)$, i.e., is independent of κ.

also in Appendix B.1. This method achieves the lower complexity bound for smooth strongly convex quadratic minimization (Nemirovsky, 1995, Chapter 12) or (Nemirovsky, 1992; Nesterov, 2003).

The next proposition provides the main results in this subsection, which is key for obtaining Algorithm 2. It characterizes the even degree minimax polynomial in the setting of Section 3, that is, when Λ is the union of 2 intervals of same size. In this case, the minimax solution is also based on Chebyshev polynomials, but composed with a degree-two polynomial σ_2^Λ.

Proposition 4.5. *Let $\Lambda = [\boldsymbol{\mu}_1, \boldsymbol{L}_1] \cup [\boldsymbol{\mu}_2, \boldsymbol{L}_2]$ be an union of two intervals of the same size ($\boldsymbol{L}_1 - \boldsymbol{\mu}_1 = \boldsymbol{L}_2 - \boldsymbol{\mu}_2$) and let m, h_0, h_1 be as defined in Algorithm 2. Then the minimax polynomial (solution to (12)) is, for all $t = 2n$, $n \in \mathbb{N}_0^+$,*

$$\frac{T_n\left(\sigma_2^\Lambda(\lambda)\right)}{T_n\left(\sigma_2^\Lambda(0)\right)} = \underset{\substack{P \in \mathbb{R}_t[X],\\ P(0)=1}}{\arg\min} \sup_{\lambda \in \Lambda} |P(\lambda)|\,,$$

$$\text{with} \quad \sigma_2^\Lambda(\lambda) = \frac{1}{2m}\left(1 + m - \lambda h_0\right)\left(1 + m - \lambda h_1\right) - 1\,.$$

4.2 Generalization to Longer Cycles

The polynomial in Example 4.4 uses a linear link function σ_1^Λ to map Λ to $[-1, 1]$. In Proposition 4.5, we see that a degree *two* link function σ_2^Λ can be used to find the minimax polynomial when Λ is the union of two intervals. This section generalizes this approach and considers higher-order polynomials for σ_K.

We start with the following parametrization, with an arbitrary polynomial σ_K of degree K,

$$P_t(\lambda; \sigma_K) \triangleq \frac{T_n\left(\sigma_K(\lambda)\right)}{T_n\left(\sigma_K(0)\right)}, \quad \forall t = Kn,\ n \in \mathbb{N}_0^+\,. \quad (14)$$

As we will see in the next subsection, this parametrization allows considering cycles of step-sizes. Our goal now is to find the σ_K that obtains the fastest convergence rate possible. The next proposition quantifies its impact on the asymptotic rate and its proof can be found in Appendix D.1.

Proposition 4.6. *For a given σ_K such that $\sup_{\lambda \in \Lambda} |\sigma_K(\lambda)| = 1$, the asymptotic rate factor τ^{σ_K} of the method associated to the polynomial (14) is*

$$1 - \tau^{\sigma_K} = \lim_{t\to\infty} \sqrt[t]{\sup_{\lambda\in\Lambda} |P_t(\lambda;\sigma_K)|} = \left(\sigma_0 - \sqrt{\sigma_0^2 - 1}\right)^{\frac{1}{K}}\,,$$

$$\text{with } \sigma_0 \triangleq \sigma_K(0)\,. \quad (15)$$

For a fixed K, the asymptotic rate (15) is a decreasing function of σ_0. This motivates the introduction of the "optimal" degree K polynomial σ_K^Λ as the one that solves

$$\sigma_K^\Lambda \triangleq \underset{\sigma \in \mathbb{R}_K[X]}{\arg\max}\, \sigma(0) \quad \text{s.t.} \quad \sup_{\lambda \in \Lambda} |\sigma(\lambda)| \le 1\,. \quad (16)$$

Using the above definition, we recover the σ_1^Λ and σ_2^Λ from Example 4.4 and Proposition 4.5.

Finding the polynomial. Finding an exact and explicit solution for the general K and Λ case is unfortunately out of reach, as it involves solving a system of K non-linear equations. Here we describe an approximate approach. Let $\sigma_K^\Lambda(x) = \sum_{i=0}^K \sigma_i x^i$. We propose to discretize Λ into N different points $\{\lambda_j\}$, then solve the linear problem

$$\max_{\sigma_i} \sigma_0 \quad \text{s.t.} \quad -1 \le \textstyle\sum_{i=0}^K \sigma_i \lambda_j^i \le 1, \quad \forall j = 1, \ldots, N\,. \quad (17)$$

To check the optimality, it suffices to verify that the polynomial σ_K^Λ satisfies the *equioscillation* property (Definition 4.3), as depicted in Figure 3.

Remark 4.7 (Relationship between optimal and minimax polynomials)**.** For later reference, we note that the optimal polynomial σ_K^Λ is equivalent to finding a minimax polynomial on Λ and to rescale it. More precisely, σ_K^Λ is optimal if and only if $\sigma_K^\Lambda/\sigma_K^\Lambda(0)$ is minimax.

4.3 Cyclical Heavy Ball and (Non-)asymptotic Rates of Convergence

We now describe the link between σ_K^Λ and Algorithm 3. Using the recurrence for Chebyshev polynomials of the first kind in (14), we have $\forall t = Kn,\ n \in \mathbb{N}_0^+$,

$$\frac{T_{n+1}(\sigma_K^\Lambda(\lambda))}{T_{n+1}(\sigma_K^\Lambda(0))} = 2\sigma_K^\Lambda(\lambda)\left[\frac{T_n(\sigma_K^\Lambda(\lambda))}{T_n(\sigma_K^\Lambda(0))}\right]\underbrace{\left[\frac{T_n(\sigma_K^\Lambda(0))}{T_{n+1}(\sigma_K^\Lambda(0))}\right]}_{=a_n} - \left[\frac{T_{n-1}(\sigma_K^\Lambda(\lambda))}{T_{n-1}(\sigma_K^\Lambda(0))}\right]\underbrace{\left[\frac{T_{n-1}(\sigma_K^\Lambda(0))}{T_{n+1}(\sigma_K^\Lambda(0))}\right]}_{=b_n}.$$

It still remains to find an algorithm associated with this polynomial. To obtain one in the form of Algorithm 1, one can use the stationary behavior of the recurrence. From (Scieur and Pedregosa, 2020), the coefficients a_n and b_n converge as $n \to \infty$ to their fixed-points a_∞ and b_∞. We therefore consider here an asymptotic polynomial $\bar{P}_t(\lambda;\sigma_K^\Lambda)$, whose recurrence satisfies

$$\bar{P}_t(\lambda;\sigma_K^\Lambda) = 2a_\infty\sigma_K^\Lambda(\lambda)\bar{P}_{t-K}(\lambda;\sigma_K^\Lambda) - b_\infty\bar{P}_{t-2K}(\lambda;\sigma_K^\Lambda)\,. \tag{18}$$

Similarly to $K = 1$, where this limit recursion corresponds to PHB, this recursion corresponds to an instance of Algorithm 3 (see Proposition 4.9 below), further motivating the cyclical heavy ball algorithm.

The following theorem is the main result of this section and characterizes the convergence rate of Algorithm 1 for arbitrary momentum and step-size sequence $\{h_i\}_{i\in[\![1,K]\!]}$.

Theorem 4.8. *With an arbitrary momentum m and an arbitrary sequence of step-sizes $\{h_i\}$, the worst-case rate of convergence $1-\tau$ of Algorithm 1 on $\mathcal{C}_\Lambda$ is*

$$\begin{cases} \sqrt{m} & \text{if } \sigma_* \le 1 \\ \sqrt{m}\left(\sigma_* + \sqrt{\sigma_*^2 - 1}\right)^{K^{-1}} & \text{if } \sigma_* \in \left(1,\ \dfrac{1+m^K}{2\left(\sqrt{m}\right)^K}\right) \\ \ge 1\,(\textit{no convergence}) & \text{if } \sigma_* \ge \dfrac{1+m^K}{2\left(\sqrt{m}\right)^K}\,, \end{cases} \tag{19}$$

where $\sigma_* \triangleq \sup_{\lambda\in\Lambda}|\sigma(\lambda;\{h_i\},m)|$, $\sigma(\lambda;\{h_i\},m)$ *is the K-degree polynomial*

$$\sigma(\lambda;\{h_i\},m) \triangleq \frac{1}{2}\mathrm{Tr}\left(M_1M_2\dots M_K\right), \tag{20}$$

and $M_i = \begin{bmatrix} \frac{1+m-h_{K-i}\lambda}{\sqrt{m}} & -1 \\ 1 & 0 \end{bmatrix}$.

By optimizing over these parameters, we obtain the Algorithm 3, a method associated to (18), whose rate is described in Proposition 4.9. All proofs can be found in Appendix D.2.

Algorithm 3: Cyclical (arbitrary K) heavy ball with optimal parameters

---

Input: Eigenvalue localization Λ, cycle length K, initialization x_0.

Preprocessing:

1. Find the polynomial σ_K^Λ such that it satisfies (16).
2. Set step-sizes $\{h_i\}_{i=0,\dots,K-1}$ and momentum m that satisfy resp. equations (21) and (22).

Set $x_1 = x_0 - \frac{h_0}{1+m}\nabla f(x_0)$
for $t = 1, 2, \dots$ **do**
  $x_{t+1} = x_t - h_{\mathrm{mod}(t,K)}\nabla f(x_t) + m(x_t - x_{t-1})$
end

Proposition 4.9. *Let $\sigma(\lambda;\{h_i\},m)$ be the polynomial defined by (20), and σ_K^Λ be the optimal link function of degree K defined by (16). If the momentum m and the sequence of step-sizes $\{h_i\}$ satisfy*

$$\sigma(\lambda;\{h_i\},m) = \sigma_K^\Lambda(\lambda)\,, \tag{21}$$

*then **1)** the parameters are optimal, in the sense that they minimize the asymptotic rate factor from Theorem 4.8, **2)** the optimal momentum parameter is*

$$m = \left(\sigma_0 - \sqrt{\sigma_0^2 - 1}\right)^{2/K}, \quad \text{where } \sigma_0 = \sigma_K^\Lambda(0)\,, \tag{22}$$

***3)** the iterates from Algo. 3 with parameters $\{h_i\}$ and m form a polynomial with recurrence (18), and **4)** Algorithm 3 achieves the worst-case rate $r_t^{\text{Alg. 3}}$ and the asymptotic rate factor $1-\tau^{\text{Alg. 3}}$*

$$r_t^{\text{Alg. 3}} = O\left(t\left(\sigma_0 - \sqrt{\sigma_0^2-1}\right)^{t/K}\right), \tag{23}$$
$$1-\tau^{\text{Alg. 3}} = \left(\sigma_0 - \sqrt{\sigma_0^2-1}\right)^{1/K}.$$

Solving the system (21) The system is constructed by identification of the coefficients in both polynomials σ_K^Λ and $\sigma(\lambda;\{h_i\},m)$, which can be solved using a naive grid-search for instance. We are not aware of any efficient algorithm to solve this system exactly, although it is possible to use iterative methods such as steepest descent or Newton's method.

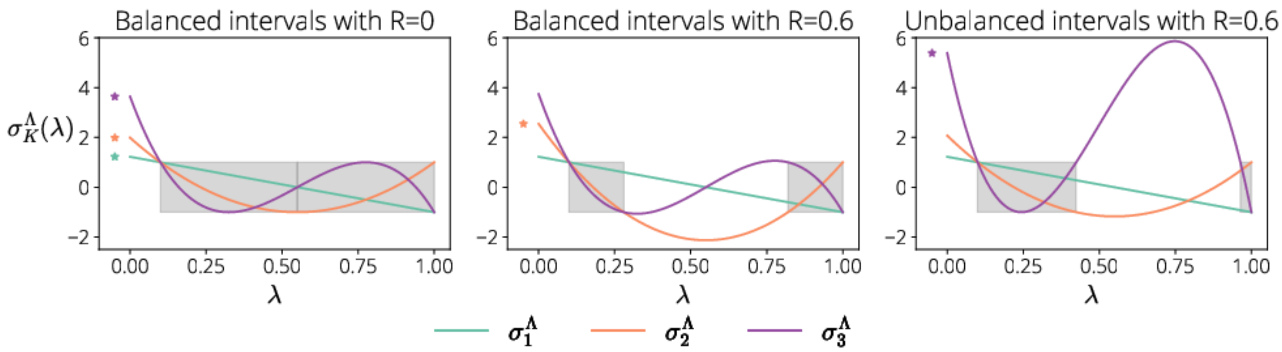

Figure 3: Examples of optimal polynomials σ_K^Λ from (16), all of them verifying the equioscillation property (Definition 4.3). The "$\star$" symbol highlights the degree of σ_K^Λ that achieves the best asymptotic rate $\tau^{\sigma_K^\Lambda}$ in (15) amongst all K (see Section 4.4). **(Left)** When Λ is an unique interval, all 3 polynomials are equivalently optimal $\tau^{\sigma_1^\Lambda} = \tau^{\sigma_2^\Lambda} = \tau^{\sigma_3^\Lambda}$. **(Center)** When Λ is the union of two intervals of the same size, the degree 2 polynomial is optimal $\tau^{\sigma_2^\Lambda} > \tau^{\sigma_3^\Lambda} > \tau^{\sigma_1^\Lambda}$. This is expected given the result in Proposition 4.5. **(Right)** When Λ is the union of two unbalanced intervals, the degree 3 polynomial instead achieves the best asymptotic rate $\tau^{\sigma_3^\Lambda} > \tau^{\sigma_2^\Lambda} > \tau^{\sigma_1^\Lambda}$ (see Section 4.4).

4.4 Best Achievables Worst-case Guarantees on $\mathcal{C}_\Lambda$

This section discusses the (asymptotic) optimality of Algorithm 3. In Section 4.2, the polynomial $P_t(\,\cdot\,;\sigma_K^\Lambda)$ was written as a composition of Chebyshev polynomials with σ_K^Λ, defined in (16). The best K is chosen as follows: we solve (16) for several values of K, then pick the smallest K among the minimizers of (15). However, following such steps does not guarantee that the polynomial $P_{t,K}^\Lambda$ is *minimax*, as it is not guaranteed to minimize the worst-case rate $\sup_{\lambda\in\Lambda}|P_t(\lambda)|$ (see (11)).

We give here an optimality certificate, linked to a generalized version of *equioscillation*. In short, if we can find K non overlapping intervals (more formally, whose interiors are disjoint) Λ_i in Λ such that $\sigma_K^\Lambda(\Lambda_i) = [-1,1]$ then $P_{t,K}^\Lambda$ is minimax for $t = nK$, $n \in \mathbb{N}_0^+$. The precise result is in Theorem C.2. A direct consequence is the asymptotic optimality of Algorithm 3.

We note that σ_K^Λ might not exist for a given Λ. A complete characterization of the set Λ for which σ_K^Λ exists is out of the scope of this paper. A partial answer is given in (Fischer, 2011) when Λ is the union of two intervals but the general case remains open.

5 Local Convergence for Non-Quadratic Functions

When f is twice-differentiable, we show local convergence rates of Algorithm 1 (see proof in Appendix E). As with Polyak heavy ball acceleration, these results are local, as the only known convergence results for Polyak heavy ball beyond quadratic objectives do not lead to an acceleration with respect to Gradient descent without momentum (See Ghadimi et al., 2015, Theorem 4). Moreover, it is possible to find pathological counter-examples and a specific initialization for which the method does not converge globally. Lessard et al. (2016, Figure 7) provides such a counter-example. Note the latest is *not* twice differentiable, but that a twice differentiable counter-example can be derived from the latest, using for instance convolutions.

Theorem 5.1 (Local convergence)**.** *Let $f:\mathbb{R}^d \mapsto \mathbb{R}$ be a twice continuously differentiable function, x_* a local minimizer, and H be the Hessian of f at x_* with $Sp(H) \subseteq \Lambda$. Let x_t denote the result of running Algorithm 1 with parameters $h_1, h_2, \cdots, h_K, m$, and let $1-\tau$ be the linear convergence rate on the quadratic objective* (OPT)*. Then we have*

$$\forall \varepsilon > 0, \exists \text{ open set } V_\varepsilon : x_0,\, x_* \in V_\varepsilon \implies \|x_t - x_*\| = O((1-\tau+\varepsilon)^t)\|x_0 - x_*\|. \quad (24)$$

where $\|\cdot\|$ denotes the Euclidean norm.

In short, when Algorithm 1 is guaranteed to converge at rate $1-\tau$ on (OPT), then the convergence rate on a nonlinear functions can be arbitrary close to $1-\tau$ when x_0 is sufficiently close to x_*.

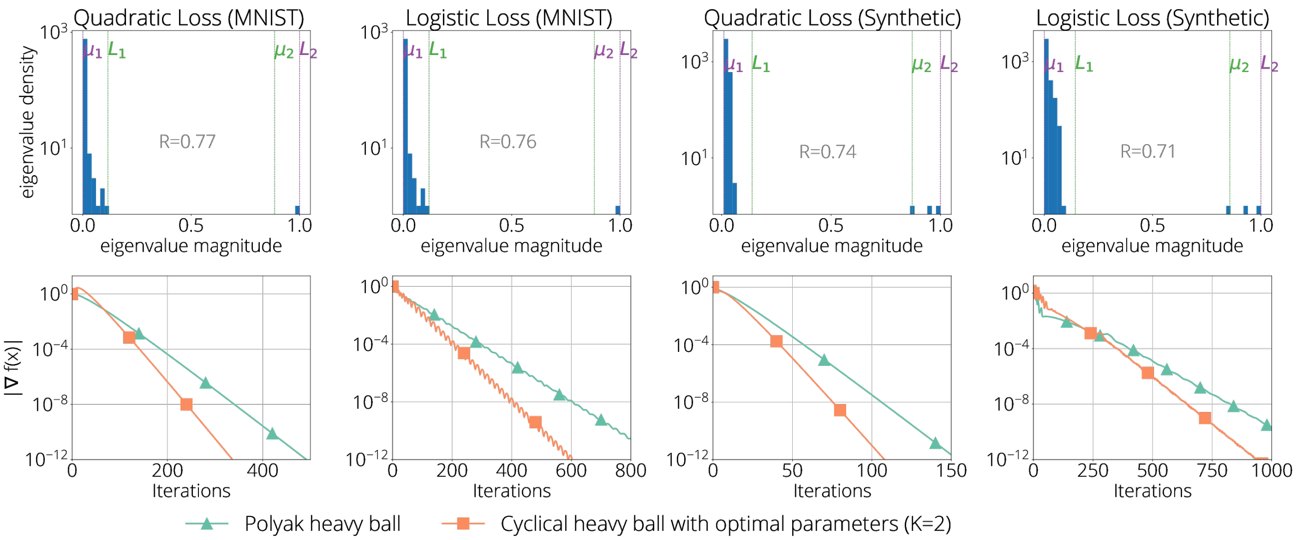

Figure 4: *Hessian Eigenvalue histogram (top row) and Benchmarks (bottom row).* The **top row** shows the Hessian eigenvalue histogram at optimum for the 4 considered problems, together with the interval boundaries $\mu_1 < L_1 < \mu_2 < L_2$ for the two-interval split of the eigenvalue support described in Section 3. In all cases, there's a non-zero gap radius R. This is shown in the **bottom row**, where we compare the suboptimality in terms of gradient norm as a function of the number of iterations. As predicted by the theory, the non-zero gap radius translates into a faster convergence of the cyclical approach, compared to PHB in all cases. The improvement is observed on both quadratic and logistic regression problems, even through the theory for the latter is limited to *local* convergence.

6 Experiments

In this section we present an empirical comparison of the cyclical heavy ball method for different length cycles across 4 different problems. We consider two different problems, quadratic and logistic regression, each applied on two datasets, the MNIST handwritten digits (Le Cun et al., 2010) and a synthetic dataset. The results of these experiments, together with a histogram of the Hessian's eigenvalues are presented in Figure 4 (see caption for a discussion).

Dataset description. The MNIST dataset consists of a data matrix A with 60000 images of handwritten digits each one with $28 \times 28 = 784$ pixels. The *synthetic* dataset is generated according to a spiked covariance model (Johnstone, 2001), which has been shown to be an accurate model of covariance matrices arising for instance in spectral clustering (Couillet and Benaych-Georges, 2016) and deep networks (Pennington and Worah, 2017; Granziol et al., 2020). In this model, the data matrix $A = XZ$ is generated from a $m \times n$ random Gaussian matrix X and an $m \times m$ deterministic matrix Z. In our case, we take $n = 1000, m = 1200$ and Z is the identity where the first three entries are multiplied by 100 (this will lead to three outlier eigenvalues). We also generate an n-dimensional target vector b as $b = Ax$ or $b = \text{sign}(Ax)$ for the quadratic and logistic problem respectively.

Objective function For each dataset, we consider a quadratic and a logistic regression problem, leading to 4 different problems. All problems are of the form $\min_{x \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \ell(A_i^\top x, b_i) + \lambda \|x\|^2$, where ℓ is a quadratic or logistic loss, A is the data matrix and b are the target values. We set the regularization parameter to $\lambda = 10^{-3}\|A\|^2$. For logistic regression, since guarantees only hold at a neighborhood of the solution (even for the 1-cycle algorithm), we initialize the first iterate as the result of 100 iteration of gradient descent. In the case of logistic regression, the Hessian eigenvalues are computed at the optimum.

7 Conclusion

This work is motivated by two recent observations from the optimization practice of machine learning. First, cyclical step-sizes have been shown to enjoy excellent empirical convergence (Loshchilov and Hutter, 2017; Smith, 2017). Second, *spectral gaps* are pervasive in the Hessian spectrum of deep learning models (Sagun et al., 2017; Papyan, 2018; Ghorbani et al., 2019; Papyan, 2019). Based on the simpler context of quadratic convex minimization, we develop a convergence-rate analysis and optimal parameters for the heavy ball method with cyclical step-sizes. This analysis highlights the regimes under which cyclical step-sizes have faster rates than classical accelerated methods. Finally, we illustrate these findings through numerical benchmarks.

Main Limitations. In Section 3 we gave explicit formulas for the optimal parameters in the case of the 2-cycle heavy ball algorithm. These formulas depend not only on extremal eigenvalues—as is usual for accelerated methods—but also on the spectral gap R. The gap can sometimes be estimated after computing the top eigenvalues (e.g. top-2 eigenvalue for MNIST). However, in general, there is no guarantee on how many eigenvalues are needed to estimate it and it must sometimes be seen as hyperparameter. Note Theorem 3.1 provides a convergence analysis also for non-optimal parameters, which would give accelerated convergence rates when doing a coarse grid-search over parameters as it is often done in empirical works.

Another limitation is the fact global convergence results rely heavily on the quadratic assumption which is quite different from our motivation, namely optimizing neural networks. Even if we provide local convergence guarantee in Section 5, we are not able to estimate the size of the optimum neighborhood for which Theorem 5.1 holds.

Another limitation regards long cycles. For cycles longer than 2, we gave an implicit formula to set the optimal parameters (Proposition 4.9). This involves solving a set of non-linear equations whose complexity increases with the cycle length. That being said, cyclical step-sizes might significantly enhance convergence speeds both in terms of worst-case rates and empirically, and this work advocates that new tuning practices involving different cycle lengths might be relevant.

Acknowledgements

We thank Courtney Paquette for very useful proofreading and 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).

References

Naman Agarwal, Surbhi Goel, and Cyril Zhang. Acceleration via Fractal Learning Rate Schedules. *arXiv preprint arXiv:2103.01338*, 2021.

Dimitri P. Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 1997.

Pafnuty Lvovich Chebyshev. *Théorie des mécanismes connus sous le nom de parallélogrammes*. Imprimerie de l'Académie impériale des sciences, 1853.

Romain Couillet and Florent Benaych-Georges. Kernel spectral clustering of large dimensional data. *Electronic Journal of Statistics*, 2016.

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

Donald A. Flanders and George Shortley. Numerical determination of fundamental modes. *Journal of Applied Physics*, 1950.

Euhanna Ghadimi, Hamid Reza Feyzmahdavian, and Mikael Johansson. Global convergence of the heavy-ball method for convex optimization. In *2015 European control conference (ECC)*, pages 310–315. IEEE, 2015.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning (ICML)*, 2019.

Gabriel Goh. Why Momentum Really Works, 2017. URL `http://distill.pub/2017/momentum/`.

Gene H. Golub and Richard S. Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods. *Numerische Mathematik*, 1961.

Diego Granziol, Xingchen Wan, Samuel Albanie, and Stephen Roberts. Explaining the Adaptive Generalisation Gap. *arXiv preprint arXiv:2011.08181*, 2020.

Iain M. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Annals of statistics*, 2001.

Yann Le Cun, Corinna Cortes, and Chris Burges. MNIST handwritten digit database. *ATT Labs [Online]*, 2010.

Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 2016.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

Arkadi S. Nemirovsky. Information-based complexity of linear operator equations. *Journal of Complexity*, 1992.

Arkadi S. Nemirovsky. Information-based complexity of convex programming. *Lecture Notes*, 1995.

Yurii Nesterov. *Introductory Lectures on Convex Optimization*. Springer, 2003.

Samet Oymak. Super-convergence with an unstable learning rate. *arXiv preprint arXiv:2102.10734*, 2021.

Vardan Papyan. The full spectrum of deepnet hessians at scale: Dynamics with SGD training and sample size. *arXiv preprint arXiv:1811.07062*, 2018.

Vardan Papyan. Measurements of Three-Level Hierarchical Structure in the Outliers in the Spectrum of Deepnet Hessians. In *International Conference on Machine Learning (ICML)*, 2019.

Fabian Pedregosa. On the Link Between Optimization and Polynomials, Part 1, 2020. URL `http://fa.bianp.net/blog/2020/polyopt/`.

Fabian Pedregosa. On the Link Between Optimization and Polynomials, Part 3, 2021. URL `http://fa.bianp.net/blog/2021/hitchhiker/`.

Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. In *Advances on Neural Information Processing Systems (NIPS)*, 2017.

Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR computational mathematics and mathematical physics*, 1964.

Heinz Rutishauser. Theory of gradient methods. In *Refined iterative methods for computation of the solution and the eigenvalues of self-adjoint boundary value problems*. Springer, 1959.

Levent Sagun, Utku Evci, V. Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the Hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Damien Scieur and Fabian Pedregosa. Universal Asymptotic Optimality of Polyak Momentum. In *International Conference on Machine Learning (ICML)*, 2020.

Leslie N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning (ICML)*, 2013.

David Young. On richardson's method for solving linear systems with positive definite matrices. *Journal of Mathematics and Physics*, 1953. URL `https://doi.org/10.1002/sapm1953321243`.

Super-Acceleration with Cyclical Step-sizes: Supplementary Materials

Organization of the appendix

The appendix contains all proofs that were not presented in the main core of the paper. We also detail all examples, and provide some complementary elements.

Appendix A details the existing link between first order methods and family of "residual polynomials". This term refers in all the appendix to the polynomials which value in 0 is 1.

In Appendix B, we recall some well known optimal methods for L-smooth μ-strongly convex quadratic minimization (i.e., when the spectrum is contained in a single interval $\Lambda = [\mu, L]$). Its purpose is exclusively to recall well-known foundation of optimization that are those algorithms and their construction.

In Appendix C, we recall the polynomial formulation of the optimal method design problem, as well as a fundamental property, called "equioscillation", to characterize the solution of this problem.

In Appendix D, we provide all proofs related to cyclic step-sizes. In particular,

- In Appendix D.1, we derive the optimal algorithm in a case where Λ is the union of 2 intervals of the same size (See (3)). This leads to the use of alternating step-sizes. The resulting algorithm has a stationary form which is Algorithm 1.
- Therefore, in Appendix D.2, we study the heavy ball with cycling step-sizes (Algorithm 1).
- In Appendix D.3 and Appendix D.4, we use our results to design methods with cycles of lengths $K = 2$ and $K = 3$. For those cases, we provide a more elegant formulation of the results.

In Appendix E, we provide a proof of Theorem 5.1 (local behavior beyond quadratics) and in Appendix F, we provide some information about the code we used for the experiments in quadratic and non quadratic settings.

Finally, in Appendix G we discuss similarities and differences with Oymak (2021).

Contents

A Relationship between first order methods and polynomials

In this section we prove some results on the relationship between polynomials and first order methods for quadratic minimization, which is the starting point for our theoretical framework. This relationship is classical and was exploited by Rutishauser (1959); Nemirovsky (1992, 1995)), to name a few. The following proposition makes this relationship precise:

Proposition 4.1. *Let $f \in \mathcal{C}_\Lambda$. The iterates x_t satisfy*

$$x_{t+1} \in x_0 + \mathrm{span}\{\nabla f(x_0), \ldots, \nabla f(x_t)\}\,, \tag{8}$$

where x_0 is the initial approximation of x_*, 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

$$\forall\, t \quad x_t - x_* = P_t(H)(x_0 - x_*), \quad P_t(0) = 1\,. \tag{9}$$

Proof. We successively prove both directions of the equivalence.

($\Longrightarrow$) *Given a first order method, we can find a sequence of polynomials $(P_t)_{t\in\mathbb{N}}$ such that, for a given quadratic function f of Hessian H and a given starting point x_0, the iterates x_t verify*

$$x_t - x_* = P_t(H)(x_0 - x_*).$$

Moreover, The polynomials sequence $(P_t)_{t\in\mathbb{N}}$ verifies the relations

$$\deg(P_{t+1}) \leq \max_{k\leq t} \deg(P_k) + 1 \quad \textit{and} \quad P_t(0) = 1.$$

We proceed by induction:

Initial case. Let $t = 0$. Then for any first order method we have the trivial relationship

$$x_0 - x_* = P_0(H)(x_0 - x_*) \quad \text{with} \quad P_0 = 1.$$

This proves the implication for $t = 0$, as P_0 is a degree 0 polynomial satisfying $P_0(0) = 1$.

Recursion. Let $t \in \mathbb{N}$. We assume the following statement true,

$$\forall\, k \leq t, \quad x_k - x_* = P_k(H)(x_0 - x_*) \quad \text{with} \quad P_k(0) = 1.$$

We now prove this statement is also true for $t+1$. Since $x_{t+1} \in x_0 + \mathrm{span}\{\nabla f(x_0), \ldots, \nabla f(x_t)\}$, there exists a family $(\gamma_{t+1,k})_{k\in[\![0,t]\!]}$ such that

$$x_{t+1} = x_0 - \gamma_{t+1,0}\nabla f(x_0) - \cdots - \gamma_{t+1,t}\nabla f(x_t). \tag{25}$$

Then, by the induction hypothesis we have:

$$\begin{aligned} x_{t+1} - x_* &= x_0 - x_* - \gamma_{t+1,0} H(x_0 - x_*) - \cdots - \gamma_{t+1,t} H(x_t - x_*) \\ &= x_0 - x_* - \gamma_{t+1,0} H P_0(H)(x_0 - x_*) - \cdots - \gamma_{t+1,t} H P_t(H)(x_0 - x_*) \\ &\triangleq P_{t+1}(H)(x_0 - x_*)\,. \end{aligned}$$

We observe that the latest polynomial has a degree at most 1 plus the highest degree of $(P_k)_{k\leq t}$ and that $P_{t+1}(0) = 1$ (since P_{t+1} is defined as 1 plus some polynomial multiple of the polynomial X), which concludes the proof.

($\Longleftarrow$): *From a family of polynomials $(P_t)_{t\in\mathbb{N}}$, with*

$$\deg(P_{t+1}) \leq \max_{k\leq t} \deg(P_k) + 1 \quad \textit{and} \quad P_t(0) = 1, \tag{26}$$

we can obtain a first order method such that, for any quadratic f (and its Hessian H) and any starting point x_0, we verify

$$\forall t \in \mathbb{N}, x_t - x_* = P_t(H)(x_0 - x_*).$$

Let the sequence $(P_t)_{t\in\mathbb{N}}$ verifies (26) for all $t \in \mathbb{N}$. Let

$$d = \max_{t' \le t} \deg(P_{t'}).$$

A gap in the sequence of degrees would stand in contradiction with our assumptions.

Since, there is no gap in degree, for any $d' \le d$ there exists $t' \le t$ such that $\deg(P_{t'}) = d'$, and therefore $\mathrm{Span}((P_k)_{k\le t}) = \mathbb{R}_d[X]$.

Moreover, we know P_{t+1} has a degree at most $d+1$ and $P_{t+1}(0) = 1$, so $\frac{1-P_{t+1}(X)}{X} \in \mathbb{R}_d[X]$.

This proves the existence of $(\gamma_{t+1,k})_{k\in[\![0,t]\!]}$ such that

$$\frac{1 - P_{t+1}(X)}{X} = \gamma_{t+1,0}P_0(X) + \cdots + \gamma_{t+1,t}P_t(X). \tag{27}$$

Then, defining

$$x_{t+1} = x_0 - \gamma_{t+1,0}\nabla f(x_0) - \cdots - \gamma_{t+1,t}\nabla f(x_t)\,, \tag{28}$$

we have

$$\begin{aligned} x_{t+1} - x_* &= x_0 - x_* - H\left(\gamma_{t+1,0}(x_0 - x_*) + \cdots + \gamma_{t+1,t}(x_t - x_*)\right) && (29)\\ &= \left(1 - X\left(\gamma_{t+1,0}P_0(X) + \cdots + \gamma_{t+1,t}P_t(X)\right)\right)(H)(x_0 - x_*) && (30)\\ &= P_{t+1}(H)(x_0 - x_*)\,. && (31) \end{aligned}$$

Defining x_t for all t according to (28) gives an algorithm that has as associated residual polynomials $(P_t)_{t\in\mathbb{N}}$. ☐

The above proposition can be used to obtain worst-case rates for first order methods by bounding their associated polynomials. Indeed, using the Cauchy-Schwartz inequality in (9) leads to

$$\|x_t - x_*\| \le \sup_{\lambda\in\Lambda} |P_t(\lambda)|\, \|x_0 - x_*\| \quad \implies \quad r_t = \sup_{\lambda\in\Lambda} |P_t(\lambda)|, \quad \text{where } P(0) = 1\,. \tag{32}$$

Therefore, finding the algorithm with the fastest worst-case rate can be equivalently framed as the problem of finding the residual polynomial with smallest value on the eigenvalue support Λ.

Then, finding the fastest algorithm is equivalent of finding, for each $t \ge 0$, the polynomial of degree t that reaches the smallest infinite norm on the set Λ. Therefore we introduce the notion of *minimax polynomial* (Definition A.1) over a set Λ as the one that reaches the smallest maximal value over Λ among a set of polynomial of fixed degree and $P(0) = 1$.

Definition A.1 (Minimax polynomial of degree t over Λ)**.** For any, $t \ge 0$, and any relatively compact (i.e. bounded) set $\Lambda \subset \mathbb{R}$, the *minimax polynomial of degree t over* Λ, written Z_t^Λ, is defined as

$$Z_t^\Lambda \triangleq \operatorname*{argmin}_{P\in\mathbb{R}_t[X]} \sup_{\lambda\in\Lambda} |P(\lambda)|, \quad \text{subject to} \quad P(0) = 1\,. \tag{33}$$

B Optimal methods for strongly convex and smooth quadratic objective

In this section, for sake of completness, we revisit some classical methods, described in e.g. (Polyak, 1964; Goh, 2017; Pedregosa, 2020, 2021), that are optimal when the Hessian eigenvalues are contained in a single interval of the form $\Lambda = [\mu, L]$. To make this setup explicit, we will denote the optimal polynomials σ_1^Λ and Z_t^Λ (respectively defined in Equation (16) and Equation (33)) by $\sigma_1^{[\mu,L]}$, and $Z_t^{[\mu,L]}$.

As mentioned in Example 4.4, the minimax polynomial $Z_t^{[\mu,L]}$ is

$$Z_t^{[\mu,L]}(\lambda) = \frac{T_t(\sigma_1^{[\mu,L]}(\lambda))}{T_t(\sigma_1^{[\mu,L]}(0))},$$

where T_t denotes the t^{th} Chebyshev polynomial (See e.g. Chebyshev (1853)) and $\sigma_1^{[\mu,L]}$ the affine function $\sigma(\lambda) \triangleq \frac{L+\mu}{L-\mu} - \frac{2}{L-\mu}\lambda$ that maps $[\mu, L]$ onto $[-1, 1]$. This can be seen a consequence of the more general *equioscillation* discussed in Appendix C. The next section presents one method which has $Z_t^{[\mu,L]}$ as associated residual polynomial. This method is known as the Chebyshev semi-iterative method.

B.1 Chebyshev semi-iterative method

The algorithm follows the three terms pattern from Equation (13) to iteratively form $Z_1^\Lambda, \ldots, Z_t^\Lambda$.

Algorithm 4: Chebyshev semi-iterative method (Golub and Varga, 1961)

Input: x_0
Initialize: $\omega_0 = 2$
$x_1 = x_0 - \frac{2}{L+\mu}\nabla f(x_0)$;

for $t = 1, \ldots$ **do**
  $\omega_{t+1} = \left(1 - \frac{1}{4}\left(\frac{1-\kappa}{1+\kappa}\right)^2 \omega_t\right)^{-1}$;
  $x_{t+1} = x_t - \frac{2}{L+\mu}\omega_t \nabla f(x_t) + (\omega_t - 1)(x_t - x_{t-1})$;
end

Theorem B.1. *The iterates produced by the Chebyshev semi-iterative method verify*

$$x_t - x_* = \frac{T_t(\sigma_1^{[\mu,L]}(H))}{T_t(\sigma_1^{[\mu,L]}(0))}(x_0 - x_*) \quad \text{for all } t \in \mathbb{N}. \tag{34}$$

Furthermore, this method enjoys a worst-case rate of the form

$$\|x_t - x_*\| \leq \frac{1}{T_t(\sigma_1^{[\mu,L]}(0))}\|x_0 - x_*\| = O\left(\left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^t\right). \tag{35}$$

Proof. Consider first an algorithm whose iterates verify (34). Then using the Cauchy-Schwartz inequality and known bounds of Chebyshev polynomials, we can show the following rate

$$\begin{aligned}
\|x_t - x_*\| &\leq \frac{\sup_{\lambda\in[\mu,L]} |T_t(\sigma_1^{[\mu,L]}(\lambda))|}{T_t(\sigma_1^{[\mu,L]}(0))}\|x_0 - x_*\| \\
&= \frac{1}{T_t\left(\frac{1+\kappa}{1-\kappa}\right)}\|x_0 - x_*\| && \text{since } \sup_{x\in[-1,1]} |T_t(x)| = 1 \\
&\leq 2\left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^t \|x_0 - x_*\| && \text{since } T_t(x) \geq \frac{\left(x+\sqrt{x^2-1}\right)^t}{2}, \forall x \notin (-1, 1).
\end{aligned}$$

It remains to prove that Algorithm 4 is the one that achieves the property (34). Using the recursion verified by Chebyshev polynomials

$$T_{t+1}(x) = 2xT_t(x) - T_{t-1}(x), \tag{36}$$

we have

$$
\begin{aligned}
x_{t+1} - x_* &= \frac{T_{t+1}(\sigma_1^{[\mu,L]}(H))}{T_{t+1}(\sigma_1^{[\mu,L]}(0))}(x_0 - x_*) \\
&= \frac{2\sigma_1^{[\mu,L]}(H)T_t(\sigma_1^{[\mu,L]}(H))(x_0 - x_*) - T_{t-1}(\sigma_1^{[\mu,L]}(H))(x_0 - x_*)}{T_{t+1}(\sigma_1^{[\mu,L]}(0))} \\
&= \frac{2\sigma_1^{[\mu,L]}(H)T_t(\sigma_1^{[\mu,L]}(0))}{T_{t+1}(\sigma_1^{[\mu,L]}(0))}(x_t - x_*) - \frac{T_{t-1}(\sigma_1^{[\mu,L]}(0))}{T_{t+1}(\sigma_1^{[\mu,L]}(0))}(x_{t-1} - x_*) \\
&= \frac{2\sigma_1^{[\mu,L]}(0)T_t(\sigma_1^{[\mu,L]}(0))}{T_{t+1}(\sigma_1^{[\mu,L]}(0))}\left(I - \frac{2}{L+\mu}H\right)(x_t - x_*) - \frac{T_{t-1}(\sigma_1^{[\mu,L]}(0))}{T_{t+1}(\sigma_1^{[\mu,L]}(0))}(x_{t-1} - x_*)\,.
\end{aligned}
$$

Let's introduce $\omega_t \triangleq \frac{2\sigma_1^{[\mu,L]}(0)T_t(\sigma_1^{[\mu,L]}(0))}{T_{t+1}(\sigma_1^{[\mu,L]}(0))}$. Then $\omega_0 = 2$ and by Chebyshev recursion (Equation (36)), $\omega_t - 1 = \frac{T_{t-1}(\sigma_1^{[\mu,L]}(0))}{T_{t+1}(\sigma_1^{[\mu,L]}(0))}$. With this notation we can write the above identity more compactly as

$$
\begin{aligned}
x_{t+1} - x_* &= \omega_t \left(I - \frac{2}{L+\mu}H\right)(x_t - x_*) - (\omega_t - 1)(x_{t-1} - x_*) \\
&= x_t - \frac{2}{L+\mu}\omega_t \nabla f(x_t) + (\omega_t - 1)(x_t - x_{t-1})\,.
\end{aligned}
$$

It remains to find a recursion on ω_t to make its use tractable. Using one more time the Chebyshev recursion Equation (36),

$$
\begin{aligned}
\omega_t^{-1} &= \frac{T_{t+1}(\sigma_1^{[\mu,L]}(0))}{2\sigma_1^{[\mu,L]}(0)T_t(\sigma_1^{[\mu,L]}(0))} \\
&= \frac{2\sigma_1^{[\mu,L]}(0)T_t(\sigma_1^{[\mu,L]}(0)) - T_{t-1}(\sigma_1^{[\mu,L]}(0))}{2\sigma_1^{[\mu,L]}(0)T_t(\sigma_1^{[\mu,L]}(0))} \\
&= 1 - \frac{1}{4\sigma_1^{[\mu,L]}(0)^2}\frac{2\sigma_1^{[\mu,L]}(0)T_{t-1}(\sigma_1^{[\mu,L]}(0))}{T_t(\sigma_1^{[\mu,L]}(0))} \\
&= 1 - \frac{1}{4\sigma_1^{[\mu,L]}(0)^2}\omega_{t-1},
\end{aligned}
$$

which can finally be written as

$$
\omega_{t+1} = \frac{1}{1 - \frac{1}{4}\left(\frac{1-\kappa}{1+\kappa}\right)^2 \omega_t},
$$

and we recognize the *Chebyshev semi-iterative method* described in Algorithm 4. □

This method, unlike the Polyak heavy ball (PHB) method, uses a different step-size and momentum at each iteration. However, both are related, as taking the limit of ω_t as $t \to \infty$ in Algorithm 4 we obtain $\omega_\infty = 1 + m$ with $m = \left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^2$. This correspond to the parameters of PHB.

We note that this is only one way to construct a method that has the Chebsyshev polynomial as residual polynomial at every iteration. However, it is possible to construct a different update that have the Chebyshev polynomial at fixed iteration, see for instance (Young, 1953; Agarwal et al., 2021) for one such alterative that does not require momentum.

B.2 Polyak heavy ball method

Algorithm 5: Polyak Heavy ball

Input: x_0

Set: $m = \left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^2$ and $h = \frac{2(1+m)}{L+\mu}$.

$x_1 = x_0 - \frac{h}{1+m}\nabla f(x_0)$

for $t = 1, \ldots$ **do**

| $x_{t+1} = x_t - h\nabla f(x_t) + m(x_t - x_{t-1})$

end

Theorem B.2. *The iterates of the heavy ball algorithm verify*

$$x_t - x_* = P_t(H)(x_0 - x_*) \quad \textit{for all } t \in \mathbb{N},$$

with P_t defined as

$$P_t(\lambda) \triangleq \left(\sqrt{m}\right)^t \left[\frac{2m}{1+m} T_t(\sigma_1^{[\mu,L]}(\lambda)) + \frac{1-m}{1+m} U_t(\sigma_1^{[\mu,L]}(\lambda))\right]. \tag{37}$$

Furthermore, this method enjoys a worst-case rate of the form

$$\|x_t - x_*\| = O\left(t\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^t\right). \tag{38}$$

Proof. From the update defined in Algorithm 5, we identify

$$\begin{aligned} P_0(\lambda) &= 1 \\ P_1(\lambda) &= 1 - \frac{h}{1+m}\lambda \\ P_{t+1}(\lambda) &= (1+m-h\lambda)P_t(\lambda) - mP_{t-1}(\lambda). \end{aligned}$$

Introducing $\tilde{P}_t \triangleq \frac{P_t}{\left(\sqrt{m}\right)^t}$, we have

$$\begin{aligned} \tilde{P}_0(\lambda) &= 1 \\ \tilde{P}_1(\lambda) &= \frac{1+m-h\lambda}{(1+m)\sqrt{m}} = \frac{2}{1+m}\sigma_1^{[\mu,L]}(\lambda) \\ \tilde{P}_{t+1}(\lambda) &= \frac{(1+m-h\lambda)}{\sqrt{m}}\tilde{P}_t(\lambda) - \tilde{P}_{t-1}(\lambda) \\ &= 2\sigma_1^{[\mu,L]}(\lambda)\tilde{P}_t(\lambda) - \tilde{P}_{t-1}(\lambda). \end{aligned}$$

This is a second order recurrence, with 2 initializations. It allows us to identify uniquely the family

$$\tilde{P}_t(\lambda) = \frac{2m}{1+m} T_t(\sigma_1^{[\mu,L]}(\lambda)) + \frac{1-m}{1+m} U_t(\sigma_1^{[\mu,L]}(\lambda)). \tag{39}$$

where U_t denotes the Chebyshev polynomial of the second kind of degree t. While both T_t and U_t verify the same recursion as $\tilde{P}_t$ and $T_0 = U_0 = \tilde{P}_0 = 1$, the difference between T and U comes when $T_1(X) = X$ and $U_1(X) = 2X$. This is how $\tilde{P}_t$ ends being a linear combination of the T_t and U_t. Finally,

$$P_t(\lambda) = \left(\sqrt{m}\right)^t \left[\frac{2m}{1+m} T_t(\sigma_1^{[\mu,L]}(\lambda)) + \frac{1-m}{1+m} U_t(\sigma_1^{[\mu,L]}(\lambda))\right]. \tag{40}$$

Since by definition $\sigma_1^{[\mu,L]}([\mu, L]) = [-1, 1]$, $T_t(\sigma_1^{[\mu,L]}(\lambda)) \leq 1$ and $U_t(\sigma_1^{[\mu,L]}(\lambda)) \leq t+1, \forall t \in \mathbb{N}$. Hence, $\forall \lambda \in [\mu, L]$,

$$P_t(\lambda) \leq \left(\sqrt{m}\right)^t \left[1 + \frac{1-m}{1+m}t\right] \leq (2\sqrt{\kappa}t + 1)\left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^t \tag{41}$$

and

$$\|x_t - x_*\| = O\left(t\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^t\right). \tag{42}$$

□

C Minimax Polynomials and Equioscillation Property

Appendix B dealt with optimal methods when $\Lambda = [\mu, L]$. Those methods could be derived since the minimax polynomial (Definition A.1) $Z_t^{[\mu,L]}$ is known.

In this section we consider the problem of finding minimax polynomials in a more general setting. We provide a characterization of the minimax polynomial defined in definition A.1. For the sake of simplicity, we actually focus on the polynomial σ_t^Λ solution of (16). We can easily adapt the result to Z_t^Λ leveraging Remark 4.7. We prove the following theorem.

Theorem C.1. *Let P_K be a degree K polynomial verifying $P_K(\Lambda) \subset [-1, 1]$. Then P_K is the unique solution σ_K^Λ of eq. (16) if and only if there exists a sorted family $(\lambda_i)_{i\in[\![0,K]\!]} \in \left(\overline{\Lambda}\right)^{K+1}$ (where $\overline{\Lambda}$ is the closure of Λ) such that $\forall i \in [\![0, K]\!], P_K(\lambda_i) = (-1)^i$.*

The following proof is technical and requires to introduce several new notations. Hence we first briefly describe the intuition before giving the actual complete proof.

($\Longleftarrow$): Assume P_K "oscillates" $K + 1$ times between 1 and -1. Since P_K has a degree K, it is completely determined by its values on those $K + 1$ points, using the Lagrange interpolation representation. We prove that P_K is optimal because any other polynomial Q_K, having different values on those $K + 1$ points would achieve a smaller value $Q_K(0)$ at 0.

($\Longrightarrow$): We prove this by contradiction. We assume that P_K doesn't oscillate $K + 1$ times between 1 and -1, and prove that $P_K(0)$ is not optimal. To do so, we build a small perturbation εQ_K such that $P_K + \varepsilon Q_K$ is a polynomial of degree K, which values on Λ are all in $[-1; 1]$, and with an higher value at 0.

(Uniqueness) We reuse the Lagrange interpolation representation to justify that 2 optimal polynomials must "oscillate" on the same points, therefore are equal.

Proof. We prove successively both directions:

($\Longleftarrow$): *Assume $\exists \lambda_0 < \lambda_1 < \cdots < \lambda_K$ such that*

$$\forall i \in [\![0, K]\!], P_K(\lambda_i) = (-1)^i \quad \text{and} \quad P_K(\Lambda) \subset [-1, 1]. \tag{43}$$

We aim to prove that P_K is the unique solution σ_K^Λ of eq. (16), that is for any other polynomial Q_K of degree K verifying $Q_K(\Lambda) \subset [-1, 1]$, $P_K(0) \geq Q_K(0)$.

We introduce such a polynomial Q_K of degree K and bounded in absolute value by 1 on Λ. Let's define, for all $i \in [\![0,\ K]\!]$,

$$v_i \triangleq Q_K(\lambda_i) \in [-1, 1]. \tag{44}$$

These $K + 1$ values characterize Q_K (of degree K), and we can decompose it over Lagrange interpolation polynomials. We have

$$Q_K = \sum_{i=0}^{K} v_i L_{\lambda_i} \quad \text{where} \quad L_{\lambda_i}(X) \triangleq \prod_{j\neq i} \frac{X - \lambda_j}{\lambda_i - \lambda_j} \ . \tag{45}$$

The value at 0 can be computed as

$$Q_K(0) = \sum_{i=0}^{K} v_i L_{\lambda_i}(0) = \sum_{i=0}^{K} v_i \prod_{j\neq i} \frac{\lambda_j}{\lambda_j - \lambda_i} \ . \tag{46}$$

Maximizing this linear function of $(v_i)_{i\in[\![0,\ K]\!]}$ over the ℓ_∞ ball $B_\infty(1) \triangleq \{(v_i)_{i\in[\![0,\ K]\!]}, \forall i, -1 \leq v_i \leq 1\}$ leads to, for $v^* \triangleq \arg\min_{v\in B_\infty(1)} \sum_{i=0}^{K} v_i \prod_{j\neq i} \frac{\lambda_j}{\lambda_j - \lambda_i}$,

$$v_i^* = \operatorname{sgn}\left(\prod_{j\neq i} \frac{\lambda_j}{\lambda_j - \lambda_i}\right) = (-1)^i. \tag{47}$$

where sgn is the sign function (which maps 0 to 0, $\mathbb{R}_{<0}$ to -1, and $\mathbb{R}_{>0}$ to 1). Finally,

$$P_K(0) \geq Q_K(0) \tag{48}$$

which concludes the proof.

($\Longrightarrow$): *Assume P_K alternates $s < K+1$ times between -1 and 1 on $\overline{\Lambda}$. We want to show that P_K is not optimal in the sense described above. To do so, we construct a perturbation of P_K that increases its value in 0 while still satisfying the constraint $P_K(\Lambda) \subset [-1, 1]$.*

Let's define

$$\lambda_0^{(1)} < \cdots < \lambda_0^{(\nu_0)} < \lambda_1^{(1)} < \cdots < \lambda_1^{(\nu_1)} < \cdots < \lambda_{s-1}^{(1)} < \cdots < \lambda_{s-1}^{(\nu_{s-1})} \tag{49}$$

such that

$$P_K(\lambda_i^{(j)}) = (-1)^i \quad \text{and} \quad \forall \lambda \in \overline{\Lambda}, \left(\exists (i,j) | \lambda = \lambda_i^{(j)} \text{ or } |P_K(\lambda)| < 1\right). \tag{50}$$

In short, $\left(\lambda_i^{(j)}\right)_{(i,j)}$ describes all the extremal points of P_K in Λ. The indices change when the sign changes, while the exponents are used to express the possible consecutive repetitions of the same value (-1 or 1).

Set $(r_i)_{i \in [\![0,s]\!]}$ as any set of positive numbers satisfying:

$$0 < r_0 < \inf(\Lambda) < \lambda_0^{(1)} < \lambda_0^{(\nu_0)} < r_1 < \cdots < r_s < \lambda_{s-1}^{(1)} < \lambda_{s-1}^{(\nu_{s-1})} < \sup(\Lambda) < r_s. \tag{51}$$

By definition, each interval $[r_i, r_{i+1}]$, $i \in [\![0,\, s-1]\!]$, contains $\lambda_i^{(j)}$ for all j, but no other extremal points of P_K in $\overline{\Lambda}$. Hence, $P_K([r_i, r_{i+1}] \cap \overline{\Lambda})$ doesn't contain $(-1)^{i+1}$. Since, $\bigcup_{i<s, i \text{ even}} [r_i, r_{i+1}] \cap \overline{\Lambda}$ is compact, and by continuity of P_K, $P_K \left(\bigcup_{i<s, i \text{ even}} [r_i, r_{i+1}] \cap \overline{\Lambda}\right)$ is compact. Therefore,

$$\exists \varepsilon_{-1} > 0 | P_K \left(\bigcup_{i<s, i \text{ even}} [r_i, r_{i+1}] \cap \overline{\Lambda}\right) \subset [-1 + \varepsilon_{-1}, 1]. \tag{52}$$

Similarly, we obtain

$$\exists \varepsilon_1 > 0 | P_K \left(\bigcup_{i<s, i \text{ odd}} [r_i, r_{i+1}] \cap \overline{\Lambda}\right) \subset [-1, 1 - \varepsilon_1]. \tag{53}$$

We are now equipped to build the aforementioned perturbation. Let

$$Q_K(X) \triangleq \prod_{i \in [\![0, s-1]\!]} (r_i - X). \tag{54}$$

Note that Q_K has a degree $s \leq K$ and satisfies

$$Q_K \left(\bigcup_{i<s, i \text{ even}} [r_i, r_{i+1}] \cap \overline{\Lambda}\right) \subset \mathbb{R}^- \quad \text{and} \quad Q_K \left(\bigcup_{i<s, i \text{ odd}} [r_i, r_{i+1}] \cap \overline{\Lambda}\right) \subset \mathbb{R}^+. \tag{55}$$

Moreover, those sets are compact , by continuity of Q_K, and consequently bounded. We can therefore choose a small enough $\varepsilon > 0$ such that

$$\varepsilon \min Q_K \left(\bigcup_{i<s, i \text{ even}} [r_i, r_{i+1}] \cap \overline{\Lambda}\right) > -\varepsilon_{-1} \quad \text{and} \quad \varepsilon \max Q_K \left(\bigcup_{i<s, i \text{ odd}} [r_i, r_{i+1}] \cap \overline{\Lambda}\right) < \varepsilon_1.$$

This leads to

$$(P_K + \varepsilon Q_K)(\Lambda) \subset [-1, 1]. \tag{56}$$

And as by definition, $Q_K(0) > 0$,

$$(P_K + \varepsilon Q_K)(0) > P_K(0). \tag{57}$$

Finally $(P_K + \varepsilon Q_K) \in \mathbb{R}_K[X]$. This proves that P_K is not optimal.

(Uniqueness) *Here, we prove that the optimal polynomial is necessarily unique. To do so, we introduce 2 optimal polynomials and show there must actually be identical.*

Let P_K an optimal polynomial and $(\lambda_i)_{i\in[\![0,K]\!]} \in \Lambda^{K+1}$ a family on which P_K interpolates alternatively 1 and -1. Let any other feasible polynomial Q_K and $(v_i)_{i\in[\![0,K]\!]}$ its values on $(\lambda_i)_{i\in[\![0,K]\!]}$:

$$Q_K = \sum_{i=0}^{K} v_i L_{\lambda_i}. \tag{58}$$

We have showed in the first point of this proof that the optimal values of v_i are alternatively 1 and -1. Consequently, if Q_K is also optimal,

$$Q_K(\lambda_i) = P_K(\lambda_i) \tag{59}$$

for all $i \in [\![0, K]\!]$, which characterizes polynomials of degree K. Then

$$Q_K = P_K \tag{60}$$

which shows that the optimal polynomial is unique. □

We now give the formal statement and the proof of the second result, used in Subsection 4.4.

Theorem C.2. *$T_n(\sigma_K)$ is optimal for all n if and only if σ_K verifies the equioscillation property (Definition 4.3, hence $\sigma_K = \sigma_K^\Lambda$ by Theorem C.1) and $\overline{\Lambda} = \sigma_K^{-1}([-1,1])$, i.e. the inverse mapping σ_K^{-1} transforms the interval $[-1,1]$ into exactly $\overline{\Lambda}$.*

Before providing the proof, we first highlight that the property

$$\forall \lambda \in \Lambda, \sigma_K(\lambda) \in [-1, 1] \tag{61}$$

can equivalently be written

$$\overline{\Lambda} \subset \sigma_K^{-1}([-1, 1]). \tag{62}$$

In other words, we are interested in the case where the reverse inclusion holds as well. This means that

$$\sigma_K(\lambda) \in [-1, 1] \Rightarrow \lambda \in \overline{\Lambda}. \tag{63}$$

This corresponds to a stronger form of optimality of σ_K: it "fully" uses the available assumption related to Λ, in the sense that no point can be added to $\overline{\Lambda}$ without breaking the condition $\sigma_K(\Lambda) \subset [-1; 1]$. For example, on Figure 3, σ_3^Λ *does not* satisfy the later property on the center graph, but satisfies it on the right graph. Here, we show that under this condition, $T_n(\sigma_K) = T_n(\sigma_K^\Lambda)$ is optimal (in the sense of (16)) for all $n \in \mathbb{N}$.

In Section 4.4, we give another view of this condition for $T_n(\sigma_K)$ to be optimal for all n. We can decompose Λ as the union of K intervals Λ_i such that they have disjoint interiors and they are all mapped to $[-1, 1]$ by σ_K. Hence, σ_K maps Λ to $[-1, 1]$ exactly K times.

Proof. From Theorem C.1, $T_n(\sigma_K)$ is optimal for all n if and only if, for all n, there exist a sorted family of $(\lambda_i)_{i\in[\![0,nK]\!]}$ such that, $T_n(\sigma_K(\lambda_i)) = (-1)^i$.

Let $n \in \mathbb{N}$. We observe that by definition of T_n,

$$T_n(\sigma_K(\lambda)) = \pm 1 \quad \text{if and only if} \quad \exists j \in [\![0, n]\!] | \sigma_K(\lambda) = \cos \frac{j\pi}{n}. \tag{64}$$

We successively treat both directions: ($\Longleftarrow$) *we assume σ_K oscillates and $\overline{\Lambda} = \sigma_K^{-1}([-1, 1])$. We aim to prove that $T_n(\sigma_K)$ is optimal for all $n \in \mathbb{N}$.*

By equioscillation property, we know that there exists λ_i' such that

$$\sigma_K(\lambda_i') = (-1)^i. \tag{65}$$

By the intermediate value theorem, we know that for any $i \in [\![0; K]\!]$, between the pair $\lambda'_i, \lambda'_{i+1}$, there exist sorted $(\mu_i^j)_{ni<j<(n+1)i}$ such that for all $j \in [\![ni+1; (n+1)i-1]\!]$,

$$\sigma_K(\mu_i^j) = \cos\frac{j\pi}{n}. \tag{66}$$

We identify $\lambda_{ni} = \lambda'_i$ and $\lambda_j = \mu^j_{\lfloor j/n \rfloor}$ for all j not multiple of n. Then, for all $\ell \in [\![0, nK]\!]$:

$$T_n(\sigma_K(\lambda_\ell)) = (-1)^\ell. \tag{67}$$

By Theorem C.1, we conclude that $T_n(\sigma_K)$ is optimal for all $n \in \mathbb{N}$.

($\Longrightarrow$) *We assume $T_n(\sigma_K)$ is optimal for all $n \in \mathbb{N}$. Clearly, σ_K is optimal ($n = 1$), and then equioscillates. We prove that moreover*

$$\overline{\Lambda} = \sigma_K^{-1}([-1, 1]). \tag{68}$$

On the one hand, for any $j \in [\![0, n]\!]$, there exist at most K different λ that verifies $\sigma_K(\lambda) = \cos\frac{j\pi}{n}$ since σ_K has a degree K and is not constant. Therefore, there exist at most $(n+1)K$ different λ such that $\exists j \in [\![0, n]\!] | \sigma_K(\lambda) = \cos\frac{j\pi}{n}$, and by Eq.(64), there thus exist at most $(n+1)K$ different λ such that $T_n(\sigma_K(\lambda) = \pm 1$.

On the other hand, the optimality of $T_n(\sigma_K)$ implies the existence of *at least* $nK + 1$ such λ in $\overline{\Lambda}$.

Hence all but at most $K - 1$ values λ such that $\sigma_K(\lambda) \in \{\cos\frac{j\pi}{n}, j \in [\![0, n]\!]\}$ belong to $\overline{\Lambda}$.

This holds for all n. Therefore for n large enough, all x such that $\sigma(x) \in [-1, 1]$ are as close as we want to some $\lambda \in \overline{\Lambda}$. Since $\overline{\Lambda}$ is a closed set, then all x such that $\sigma(x) \in [-1, 1]$ are actually in $\overline{\Lambda}$.

We conclude

$$\overline{\Lambda} \supset \sigma_K^{-1}([-1, 1]). \tag{69}$$

□

D Cyclical step-sizes

In this appendix, we provide an analysis of momentum methods with cyclical step-sizes and derive some non-asymptotically optimal variants.

D.1 Derivation of optimal algorithm with $K = 2$ alternating step-sizes

In this section, we consider the case where Λ is the union of 2 intervals of same size, as described in Section 3.

We start by introducing the following algorithm, and we will prove later that this algorithm is optimal (Theorem D.1)

Theorem D.1. *Let $f \in \mathcal{C}_\Lambda$ and $x_0 \in \mathbb{R}^d$. Assume Λ defined as in (3). The iterates of Algorithm 6 verifies the condition*

$$x_{2n} - x_* = \frac{T_n(\sigma_2^\Lambda(H))}{T_n(\sigma_2^\Lambda(0))}(x_0 - x_*) \tag{70}$$

and this is the optimal convergence rate over $\mathcal{C}_\Lambda$.

Proof. We begin by showing the optimality of the algorithm. Using Proposition D.2, the polynomial in (70) equioscillates on Λ, which makes it optimal by Theorem C.1. By optimal, this means this is the optimal convergence rate any first order algorithm can reach (See (11)). We invite the reader to read Appendix D.3, where we study in details the properties of the alternating steps sizes strategy (i.e., $K = 2$).

As in Appendix B.1, we derive here the constructive approach that leads us to this algorithm.

We now start showing that the iterates of Algorithm 6 follow (70). From eq. (70), projecting onto the eigenspace of eigenvalue λ,

$$x_{2n} - x_* = \frac{T_n(\sigma_2^\Lambda(\lambda))}{T_n(\sigma_2^\Lambda(0))}(x_0 - x_*). \tag{71}$$

Algorithm 6: Optimal momentum method with alternating step-sizes ($K = 2$)

Input: Initialization x_0, $\mu_1 < L_1 < \mu_2 < L_2$ (where $L_1 - \mu_1 = L_2 - \mu_2$)

Set: $\rho = \frac{L_2+\mu_1}{L_2-\mu_1}$, $R = \frac{\mu_2-L_1}{L_2-\mu_1}$, $c = \sqrt{\frac{\rho^2-R^2}{1-R^2}}$

$\omega_0 = 2$

$x_1 = x_0 - \frac{1}{L_1}\nabla f(x_0)$

for $t = 1, 2, \ldots$ **do**

$$\omega_t = \left(1 - \frac{1}{4c^2}\omega_{t-1}\right)^{-1}$$

$$h_t = \frac{\omega_t}{L_1} \quad \text{(if } t \text{ is even)}, \qquad h_t = \frac{\omega_t}{\mu_2} \quad \text{(if } t \text{ is odd)}$$

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

end

Then, we find a recursion definition for the subsequence $(x_{2n})_{n\in\mathbb{N}}$. Let $n \geq 1$.

$$\begin{aligned} x_{2(n+1)} - x_* &= \frac{T_{n+1}(\sigma_2^\Lambda(\lambda))}{T_{n+1}(\sigma_2^\Lambda(0))}(x_0 - x_*), &(72)\\ &= \frac{2\,\sigma_2^\Lambda(\lambda)\,T_n(\sigma_2^\Lambda(\lambda)) - T_{n-1}(\sigma_2^\Lambda(\lambda))}{T_{n+1}(\sigma_2^\Lambda(0))}(x_0 - x_*), &(73)\\ &= \frac{2\,\sigma_2^\Lambda(\lambda)\,T_n(\sigma_2^\Lambda(0))}{T_{n+1}(\sigma_2^\Lambda(0))}(x_{2n} - x_*) - \frac{T_{n-1}(\sigma_2^\Lambda(0))}{T_{n+1}(\sigma_2^\Lambda(0))}(x_{2(n-1)} - x_*). &(74) \end{aligned}$$

Note that if $\sigma_2^\Lambda(\lambda)$ were a degree 1 polynomial in λ, then we would recognize a momentum update. Here, $\sigma_2^\Lambda(\lambda)$ is actually a degree 2 polynomial in λ. We will then try to identify 2 steps of momentum. From here, let

$$c \triangleq \frac{1}{2}\left(\left(\sigma_K(0) + \sqrt{\sigma_K(0)^2 - 1}\right)^{1/2} + \left(\sigma_K(0) - \sqrt{\sigma_K(0)^2 - 1}\right)^{1/2}\right) = \sqrt{\frac{\sigma_K(0)+1}{2}} \tag{75}$$

be the unique positive real number c verifying $T_2(c) = 2c^2 - 1 = \sigma_K(0)$. We end up with

$$x_{2(n+1)} - x_* = \frac{2\,\sigma_2^\Lambda(\lambda)\,T_{2n}(c)}{T_{2(n+1)}(c)}(x_{2n} - x_*) - \frac{T_{2(n-1)}(c)}{T_{2(n+1)}(c)}(x_{2(n-1)} - x_*). \tag{76}$$

Note, the above equation suggests to introduce the sequence $z_l \triangleq T_l(c)(x_l - x_*)$. Indeed, the above equality simplifies

$$z_{2(n+1)} = 2\,\sigma_2^\Lambda(\lambda)\,z_{2n} - z_{2(n-1)}. \tag{77}$$

Let's look for two steps of the cyclical heavy ball method that are together equivalent to (76). We look for an algorithm of the form

$$\forall n \geq 0, x_{n+1} = x_n - h_n\nabla f(x_n) + \frac{T_{n-1}(c)}{T_{n+1}(c)}\left(x_n - x_{n-1}\right), \tag{78}$$

i.e, projecting again onto the eigenspace of eigenvalue λ, we obtain

$$\forall n \geq 0, x_{n+1} - x_* = \left(1 + \frac{T_{n-1}(c)}{T_{n+1}(c)} - h_n\lambda\right)(x_n - x_*) - \frac{T_{n-1}(c)}{T_{n+1}(c)}\left(x_{n-1} - x_*\right). \tag{79}$$

Here we introduce the notation

$$\omega_l \triangleq \left(1 + \frac{T_{l-1}(c)}{T_{l+1}(c)}\right) = 2c\frac{T_l(c)}{T_{l+1}(c)}, \tag{80}$$

and the change of variable

$$\tilde{h_l} \triangleq \frac{h_l}{\omega_l}. \tag{81}$$

We rewrite (79) in terms of the sequence z and using the sequence $\tilde{h}$,

$$\forall n \geq 0, z_{n+1} = T_{n+1}(c)\left(1 + \frac{T_{n-1}(c)}{T_{n+1}(c)} - h_n\lambda\right)(x_n - x_*) - z_{n-1} \tag{82}$$

$$= \left(2cT_n(c)(1 - \tilde{h}_n\lambda)\right)(x_n - x_*) - z_{n-1} \tag{83}$$

$$= \left(2c(1 - \tilde{h}_n\lambda)\right) z_n - z_{n-1}. \tag{84}$$

We now need to find the right sequence $\tilde{h}_n$ such that we recover eq. (77). Combining the 2 following

$$z_{2n+1} = \left(2c(1 - \tilde{h}_{2n}\lambda)\right) z_{2n} - z_{2n-1} \tag{85}$$

$$z_{2n+2} = \left(2c(1 - \tilde{h}_{2n+1}\lambda)\right) z_{2n+1} - z_{2n} \tag{86}$$

by isolating the odd index in the second equation and plugging it in the first one, we get

$$z_{2n+2} = \left(4c^2(1 - \tilde{h}_{2n}\lambda)(1 - \tilde{h}_{2n+1}\lambda) - 1 - \frac{2c(1 - \tilde{h}_{2n+1}\lambda)}{2c(1 - \tilde{h}_{2n-1}\lambda)}\right) z_{2n} - \frac{2c(1 - \tilde{h}_{2n+1}\lambda)}{2c(1 - \tilde{h}_{2n-1}\lambda)} z_{2n-2}. \tag{87}$$

We need to identify

$$2\sigma_2^\Lambda(\lambda) = 4c^2(1 - \tilde{h}_{2n}\lambda)(1 - \tilde{h}_{2n+1}\lambda) - 1 - \frac{2c(1 - \tilde{h}_{2n+1}\lambda)}{2c(1 - \tilde{h}_{2n-1}\lambda)}, \tag{88}$$

$$1 = \frac{2c(1 - \tilde{h}_{2n+1}\lambda)}{2c(1 - \tilde{h}_{2n-1}\lambda)}. \tag{89}$$

Hence, we conclude from the second equation that $\tilde{h}_{2n+1} = \tilde{h}_{2n-1} = \tilde{h}_1$ is independent of n. And the first equation then becomes

$$2\sigma_2^\Lambda(\lambda) = 4c^2(1 - \tilde{h}_{2n}\lambda)(1 - \tilde{h}_1\lambda) - 2 \tag{90}$$

leading also to $\tilde{h}_{2n}$ independent of n. We observe an alternating strategy of the "pseudo-step-sizes" $\tilde{h}_0$ and $\tilde{h}_1$. Finally, we must fix them to

$$\sigma_2^\Lambda(\lambda) = 2c^2(1 - \tilde{h}_0\lambda)(1 - \tilde{h}_1\lambda) - 1. \tag{91}$$

Note this is possible because the equation above is valid for $\lambda = 0$ for any choice of $\tilde{h}_0$ and $\tilde{h}_1$ and the polynomial $\sigma_2^\Lambda + 1$ can be defined by its value in 0 and its roots that are exactly $\frac{1}{\tilde{h}_0}$ and $\frac{1}{\tilde{h}_1}$. And from (155), those values are $\boldsymbol{\mu}_2$ and $\boldsymbol{L}_1$, which gives the values $\tilde{h}_0 = \frac{1}{\boldsymbol{L}_1}$ and $\tilde{h}_1 = \frac{1}{\boldsymbol{\mu}_2}$.

We now sum up what we have so far. Setting c, $\tilde{h}_0$ and $\tilde{h}_1$ as described above, the iterations

$$\forall n \geq 1, x_{n+1} = x_n - \left(1 + \frac{T_{n-1}(c)}{T_{n+1}(c)}\right)\tilde{h}_{\mathrm{mod}(n,2)}\nabla f(x_n) + \frac{T_{n-1}(c)}{T_{n+1}(c)}(x_n - x_{n-1}) \tag{92}$$

lead to the recursion (77).

Let define $x_1 = x_0 - \tilde{h}_0\nabla f(x_0)$, and from the above

$$x_2 = x_1 - \left(1 + \frac{1}{2c^2 - 1}\right)\tilde{h}_1\lambda(x_1 - x_*) + \frac{1}{2c^2 - 1}(x_1 - x_0) \tag{93}$$

$$x_2 - x_* = \frac{2c^2}{\sigma_2^\Lambda(0)}\left(1 - \tilde{h}_1\lambda\right)(x_1 - x_*) - \frac{1}{\sigma_2^\Lambda(0)}(x_0 - x_*) \tag{94}$$

$$= \frac{2c^2}{\sigma_2^\Lambda(0)}\left(1 - \tilde{h}_1\lambda\right)\left(1 - \tilde{h}_0\lambda\right)(x_0 - x_*) - \frac{1}{\sigma_2^\Lambda(0)}(x_0 - x_*) \tag{95}$$

$$= \frac{\sigma_2^\Lambda(\lambda)}{\sigma_2^\Lambda(0)}(x_0 - x_*) \tag{96}$$

$$z_2 = \sigma_2^\Lambda(\lambda). \tag{97}$$

Finally, the sequence z_{2n} is defined by

$$z_0 = 1, \tag{98}$$
$$z_1 = \sigma_2^\Lambda(\lambda), \tag{99}$$
$$z_{2(n+1)} = 2\,\sigma_2^\Lambda(\lambda)\, z_{2n} - z_{2(n-1)}. \tag{100}$$

which defines exactly $T_n(\sigma_2^\Lambda(\lambda))$. We conclude $x_{2n} - x_* = \frac{T_n(\sigma_2^\Lambda(\lambda))}{T_n(\sigma_2^\Lambda(0))}(x_0 - x_*)$.

We sum up the algorithm used to reach the above equality:

$$x_1 = x_0 - \tilde{h}_0 \nabla f(x_0), \tag{101}$$
$$\forall n \geq 0, x_{n+1} = x_n - \omega_n \tilde{h}_n \nabla f(x_n) + (\omega_n - 1)(x_n - x_{n-1})\ . \tag{102}$$

with $\omega_n = \left(1 + \frac{T_{n-1}(c)}{T_{n+1}(c)}\right) = \frac{2cT_n(c)}{T_{n+1}(c)}$. Note the recursion

$$\omega_n^{-1} = \frac{T_{n+1}(c)}{2cT_n(c)} \tag{103}$$
$$= \frac{2cT_n(c) - T_{n-1}(c)}{2cT_n(c)} \tag{104}$$
$$= 1 - \frac{T_{n-1}(c)}{2cT_n(c)} \tag{105}$$
$$= 1 - \frac{1}{4c^2}\frac{2cT_{n-1}(c)}{T_n(c)} \tag{106}$$
$$= 1 - \frac{1}{4c^2}\omega_{n-1}. \tag{107}$$

Finally, the sequence ω can be computed online using the recursion

$$\omega_n = \frac{1}{1 - \frac{1}{4c^2}\omega_{n-1}} \tag{108}$$

with $\omega_0 = 2$. □

In this appendix, as well as in Appendix B, we end up with some equality of the form

$$\|x_t - x_*\| = \frac{T_n(\sigma_K(H))}{T_n(\sigma_K(0))}\|x_0 - x_*\|\,. \tag{109}$$

The next theorem explains how to derive the rate factor from it.

Proposition 4.6. *For a given σ_K such that $\sup_{\lambda\in\Lambda}|\sigma_K(\lambda)| = 1$, the asymptotic rate factor τ^{σ_K} of the method associated to the polynomial (14) is*

$$1 - \tau^{\sigma_K} = \lim_{t\to\infty}\sqrt[t]{\sup_{\lambda\in\Lambda}|P_t(\lambda;\sigma_K)|} = \left(\sigma_0 - \sqrt{\sigma_0^2 - 1}\right)^{\frac{1}{K}}\,,$$
$$\text{with } \sigma_0 \triangleq \sigma_K(0)\,. \tag{15}$$

Proof. We observe that the rate factor of the method is upper bounded by

$$\sqrt[t]{\sup_{\lambda\in\Lambda}|Z_t^\Lambda(\lambda)|} = \sqrt[t]{\sup_{\lambda\in\Lambda}\left|\frac{T_{t/K}\left(\sigma_K^\Lambda(\lambda)\right)}{T_{t/K}\left(\sigma_0\right)}\right|} = \sqrt[t]{\frac{1}{|T_{t/K}\left(\sigma_0\right)|}} \quad \text{if } \sup_{\lambda\in\Lambda}|\sigma_K(\lambda)| = 1. \tag{110}$$

Since $\sigma_0 > 1$, and by using the explicit formula of Chebyshev polynomials, we have that

$$T_{t/K}\left(\sigma_0\right) = \frac{\left(\sigma_0 + \sqrt{\sigma_0^2-1}\right)^{t/K} + \left(\sigma_0 - \sqrt{\sigma_0^2-1}\right)^{t/K}}{2} \underset{t\to\infty}{\sim} \frac{\left(\sigma_0 + \sqrt{\sigma_0^2-1}\right)^{t/K}}{2}. \tag{111}$$

Algorithm 7:
Cyclical heavy ball $\mathrm{HB}_K(h_0, \ldots, h_{K-1}; m)$

Input: Initialization x_0, momentum $m \in (0,1)$, step-sizes $\{h_0, \ldots, h_{K-1}\}$
$x_1 = x_0 - \frac{h_0}{1+m}\nabla f(x_0)$
for $t = 1, 2, \ldots$ **do**

$$x_{t+1} = x_t - h_{\mathrm{mod}(t,K)}\nabla f(x_t) + m(x_t - x_{t-1})$$

end

Taking the limit gives

$$\lim_{t\to\infty} \sqrt[t]{\frac{1}{|T_{t/K}(\sigma_0)|}} = \left(\frac{1}{\sigma_0 + \sqrt{\sigma_0^2 - 1}}\right)^{\frac{1}{K}} = \left(\sigma_0 - \sqrt{\sigma_0^2 - 1}\right)^{\frac{1}{K}}. \tag{112}$$

□

D.2 Derivation of heavy ball with K step-sizes cycle

In this section, we consider heavy ball algorithm with a cycle of K different step-sizes. For convenience, we restate Algorithm 1 below.

We first recall the convergence theorem 4.8 stated in Section 4.3.

Theorem 4.8. *With an arbitrary momentum m and an arbitrary sequence of step-sizes $\{h_i\}$, the worst-case rate of convergence $1-\tau$ of Algorithm 1 on $\mathcal{C}_\Lambda$ is*

$$\begin{cases} \sqrt{m} & \text{if } \sigma_* \leq 1 \\ \sqrt{m}\left(\sigma_* + \sqrt{\sigma_*^2 - 1}\right)^{K^{-1}} & \text{if } \sigma_* \in \left(1, \ \dfrac{1+m^K}{2\left(\sqrt{m}\right)^K}\right) \\ \geq 1 \, (\textit{no convergence}) & \text{if } \sigma_* \geq \dfrac{1+m^K}{2\left(\sqrt{m}\right)^K}, \end{cases} \tag{19}$$

where $\sigma_* \triangleq \sup_{\lambda\in\Lambda} |\sigma(\lambda; \{h_i\}, m)|$, $\sigma(\lambda; \{h_i\}, m)$ *is the K-degree polynomial*

$$\sigma(\lambda; \{h_i\}, m) \triangleq \frac{1}{2}\mathrm{Tr}\left(M_1 M_2 \ldots M_K\right), \tag{20}$$

and $M_i = \begin{bmatrix} \frac{1+m-h_{K-i}\lambda}{\sqrt{m}} & -1 \\ 1 & 0 \end{bmatrix}$.

Proof. Note a first trick. Let's define $x_{-1} \triangleq x_0 - \dfrac{h_0}{1+m}\nabla f(x_0)$. This way, $x_{t+1} = x_t - h_{\mathrm{mod}(t,K)}\nabla f(x_t) + m(x_t - x_{t-1})$ holds for any $t \geq 0$ (including $t = 0$).

Now, let's introduce the polynomials P_t defined by Proposition 4.1 as $x_t - x_* = P_t(H)(x_0 - x_*)$. From now, in order to highlight the K-cyclic behavior, we introduce the indexation $t = nK + r$, with $r \in [\![0, K-1]\!]$.

We verify the following:

$$P_{-1}(\lambda) = 1 - \frac{h_0\lambda}{1+m}, \tag{113}$$

$$P_0(\lambda) = 1, \tag{114}$$

$$\forall n \geq 0, r \in [\![0, K-1]\!], \quad P_{nK+r+1}(\lambda) = (1 + m - h_r\lambda)P_{nK+r}(\lambda) - mP_{nK+r-1}(\lambda). \tag{115}$$

In order to get rid of the last occurrence of m in equation above, we introduce $\tilde{P}_t(\lambda) \triangleq \frac{1}{\left(\sqrt{m}\right)^t}P_t(\lambda)$.

This way, the above can be written

$$\tilde{P}_{-1}(\lambda) = \sqrt{m}\left(1 - \frac{h_0\lambda}{1+m}\right) = \frac{2m}{1+m}\sigma_0(\lambda), \tag{116}$$

$$\tilde{P}_0(\lambda) = 1, \tag{117}$$

$$\forall n \geq 0, r \in [\![0, K-1]\!], \quad \tilde{P}_{nK+r+1}(\lambda) = \frac{1+m-h_r\lambda}{\sqrt{m}}\tilde{P}_{nK+r}(\lambda) - \tilde{P}_{nK+r-1}(\lambda). \tag{118}$$

In the following, we want to determine a formulation for the polynomials $\tilde{P}_{nK}$. In order to do so, we introduce the following operator:

$$A(\lambda) \triangleq \begin{pmatrix} \frac{1+m-h_{K-1}\lambda}{\sqrt{m}} & -1 \\ 1 & 0 \end{pmatrix} \cdots \begin{pmatrix} \frac{1+m-h_0\lambda}{\sqrt{m}} & -1 \\ 1 & 0 \end{pmatrix} \triangleq \begin{pmatrix} a(\lambda) & b(\lambda) \\ c(\lambda) & d(\lambda) \end{pmatrix} \tag{119}$$

as well as the scalar valued function

$$\sigma(\lambda; \{h_i\}, m) \triangleq \frac{1}{2}\mathrm{Tr}(A(\lambda))\,. \tag{120}$$

This operator comes naturally in

$$\begin{pmatrix} \tilde{P}_{(n+1)K}(\lambda) \\ \tilde{P}_{(n+1)K-1}(\lambda) \end{pmatrix} = \begin{pmatrix} \frac{1+m-h_{K-1}\lambda}{\sqrt{m}} & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \tilde{P}_{(n+1)K-1}(\lambda) \\ \tilde{P}_{(n+1)K-2}(\lambda) \end{pmatrix} \tag{121}$$

$$= \begin{pmatrix} \frac{1+m-h_{K-1}\lambda}{\sqrt{m}} & -1 \\ 1 & 0 \end{pmatrix} \cdots \begin{pmatrix} \frac{1+m-h_0\lambda}{\sqrt{m}} & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \tilde{P}_{nK}(\lambda) \\ \tilde{P}_{nK-1}(\lambda) \end{pmatrix} \tag{122}$$

$$= A(\lambda) \begin{pmatrix} \tilde{P}_{nK}(\lambda) \\ \tilde{P}_{nK-1}(\lambda) \end{pmatrix}. \tag{123}$$

Looking K steps at a time makes the analysis much easier as the process applying K steps is then homogeneous (we apply A and A doesn't depend on the index of the iterate).

$$\tilde{P}_{(n+1)K}(\lambda) = a(\lambda)\tilde{P}_{nK}(\lambda) + b(\lambda)\tilde{P}_{nK-1}(\lambda), \tag{124}$$

$$\tilde{P}_{(n+1)K-1}(\lambda) = c(\lambda)\tilde{P}_{nK}(\lambda) + d(\lambda)\tilde{P}_{nK-1}(\lambda). \tag{125}$$

Combining the two above equations (First one with incremented n + $b(\lambda)$ times the second one - $d(\lambda)$ times the first one) leads to

$$\tilde{P}_{(n+2)K}(\lambda) = (a(\lambda)+d(\lambda))\tilde{P}_{(n+1)K}(\lambda) - (a(\lambda)d(\lambda) - b(\lambda)c(\lambda))\tilde{P}_{nK}(\lambda) \tag{126}$$

$$= 2\sigma(\lambda; \{h_i\}, m)\tilde{P}_{(n+1)K}(\lambda) - \tilde{P}_{nK}(\lambda) \tag{127}$$

where the second inequality is deduced after we recognize

$$a(\lambda) + d(\lambda) = \mathrm{Tr}(A(\lambda)) = 2\sigma(\lambda; \{h_i\}, m) \tag{128}$$

and

$$a(\lambda)d(\lambda) - b(\lambda)c(\lambda) = \mathrm{Det}(A(\lambda)) = 1 \tag{129}$$

($A(\lambda)$ is the product of matrices of determinant 1).

In equation (127) we recognize the recursion verified by e.g. $(T_n(\sigma(\lambda; \{h_i\}, m)))_{n\in\mathbb{N}}$, or $(U_n(\sigma(\lambda; \{h_i\}, m)))_{n\in\mathbb{N}}$, where T_n (resp. U_n) denotes the first (resp. second) type Chebyshev polynomial of degree n.

Moreover we verify the initialization

$$\begin{aligned}\tilde{P}_0(\lambda) &= 1, && (130)\\ \tilde{P}_K(\lambda) &= a(\lambda)\tilde{P}_0(\lambda) + b(\lambda)\tilde{P}_{-1}(\lambda) && (131)\\ &= a(\lambda) + b(\lambda)\frac{m}{1+m}\frac{1+m-h_0\lambda}{\sqrt{m}}. && (132)\end{aligned}$$

We also notice that

$$U_n(\sigma(\lambda;\{h_i\},m)) + \left(b(\lambda)\frac{m}{1+m}\frac{1+m-h_0\lambda}{\sqrt{m}} - d(\lambda)\right)U_{n-1}(\sigma(\lambda;\{h_i\},m)) \tag{133}$$

verifies the same recursion of order 2 than $\tilde{P}_{Kn}$ as well as the same 2 initial terms.

Finally, we conclude

$$\tilde{P}_{nK}(\lambda) = U_n(\sigma(\lambda;\{h_i\},m)) + \left(b(\lambda)\frac{m}{1+m}\frac{1+m-h_0\lambda}{\sqrt{m}} - d(\lambda)\right)U_{n-1}(\sigma(\lambda;\{h_i\},m)) \tag{134}$$

and

$$P_{nK}(\lambda) = \left(\sqrt{m}\right)^{nK}\tilde{P}_{nK}(\lambda). \tag{135}$$

Now we have the full expression of the polynomials associated to algorithm 1. Then we can study it's rate of convergence.

Note for any $r \in [\![0, K-1]\!]$, we can have a similar expression of the form

$$P_{nK+r}(\lambda) = \left(\sqrt{m}\right)^{nK}\left(Q_r^1(\lambda)U_n(\sigma(\lambda;\{h_i\},m)) + Q_r^2(\lambda)U_{n-1}(\sigma(\lambda;\{h_i\},m))\right) \tag{136}$$

with Q_r^1 and Q_r^2 some fixed polynomials. This is the consequence of the fact that all sequences $\tilde{P}_{nK+r}(\lambda)$ verify the same recursion formula. Only initialization are different.

In order to study the factor rate of this algorithm, let's first introduce M an upper bound of all the $|Q_r^i|$. For instance, let M defined as follow.

$$M = \max_{r\in[\![0,K-1]\!],i\in\{1,2\}}\sup_{\lambda\in\Lambda}|Q_r^i(\lambda)|. \tag{137}$$

Then,

$$\begin{aligned}\|x_t - x_*\| &\le \sup_{\lambda\in\Lambda}|P_t(\lambda)|\|x_0 - x_*\| && (138)\\ &\le M\left(\sqrt{m}\right)^t\left(\sup_{\lambda\in\Lambda}|U_n(\sigma(\lambda;\{h_i\},m))| + \sup_{\lambda\in\Lambda}|U_{n-1}(\sigma(\lambda;\{h_i\},m))|\right)\|x_0 - x_*\|, && (139)\end{aligned}$$

with $n = \lfloor \frac{t}{K} \rfloor$.

Set $\sigma_{\sup} \triangleq \sup_{\lambda\in\Lambda}|\sigma(\lambda;\{h_i\},m)|$. The worst-case rate verifies

$$\text{If } \sigma_{\sup} \le 1, \text{ then } r_t \le M\left(\sqrt{m}\right)^t(n+1+n) = O\left(t\left(\sqrt{m}\right)^t\right). \tag{140}$$

$$\text{If } \sigma_{\sup} > 1, \text{ then } r_t = O\left(\left(\sqrt{m}\right)^t\left(\sigma_{\sup} + \sqrt{\sigma_{\sup}^2 - 1}\right)^n\right). \tag{141}$$

The first case analysis comes from the fact that U_n is bounded by $n+1$ on $[-1,1]$, while the second cases analysis comes from the fact that $U_n(x)$ grows exponentially fast outside of $[-1,1]$ at a rate $x + \sqrt{x^2-1}$.

Then the factor rate verifies

$$\text{If } \sigma_{\text{sup}} \leq 1,\ 1-\tau = \sqrt{m}. \tag{142}$$

$$\text{If } \sigma_{\text{sup}} > 1,\ 1-\tau = \sqrt{m}\left(\sigma_{\text{sup}} + \sqrt{\sigma_{\text{sup}}^2 - 1}\right)^{1/K}. \tag{143}$$

It remains to notice that $\sqrt{m}\left(\sigma_{\text{sup}} + \sqrt{\sigma_{\text{sup}}^2 - 1}\right)^{1/K} < 1$ is equivalent to $\sigma_{\text{sup}} < \frac{1+m^k}{2\left(\sqrt{m}\right)^k}$.

□

From this factor rate analysis, we can state Proposition 4.9 of Section 4.3.

Proposition 4.9. *Let $\sigma(\lambda; \{h_i\}, m)$ be the polynomial defined by (20), and σ_K^Λ be the optimal link function of degree K defined by (16). If the momentum m and the sequence of step-sizes $\{h_i\}$ satisfy*

$$\sigma(\lambda; \{h_i\}, m) = \sigma_K^\Lambda(\lambda)\,, \tag{21}$$

*then **1)** the parameters are optimal, in the sense that they minimize the asymptotic rate factor from Theorem 4.8, **2)** the optimal momentum parameter is*

$$m = \left(\sigma_0 - \sqrt{\sigma_0^2 - 1}\right)^{2/K}, \quad \textit{where } \sigma_0 = \sigma_K^\Lambda(0)\,, \tag{22}$$

***3)** the iterates from Algo. 3 with parameters $\{h_i\}$ and m form a polynomial with recurrence (18), and **4)** Algorithm 3 achieves the worst-case rate $r_t^{\text{Alg. 3}}$ and the asymptotic rate factor $1-\tau^{\text{Alg. 3}}$*

$$r_t^{\text{Alg. 3}} = O\left(t\left(\sigma_0 - \sqrt{\sigma_0^2 - 1}\right)^{t/K}\right), \tag{23}$$

$$1 - \tau^{\text{Alg. 3}} = \left(\sigma_0 - \sqrt{\sigma_0^2 - 1}\right)^{1/K}.$$

Proof. For now we don't assume assumption 21 yet. Set $\sigma_0 \triangleq \sigma(0; \{h_i\}, m)$. Then, by definition (20) of $\sigma(\lambda; \{h_i\}, m)$,

$$\sigma_0 = \frac{1}{2}\text{Tr}\left(\begin{bmatrix} \frac{1+m}{\sqrt{m}} & -1 \\ 1 & 0 \end{bmatrix}^K\right) = T_K\left(\frac{1+m}{2\sqrt{m}}\right) = \frac{1+m^K}{2\left(\sqrt{m}\right)^K}. \tag{144}$$

Hence, reversing this equality,

$$\sqrt{m} = \left(\sigma_0 - \sqrt{\sigma_0^2 - 1}\right)^{\frac{1}{K}}. \tag{145}$$

From Theorem 4.8, we therefore know

$$\text{If } \sigma_{\text{sup}} \leq 1,\ 1-\tau = \left(\sigma_0 - \sqrt{\sigma_0^2 - 1}\right)^{\frac{1}{K}}. \tag{146}$$

$$\text{If } \sigma_{\text{sup}} > 1,\ 1-\tau = \left(\sigma_0 - \sqrt{\sigma_0^2 - 1}\right)^{\frac{1}{K}}\left(\sigma_{\text{sup}} + \sqrt{\sigma_{\text{sup}}^2 - 1}\right)^{1/K}. \tag{147}$$

But, one can check that

$$\left(\sigma_0 - \sqrt{\sigma_0^2 - 1}\right)^{\frac{1}{K}}\left(\sigma_{\text{sup}} + \sqrt{\sigma_{\text{sup}}^2 - 1}\right)^{1/K} \geq \left(\frac{\sigma_0}{\sigma_{\text{sup}}} - \sqrt{\left(\frac{\sigma_0}{\sigma_{\text{sup}}}\right)^2 - 1}\right)^{\frac{1}{K}} \tag{148}$$

which shows that a tuning generating the polynomial $\frac{\sigma(\lambda; \{h_i\}, m)}{\sigma_{\text{sup}}}$ would lead to a better convergence rate. Hence, we should look for polynomials $\sigma(\lambda; \{h_i\}, m)$ verifying $\sigma_{\text{sup}} \leq 1$. And then,

$$1 - \tau = \sqrt{m} = \left(\sigma_0 - \sqrt{\sigma_0^2 - 1}\right)^{\frac{1}{K}}. \tag{149}$$

which explain we aim at maximizing σ_0 subject to $\sigma_{\text{sup}} \leq 1$ ((16)).

Finally, we proved **1)**: if $\sigma(\lambda; \{h_i\}, m) = \sigma_K^\Lambda(\lambda)$, then the tuning is optimal in the sense that this is the one that minimizes the asymptotic rate factor among all K steps-sizes based tuning.

From now, we assume

$$\sigma(\lambda; \{h_i\}, m) = \sigma_K^\Lambda(\lambda). \tag{150}$$

Therefore,

$$\sigma_0 = \sigma_K^\Lambda(0) \tag{151}$$

and **2)** is already proven above.

3) follows directly from the definition of $\sigma_K^\Lambda(\lambda)$.

Finally, since $\sigma_{\text{sup}} \leq 1$, we know

$$1 - \tau = \sqrt{m} = \left(\sigma_0 - \sqrt{\sigma_0^2 - 1}\right)^{1/K} \tag{152}$$

which proves part of **4)**.

To prove the expression of the worst-case rate r_t, we need to apply the intermediate result (140) instead of Theorem 4.8.

□

D.3 Example: alternating step-sizes ($K = 2$)

Proposition D.2. *The strategy with 2 step-sizes is optimal on the union of two intervals if and only if they have the same length.*

Proof. This is a direct consequence of Theorem C.2, which implies $\sigma_2^\Lambda(\boldsymbol{\mu}_1) = \sigma_2^\Lambda(\boldsymbol{L}_2) = 1$ and $\sigma_2^\Lambda(\boldsymbol{\mu}_2) = \sigma_2^\Lambda(\boldsymbol{L}_1) = -1$.

This is feasible if and only if $\boldsymbol{L}_2 - \boldsymbol{\mu}_2 = \boldsymbol{L}_1 - \boldsymbol{\mu}_1$ since σ_2^Λ is a degree 2 polynomial.

Indeed, set $\sigma_2^\Lambda(x) = a(x-b)^2 + c$. Then, $\sigma_2^\Lambda(\boldsymbol{\mu}_1) = \sigma_2^\Lambda(\boldsymbol{L}_2)$ implies $a(\boldsymbol{\mu}_1 - b)^2 + c = a(\boldsymbol{L}_2 - b)^2 + c$, then $|\boldsymbol{\mu}_1 - b| = |\boldsymbol{L}_2 - b|$ and finally $b = \frac{\boldsymbol{\mu}_1 + \boldsymbol{L}_2}{2}$. Similarly, $\sigma_2^\Lambda(\boldsymbol{\mu}_2) = \sigma_2^\Lambda(\boldsymbol{L}_1)$ implies $b = \frac{\boldsymbol{\mu}_2 + \boldsymbol{L}_1}{2}$.

We conclude $\frac{\boldsymbol{\mu}_1 + \boldsymbol{L}_2}{2} = \frac{\boldsymbol{\mu}_2 + \boldsymbol{L}_1}{2}$, and $\boldsymbol{L}_2 - \boldsymbol{\mu}_2 = \boldsymbol{L}_1 - \boldsymbol{\mu}_1$. □

Proposition 4.5. *Let $\Lambda = [\boldsymbol{\mu}_1, \boldsymbol{L}_1] \cup [\boldsymbol{\mu}_2, \boldsymbol{L}_2]$ be an union of two intervals of the same size ($\boldsymbol{L}_1 - \boldsymbol{\mu}_1 = \boldsymbol{L}_2 - \boldsymbol{\mu}_2$) and let m, h_0, h_1 be as defined in Algorithm 2. Then the minimax polynomial (solution to (12)) is, for all $t = 2n$, $n \in \mathbb{N}_0^+$,*

$$\frac{T_n\left(\sigma_2^\Lambda(\lambda)\right)}{T_n\left(\sigma_2^\Lambda(0)\right)} = \underset{\substack{P \in \mathbb{R}_t[X], \\ P(0)=1}}{\arg\min} \sup_{\lambda \in \Lambda} |P(\lambda)| \,,$$

$$\textit{with} \quad \sigma_2^\Lambda(\lambda) = \frac{1}{2m}\left(1 + m - \lambda h_0\right)\left(1 + m - \lambda h_1\right) - 1 \,.$$

Proof. From Theorem C.2,

$$\sigma_2^\Lambda(\boldsymbol{\mu}_1) = 1, \tag{153}$$
$$\sigma_2^\Lambda(\boldsymbol{L}_1) = -1, \tag{154}$$
$$\sigma_2^\Lambda(\boldsymbol{\mu}_2) = -1, \tag{155}$$
$$\sigma_2^\Lambda(\boldsymbol{L}_2) = 1, \tag{156}$$

and this implies that $\frac{T_n\left(\sigma_2^\Lambda(\lambda)\right)}{T_n\left(\sigma_2^\Lambda(0)\right)}$ is optimal.

In particular, L_1 and μ_2 are roots of $\sigma_2^\Lambda + 1$. Therefore, we know there exists a constant c such that $\sigma_2^\Lambda(\lambda) = c(1 - \frac{\lambda}{L_1})(1 - \frac{\lambda}{\mu_2}) - 1$. Moreover, evaluating this in μ_1 gives $\sigma_2^\Lambda(\mu_1) = c(1 - \frac{\mu_1}{L_1})(1 - \frac{\mu_1}{\mu_2}) - 1 = 1$, so

$$
\begin{aligned}
c &= \frac{2}{(1 - \frac{\mu_1}{L_1})(1 - \frac{\mu_1}{\mu_2})} && (157)\\
&= \frac{2 L_1 \mu_2}{(L_1 - \mu_1)(\mu_2 - \mu_1)} && (158)\\
&= 2\frac{\left(\frac{\mu_1 + L_2}{2}\right)^2 - R^2\left(\frac{L_2 - \mu_1}{2}\right)^2}{\frac{1-R^2}{4}(L_2 - \mu_1)^2} && (159)\\
&= 2\frac{\rho^2 - R^2}{1 - R^2}. && (160)
\end{aligned}
$$

Then,

$$\sigma_2^\Lambda(\lambda) = 2\frac{\rho^2 - R^2}{1 - R^2}(1 - \frac{\lambda}{L_1})(1 - \frac{\lambda}{\mu_2}) - 1 \tag{161}$$

which can be written

$$\sigma_2^\Lambda(\lambda) = 2\left(\frac{1+m}{2\sqrt{m}}\right)^2\left(1 - \frac{\lambda}{L_1}\right)\left(1 - \frac{\lambda}{\mu_2}\right) - 1 \tag{162}$$

with $\left(\frac{1+m}{2\sqrt{m}}\right)^2 = \frac{\rho^2 - R^2}{1-R^2}$. Finally, $m = \left(\frac{\sqrt{\rho^2 - R^2} - \sqrt{\rho^2 - 1}}{\sqrt{1-R^2}}\right)^2$. ∎

Theorem 3.1 (Rate factor of $\mathrm{HB}_2(h_0, h_1; m)$)**.** *Let $f \in \mathcal{C}_\Lambda$ and consider the cyclical heavy ball method with step-sizes h_0, h_1 and momentum parameter m. The asymptotic rate factor of Algorithm 1 with cycles of length two is*

$$1 - \tau = \begin{cases} \sqrt{m} & \text{if } \sigma_* \leq 1, \\ \sqrt{m}\left(\sigma_* + \sqrt{\sigma_*^2 - 1}\right)^{\frac{1}{2}} & \text{if } \sigma_* \in \left(1, \, \frac{1+m^2}{2m}\right), \\ \geq 1 \text{ (no convergence)} & \text{if } \sigma_* \geq \frac{1+m^2}{2m}, \end{cases}$$

$$\textit{with} \quad \sigma_* = \max_{\lambda \in \left\{\mu_1, L_1, \mu_2, L_2, (1+m)\frac{h_0 + h_1}{2h_0 h_1}\right\} \cap \Lambda} |\sigma_2(\lambda)|$$

$$\textit{and} \quad \sigma_2(\lambda) = 2\left(\frac{1 + m - \lambda h_0}{2\sqrt{m}}\right)\left(\frac{1 + m - \lambda h_1}{2\sqrt{m}}\right) - 1\,.$$

Proof. From Theorem 4.8 applied to $K = 2$, we immediately have the above result with

$$\sigma_{\sup} = \sup_{\lambda \in \Lambda}\left|2\left(\frac{1 + m - \lambda h_0}{2\sqrt{m}}\right)\left(\frac{1 + m - \lambda h_1}{2\sqrt{m}}\right) - 1\right|.$$

To conclude the proof, we need to prove that the optimal value of $|\sigma_2^\Lambda|$ can only be reached on $\left\{\mu_1, L_1, \mu_2, L_2, (1+m)\frac{h_0+h_1}{2h_0h_1}\right\}$. Indeed, σ_2^Λ being convex, its maximal value can only be reached on $\{\mu_1, L_2\}$. Its minimal value is reached on $(1+m)\frac{h_0+h_1}{2h_0h_1}$. Therefore, over Λ, the minimal value of σ_2^Λ is reached on $(1+m)\frac{h_0+h_1}{2h_0h_1}$ if the latest belongs to Λ. Otherwise, its minimal value is reached to the closest point in Λ to $(1+m)\frac{h_0+h_1}{2h_0h_1}$, namely, it can be any point of $\{\mu_1, L_1, \mu_2, L_2\}$. ∎

Proposition D.3 (Residual polynomial in the robust region)**.** *Assuming $\sigma_2^\Lambda(\lambda) \geq -1, \quad \forall \lambda \in \Lambda$, the residual polynomial associated with the cyclical heavy ball algorithm is*

$$
\begin{aligned}
P_{2n}(\lambda) = m^n \Bigg[&\frac{2m}{1+m} T_{2n}\left(\sqrt{\left(\frac{1 + m - \lambda h_0}{2\sqrt{m}}\right)\left(\frac{1 + m - \lambda h_1}{2\sqrt{m}}\right)}\right) \\
&+ \frac{1-m}{1+m} U_{2n}\left(\sqrt{\left(\frac{1 + m - \lambda h_0}{2\sqrt{m}}\right)\left(\frac{1 + m - \lambda h_1}{2\sqrt{m}}\right)}\right)\Bigg].
\end{aligned} \tag{163}
$$

Remark D.4. The assumption $\sigma_2(\lambda) \geq -1, \quad \forall \lambda \in \Lambda$ is verified in the robust region, and is useful here because the term $\left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)\left(\frac{1+m-\lambda h_1}{2\sqrt{m}}\right)$ is equal to $\frac{1+\sigma_2(\lambda)}{2}$ and must be positive to make the above expression well defined. Otherwise the result can hold replacing the square root with some complex number, but it brings no value when we derive the convergence rate from it.

Proof. This proof reuses elements of the proof of Theorem (4.8), especially Equation (127). For sake of completeness and simplicity, we prove this result again directly in the special case $K = 2$.

We first recall the recursion of Algorithm 1 for $K = 2$. For sake of simplicity, we directly projet it onto the eigenspace associated to the eigenvalue λ of the Hessian of the objective function.

$$\begin{aligned} x_{2n+1} - x_* &= (1 + m - h_0\lambda)(x_{2n} - x_*) - m(x_{2n-1} - x_*). \\ x_{2n+2} - x_* &= (1 + m - h_1\lambda)(x_{2n+1} - x_*) - m(x_{2n} - x_*). \end{aligned} \tag{164}$$

Identifying $x_t - x_* = P_t(\lambda)(x_0 - x_*)$ and $P_t(\lambda) = \left(\sqrt{m}\right)^t \tilde{P}_t(\lambda)$,

$$\begin{aligned} \tilde{P}_{2n+1}(\lambda) &= \tfrac{1+m-h_0\lambda}{\sqrt{m}}\tilde{P}_{2n}(\lambda) - \tilde{P}_{2n-1}(\lambda), \\ \tilde{P}_{2n+2}(\lambda) &= \tfrac{1+m-h_1\lambda}{\sqrt{m}}\tilde{P}_{2n+1}(\lambda) - \tilde{P}_{2n}(\lambda). \end{aligned} \tag{165}$$

Multiplying the first equation by $\frac{1+m-h_1\lambda}{\sqrt{m}}$ and replacing $\frac{1+m-h_1\lambda}{\sqrt{m}}\tilde{P}_{2n+1}(\lambda)$ and $\frac{1+m-h_1\lambda}{\sqrt{m}}\tilde{P}_{2n-1}(\lambda)$ accordingly to the second equation leads to

$$\tilde{P}_{2n+2}(\lambda) + \tilde{P}_{2n}(\lambda) = \frac{1+m-h_0\lambda}{\sqrt{m}}\frac{1+m-h_1\lambda}{\sqrt{m}}\tilde{P}_{2n}(\lambda) - \left(\tilde{P}_{2n}(\lambda) + \tilde{P}_{2n-2}(\lambda)\right) \tag{166}$$

which can be written as in equation (127)

$$\tilde{P}_{2n+2}(\lambda) = \left(\frac{1+m-h_0\lambda}{\sqrt{m}}\frac{1+m-h_1\lambda}{\sqrt{m}} - 2\right)\tilde{P}_{2n}(\lambda) - \tilde{P}_{2n-2}(\lambda). \tag{167}$$

Moreover,

$$\begin{aligned} x_1 - x_* &= (1 - \tfrac{h_0}{1+m}\lambda)(x_0 - x_*), \\ x_2 - x_* &= (1 + m - h_1\lambda)(x_1 - x_*) - m(x_0 - x_*), \end{aligned} \tag{168}$$

leading to the initialization

$$\begin{aligned} \tilde{P}_1(\lambda) &= \tfrac{1}{\sqrt{m}}(1 - \tfrac{h_0}{1+m}\lambda)\tilde{P}_0(\lambda), \\ \tilde{P}_2(\lambda) &= \tfrac{1+m-h_1\lambda}{\sqrt{m}}\tilde{P}_1(\lambda) - \tilde{P}_0(\lambda). \end{aligned} \tag{169}$$

hence,

$$\tilde{P}_2(\lambda) = \left(\frac{1}{1+m}\frac{1+m-h_0\lambda}{\sqrt{m}}\frac{1+m-h_1\lambda}{\sqrt{m}} - 1\right) \tag{170}$$

and recall

$$\tilde{P}_0(\lambda) = 1. \tag{171}$$

It remains to notice that

$$\begin{aligned} &\frac{2m}{1+m}T_{2n}\left(\sqrt{\left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)\left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)}\right) \\ &+\frac{1-m}{1+m}U_{2n}\left(\sqrt{\left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)\left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)}\right) \end{aligned} \tag{172}$$

verifies the same recursion as well as the same initialization for $n = 0$ and $n = 1$. This allows us to identify the 2 sequences of polynomials

$$\begin{aligned} \tilde{P}_{2n}(\lambda) = &\frac{2m}{1+m}T_{2n}\left(\sqrt{\left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)\left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)}\right) \\ &+ \frac{1-m}{1+m}U_{2n}\left(\sqrt{\left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)\left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)}\right) \end{aligned} \tag{173}$$

which concludes the proof.

□

Corollary 3.2. *The non-asymptotic and asymptotic worst-case rates $r_t^{Alg.\ 2}$ and $1-\tau^{Alg.\ 2}$ of Algorithm 2 over $\mathcal{C}_\Lambda$ for even iteration number t are*

$$r_t^{Alg.\ 2} = \left(\tfrac{\sqrt{\rho^2-R^2}-\sqrt{\rho^2-1}}{\sqrt{1-R^2}}\right)^t \left(1+t\sqrt{\tfrac{\rho^2-1}{\rho^2-R^2}}\right),$$
$$1-\tau^{Alg.\ 2} = \frac{\sqrt{\rho^2-R^2}-\sqrt{\rho^2-1}}{\sqrt{1-R^2}}.$$

Proof. From Proposition 4.5, Algorithm 2's parameter make $\sigma(\lambda;\{h_i\},m)=\sigma_2^\Lambda$. In particular, by definition,

$$-1 \le 2\left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)\left(\frac{1+m-\lambda h_1}{2\sqrt{m}}\right)-1 \le 1. \tag{174}$$

and then

$$0 \le \sqrt{\left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)\left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)} \le 1. \tag{175}$$

And we know that $\forall x \le 1$, $T_n(x)\le 1$ and $U_n(x)\le n+1$.

Therefore, using optimal parameters, and from Proposition D.3

$$\tilde{P}_{2n}(\lambda) \le \frac{2m}{1+m} + (2n+1)\frac{1-m}{1+m} = 1+2n\frac{1-m}{1+m}. \tag{176}$$

And the worst-case rate is then upper bounded

$$r_t = \left(1+t\frac{1-m}{1+m}\right)\left(\sqrt{m}\right)^t \tag{177}$$

for all t even.

It remains to plug m expression into the above to conclude. □

Note that in the proof above, all the expressions are symmetric in (h_0,h_1), which implies that swapping those 2 step-sizes doesn't impact this statement.

Remark D.5. The previous statement provides the convergence rate of Algorithm 2. It does not state that this is the optimal way to tune Algorithm 1, but comparing the obtained rate to the one of Line 6 does. Another way to derive the optimal parameters, is to start from the result of Theorem 3.1 applied on a 2 step-sizes strategy, or from the result of Proposition D.3. This leads to minimizing m under the constraints that $\zeta(\lambda) \triangleq \left(\frac{1+m-\lambda h_0}{2\sqrt{m}}\right)\left(\frac{1+m-\lambda h_1}{2\sqrt{m}}\right)$ has values between 0 and 1 on $\Lambda=[\mu_1, L_1]\cup[\mu_2, L_2]$. By symmetry of Λ and the convex parabola ζ, we know that optimal parameters verify $\zeta(L_1)=\zeta(\mu_2)=0$. And therefore, $\zeta(\mu_1)=\zeta(L_2)=1$ maximizes the range of allowed m. This way we recover the tuning of Algorithm 2. Note that ζ is related to $\sigma_2^{(\Lambda)}$ through the relation $\sigma_2^{(\Lambda)}=2\zeta-1$, and therefore the 4 mentioned equalities are equivalent to the equioscillation property.

The next theorem sums up the results of Proposition 3.3 and Table 1.

Theorem D.6 (Asymptotic speedup of HB with alternating step-sizes)**.**

1. *Let $R\in[0,1)$ be a fixed number, then* $\sqrt{m} \underset{\kappa\to 0}{=} 1-\frac{2\sqrt{\kappa}}{\sqrt{1-R^2}}+o(\sqrt{\kappa})$.

2. *Let*
$$R(\kappa) \underset{\kappa\to 0}{=} 1-\frac{\sqrt{\kappa}}{2}+o(\sqrt{\kappa}), \qquad i.e., \quad \Lambda \approx [\mu, \mu+\frac{\sqrt{\mu L}}{4}]\cup[L-\frac{\sqrt{\mu L}}{4}, L],$$
then $\sqrt{m} \underset{\kappa\to 0}{=} 1-2\sqrt[4]{\kappa}+o(\sqrt[4]{\kappa})$, *therefore leasing to a new square root acceleration.*

3. Let

$$R(\kappa) \underset{\kappa\to 0}{=} 1 - 2\gamma\kappa + o(\kappa), \qquad \text{i.e.,} \quad \Lambda \approx [\mu, (1+\gamma)\mu] \cup [L - \gamma\mu, L],$$

then $\sqrt{m} \underset{\kappa\to 0}{=} \sqrt{1 + \frac{1}{\gamma}} - \sqrt{\frac{1}{\gamma}} + o(\kappa)$, *therefore leading to a constant complexity.*

This is summed up in the Table 2.

Relative gap R	Set Λ	Rate factor τ	Speedup τ/τ^{PHB}
$R \in [0,1)$	$[\mu, \mu + \frac{1-R}{2}(L-\mu)] \cup [L - \frac{1-R}{2}(L-\mu), L]$	$\frac{2\sqrt{\kappa}}{\sqrt{1-R^2}}$	$(1-R^2)^{-\frac{1}{2}}$
$R = 1 - \sqrt{\kappa}/2$	$[\mu, \mu + \frac{\sqrt{\mu L}}{4}] \cup [L - \frac{\sqrt{\mu L}}{4}, L]$	$2\sqrt[4]{\kappa}$	$\kappa^{-\frac{1}{4}}$
$R = 1 - 2\gamma\kappa$	$[\mu, (1+\gamma)\mu] \cup [L - \gamma\mu, L]$	indep. of κ	$O(\kappa^{-\frac{1}{2}})$

Table 2: Case study of the convergence of Algorithm 2 as a function of R, in the regime where $\kappa \to 0$. The **first line** corresponds to a situation where R is independent of κ, and we observe a constant gain w.r.t. heavy ball. The **second line** study a setting in which R depends on $\sqrt{\kappa}$, meaning the two intervals in Λ are relatively small. The asymptotic rate reads $(1 - 2\sqrt[4]{\kappa})^t$, beating the $(1 - 2\sqrt{\kappa})^t$ lower bound. Finally, in the **third line**, R depends on κ, the two intervals in Λ are so small that the convergence becomes $O(1)$, i.e., is independent of κ.

Proof.

1. Let $R \in [0,1)$. The momentum m satisfies

$$\begin{aligned}
\sqrt{m} &\underset{\kappa\to 0}{=} \frac{\sqrt{1 + O(\kappa) - R^2} - \sqrt{4\kappa + O(\kappa^2)}}{\sqrt{1-R^2}} \\
&\underset{\kappa\to 0}{=} \frac{\sqrt{1-R^2} + O(\kappa) - 2\sqrt{\kappa} + O(\kappa)}{\sqrt{1-R^2}} \\
&\underset{\kappa\to 0}{=} 1 - \frac{2\sqrt{\kappa}}{\sqrt{1-R^2}} + O(\kappa).
\end{aligned}$$

2. Let $R(\kappa) \underset{\kappa\to 0}{=} 1 - \frac{\sqrt{\kappa}}{2} + o(\sqrt{\kappa})$. The momentum m verifies

$$\begin{aligned}
\sqrt{m} &= \sqrt{\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - R^2}{1-R^2}} - \sqrt{\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - 1}{1-R^2}} \\
&= \sqrt{\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - 1}{1-R^2} + 1} - \sqrt{\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - 1}{1-R^2}}.
\end{aligned}$$

We first focus on

$$\begin{aligned}
\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - 1}{1-R^2} &\underset{\kappa\to 0}{=} \frac{4\kappa + O(\kappa^2)}{\sqrt{\kappa} + o(\sqrt{\kappa})} \\
&\underset{\kappa\to 0}{=} 4\sqrt{\kappa} + o(\sqrt{\kappa}).
\end{aligned}$$

Then,

$$\begin{aligned}
\sqrt{m} &= \sqrt{\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - 1}{1-R^2} + 1} - \sqrt{\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - 1}{1-R^2}} \\
&\underset{\kappa\to 0}{=} \sqrt{1 + 4\sqrt{\kappa} + o(\sqrt{\kappa})} - \sqrt{4\sqrt{\kappa} + o(\sqrt{\kappa})} \\
&\underset{\kappa\to 0}{=} 1 + 2\sqrt{\kappa} + o(\sqrt{\kappa}) - 2\sqrt[4]{\kappa} + o(\sqrt[4]{\kappa}) \\
&\underset{\kappa\to 0}{=} 1 - 2\sqrt[4]{\kappa} + o(\sqrt[4]{\kappa}).
\end{aligned}$$

3. Let $R(\kappa) \underset{\kappa\to 0}{=} 1 - 2\gamma\kappa + o(\kappa)$. The momentum m verifies

$$\begin{aligned}
\sqrt{m} &= \sqrt{\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - R^2}{1-R^2}} - \sqrt{\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - 1}{1-R^2}} \\
&= \sqrt{\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - 1}{1-R^2} + 1} - \sqrt{\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - 1}{1-R^2}}.
\end{aligned}$$

We first focus on

$$\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - 1}{1-R^2} \underset{\kappa\to 0}{=} \frac{4\kappa + O(\kappa^2)}{4\gamma\kappa + o(\kappa)} \underset{\kappa\to 0}{=} \frac{1}{\gamma} + o(\kappa).$$

Then,

$$\begin{aligned}
\sqrt{m} &= \sqrt{\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - 1}{1-R^2} + 1} - \sqrt{\frac{\left(\frac{1+\kappa}{1-\kappa}\right)^2 - 1}{1-R^2}} \\
&\underset{\kappa\to 0}{=} \sqrt{1 + \frac{1}{\gamma} + o(\kappa)} - \sqrt{\frac{1}{\gamma} + o(\kappa)} \\
&\underset{\kappa\to 0}{=} \sqrt{1 + \frac{1}{\gamma}} - \sqrt{\frac{1}{\gamma}} + o(\kappa).
\end{aligned}$$

□

D.4 Example: 3 cycling step-sizes

Proposition D.7. *The strategy with 3 step-sizes is optimal on the union of two intervals if and only if they are of the form*

$$\left[\mu, \mu + (L-\mu)\left(\frac{1}{2} - \frac{R}{2} + \frac{1-R^2}{4}\right)\right] \cup \left[L - (L-\mu)\left(\frac{1}{2} - \frac{R}{2} - \frac{1-R^2}{4}\right), L\right],$$

for some $R \in [0, 1]$.

Proof. From Theorem C.2, we know that $T_n(\sigma_3)$ is optimal for all n if and only if, Λ is the union of 3 different intervals that are mapped on $[-1, 1]$. Since, we are looking for Λ being the union of 2 intervals, we know 2 of the 3 intervals Λ is composed of share an extremity. Recall $\Lambda = [\mu_1, L_1] \cup [\mu_2, L_2]$. By symmetry, we can assume without loss of generality that $[\mu_1, L_1]$ is mapped to $[-1, 1]$ twice, and $[\mu_2, L_2]$ once. Let's then introduce $x \in (\mu_1, L_1)$ and say:

$$\sigma_3(\mu_1) = 1, \tag{178}$$

$$\sigma_3(x) = -1, \tag{179}$$

$$\sigma_3(L_1) = 1, \tag{180}$$

$$\sigma_3(\mu_2) = 1, \tag{181}$$

$$\sigma_3(L_2) = -1. \tag{182}$$

Note we also know that x is a local minima of σ_3, leading to $\sigma_3'(x) = 0$. We now know 3 roots of $\sigma_3 + 1$ and 3 roots of $\sigma_3 - 1$, leading to:

$$\sigma_3(\lambda) - 1 = c(\lambda - \mu_1)(\lambda - L_1)(\lambda - \mu_2), \tag{183}$$

$$\sigma_3(\lambda) + 1 = c(\lambda - x)^2(\lambda - L_2), \tag{184}$$

for some non-zero constant c. Here, we want to remove the dependency in x or c. Using the two equalities above,

$$(\lambda - x)^2(\lambda - L_2) - (\lambda - \mu_1)(\lambda - L_1)(\lambda - \mu_2) = \frac{2}{c}. \tag{185}$$

Matching the coefficients of the above polynomial leads to

$$2x + L_2 = \mu_1 + L_1 + \mu_2 \tag{186}$$

$$\text{and} \tag{187}$$

$$2xL_2 + x^2 = \mu_1 L_1 + \mu_1\mu_2 + L_1\mu_2. \tag{188}$$

We plug the expression of x we get from the first equality into the second one,

$$L_2(\mu_1 + L_1 + \mu_2 - L_2) + \left(\frac{\mu_1 + L_1 + \mu_2 - L_2}{2}\right)^2 = \mu_1 L_1 + \mu_1\mu_2 + L_1\mu_2. \tag{189}$$

From here, for simplicity, we define

$$r_i \triangleq \frac{L_i - \mu_i}{L_2 - \mu_1}, \qquad \text{for } i \in \{1, 2\}. \tag{190}$$

Replacing L_1 and μ_2 by their expression using μ_1, L_2, r_1 and r_2 leads to

$$r_1 = 2\sqrt{r_2} - r_2. \tag{191}$$

The reciprocal holds and we can find x using Equation (186) or (188). Note if Equation (191) holds, we can directly express σ_3 as the unique polynomial verifying

$$\sigma_3(\mu_1) = 1, \tag{192}$$

$$\sigma_3(L_1) = 1, \tag{193}$$

$$\sigma_3(\mu_2) = 1, \tag{194}$$

$$\sigma_3(L_2) = -1. \tag{195}$$

We can therefore conclude

$$\sigma_3(\lambda) = 1 - 2\frac{(\lambda - \mu_1)(\lambda - L_1)(\lambda - \mu_2)}{(L_2 - \mu_1)(L_2 - L_1)(L_2 - \mu_2)}. \tag{196}$$

From the new notations $r_1, r_2, \mu = \mu_1, L = L_2$, we know $T_n(\sigma_3^\Lambda)$ is optimal for all n if and only if

$$\Lambda = [\mu, \mu + r_1(L - \mu)] \cup [L - r_2(L - \mu), L]. \tag{197}$$

Let R be

$$R = \frac{\mu_2 - L_1}{L_2 - \mu_1} \tag{198}$$

as in the 2 step-sizes setting. Here, we have $R = 1 - r_1 - r_2$ and we assume $r_1 = 2\sqrt{r_2} - r_2$. Combining those 2 equalities gives:

$$r_1 = \frac{1}{2} - \frac{R}{2} + \frac{1 - R^2}{4}, \tag{199}$$

$$r_2 = \frac{1}{2} - \frac{R}{2} - \frac{1 - R^2}{4}, \tag{200}$$

leading to the desired result, i.e.,

$$\Lambda = [\mu, \mu + (L-\mu)(\frac{1}{2} - \frac{R}{2} + \frac{1-R^2}{4})] \cup [L - (L-\mu)(\frac{1}{2} - \frac{R}{2} - \frac{1-R^2}{4}), L].$$

□

Theorem D.8 (Asymptotic speedup of heavy ball when cycling over 3 step-sizes)**.** *Let $R \in [0,1)$ be a fixed number, then*

$$\sqrt{m} \underset{\kappa\to 0}{=} 1 - 2\sqrt{\kappa}\sqrt{\frac{1-R^2/9}{1-R^2}} + o(\sqrt{\kappa}). \tag{201}$$

Proof. From Equation (145),

$$\sqrt{m} = \left(\sigma_3^{(\Lambda)}(0) - \sqrt{\sigma_3^{(\Lambda)}(0)^2 - 1}\right)^{\frac{1}{3}} \quad \text{with} \;\; \sigma_3^{(\Lambda)}(0) = 1 + 2\frac{\mu_1 L_1 \mu_2}{(L_2-\mu_1)(L_2-L_1)(L_2-\mu_2)}.$$

Using the previous notations,

$$\mu = \mu_1, \tag{202}$$

$$L = L_2, \tag{203}$$

$$\kappa = \frac{\mu}{L}, \tag{204}$$

$$r_i \triangleq \frac{L_i - \mu_i}{L_2 - \mu_1}, \qquad \text{for } i \in \{1, 2\}, \tag{205}$$

we can write $\sigma_3^{(\Lambda)}$ as

$$\sigma_3^{(\Lambda)}(0) = 1 + 2\frac{\mu_1 L_1 \mu_2}{(L_2-\mu_1)(L_2-L_1)(L_2-\mu_2)}, \tag{206}$$

$$= 1 + 2\frac{\kappa(\kappa + r_1(1-\kappa))(1 - r_2(1-\kappa))}{(1-\kappa)^3(1-r_1)r_2}, \tag{207}$$

$$\underset{\kappa\to 0}{=} 1 + 2\kappa\frac{r_1(1-r_2)}{(1-r_1)r_2}, \tag{208}$$

$$= 1 + 2\kappa\frac{\left(\frac{1}{2} - \frac{R}{2} + \frac{1-R^2}{4}\right)\left(\frac{1}{2} + \frac{R}{2} - \frac{1-R^2}{4}\right)}{\left(\frac{1}{2} + \frac{R}{2} + \frac{1-R^2}{4}\right)\left(\frac{1}{2} - \frac{R}{2} - \frac{1-R^2}{4}\right)}, \tag{209}$$

$$= 1 + 2\kappa\frac{9 - 10R^2 + R^4}{1 - 2R^2 + R^4}, \tag{210}$$

$$= 1 + 2\kappa\frac{\left(1-R^2\right)\left(9-R^2\right)}{\left(1-R^2\right)^2}, \tag{211}$$

$$= 1 + 2\kappa\frac{9-R^2}{1-R^2}. \tag{212}$$

Then introducing briefly $\varepsilon \triangleq \kappa\frac{9-R^2}{1-R^2} \underset{\kappa\to 0}{\to} 0$,

$$\sqrt{m} = \left(\sigma_3^{(\Lambda)}(0) - \sqrt{\sigma_3^{(\Lambda)}(0)^2 - 1}\right)^{\frac{1}{3}}, \tag{213}$$

$$= \left(1 + 2\varepsilon - \sqrt{1 + 4\varepsilon + 4\varepsilon^2 - 1}\right)^{\frac{1}{3}}, \tag{214}$$

$$\underset{\kappa\to 0}{=} 1 - \frac{2}{3}\sqrt{\varepsilon} + o(\sqrt{\varepsilon}). \tag{215}$$

Plugging ε expression into the latest gives

$$\sqrt{m} \underset{\kappa \to 0}{=} 1 - 2\sqrt{\kappa}\sqrt{\frac{1 - R^2/9}{1 - R^2}} + o(\sqrt{\kappa}). \tag{216}$$

□

E Beyond quadratic objective: local convergence of cycling methods

In this section, we prove a result of local convergence of the cyclical heavy ball method out of quadratic setting. We first recall the Theorem 5.1 stated in Section 5:

Theorem 5.1 (Local convergence)**.** *Let $f : \mathbb{R}^d \mapsto \mathbb{R}$ be a twice continuously differentiable function, x_* a local minimizer, and H be the Hessian of f at x_* with $Sp(H) \subseteq \Lambda$. Let x_t denote the result of running Algorithm 1 with parameters $h_1, h_2, \cdots, h_K, m$, and let $1 - \tau$ be the linear convergence rate on the quadratic objective (OPT). Then we have*

$$\begin{aligned} &\forall \varepsilon > 0, \exists \text{ open set } V_\varepsilon : x_0,\, x_* \in V_\varepsilon \\ \implies &\|x_t - x_*\| = O((1 - \tau + \varepsilon)^t)\|x_0 - x_*\|. \end{aligned} \tag{24}$$

Proof. For any k multiple of K, consider S_k the operator applying k steps of cycling Heavy Ball on the iterates x_t and x_{t-1} (note since k is a multiple of K, Algorithm 1 consists in repeating the operator S_k). Namely S_k is an operator on $\mathbb{R}^{2d}$ verifying $S_k((x_t, x_{t-1})) = (x_{t+k}, x_{t+k-1})$. This operator is a composition of gradients of f and affine functions, and so it is continuously differentiable.

Applying the mean value theorem along each coordinate of S_k, we have that there exists a matrix-valued function $M(v_1, v_2)$ for all v_1, v_2 in the domain of S_k such that

$$S_k(v_1) - S_k(v_2) = M(v_1, v_2)(v_1 - v_2)\,, \tag{217}$$

where the i^{th} rows of $M(v_1, v_2)$ is the gradient of the i^{th} output of S_k evaluated at a vector on the segment between v_1 and v_2.

$$M(v_1, v_2) = \begin{pmatrix} \nabla(S_k)_1(w_1)^T \\ \vdots \\ \nabla(S_k)_i(w_i)^T \\ \vdots \\ \nabla(S_k)_{2d}(w_{2d})^T \end{pmatrix} \text{ where } \forall i \in [\![1, 2d]\!], \begin{cases} (S_k)_i \text{ denotes the } i\text{th coordinate of } S_k. \\ \\ w_i \text{ is a point on the segment } [v_1, v_2]. \end{cases} \tag{218}$$

By continuity of those gradients, taking v_1 and v_2 sufficiently close to (x_*, x_*), $M(v_1, v_2)$ can be chosen arbitrarily close to the Jacobian of S_k in (x_*, x_*) denoted by JS_k^*.

Since by assumption the algorithm converges on the quadratic form induced by H at the rate $1 - \tau$, we conclude that the spectral radius of JS_k^* is upper bounded by $1 - \tau$.

From the previous point, we can find a small enough neighborhood of (x_*, x_*) such that $M(v_1, v_2)$ has a spectral radius arbitrarily close to $1 - \tau$, in particular smaller than 1.

Furthermore, it's known for any $\varepsilon > 0$, there exists an operator norm $\|.\|$ such that $\|M(v_1, v_2)\| < 1 - \tau + \varepsilon$. (see e.g. (Bertsekas, 1997, Proposition A.15)).

Hence, for any $\varepsilon > 0$, there exists a neighborhood V of (x_*, x_*) and an operator norm $\|.\|$ as described above such that S_k is a $(1 - \tau + \varepsilon)$-contraction on V for the norm $\|.\|$.

This leads to convergence to the only fixed point (x_*, x_*) with a convergence rate smaller than any $1 - \tau + \varepsilon$.

Moreover, the first step of the Algorithm 1 is continuous with respect to x_0. Hence, for any $V \in \mathbb{R}^{2d}$ neighborhood of (x_*, x_*), there exists $W \in \mathbb{R}^d$ a neighborhood of x_*, such that

$$x_0 \in W \implies (x_1, x_0) \in V. \tag{219}$$

Finally, for any $\varepsilon > 0$, there exists W a neighborhood of x_* such that the Algorithm 1 converges to x_* with a rate smaller than $1 - \tau + \varepsilon$.

□

F Experimental setup

Benchmarks we run using a Google colab public instance with a single CPU. Producing the results of Figure 4 took 50 minutes with this setup. The code to reproduce this figure is attached with the supplementary material in the jupyter notebook benchmarks.ipynb .

G Comparison with Oymak (2021)

The work of Oymak (2021) also exploits cyclical step-sizes for when the spectral structure of the Hessian contains a gap. This work appeared concurrently to the first version of this manuscript and takes a somewhat different stand for exploiting this particular spectral structure. We summarize the main differences in the table below.

	Oymak (2021)	This work
Algorithm	Gradient descent	Heavy ball-type (optimal algorithm)
Cycle length K	K is a function of the spectral assumptions	K is a choice
Structure of the cycle	$\underbrace{\eta_+, \ldots, \eta_+}_{K-1 \text{ times}}, \eta_-$ (Oymak, 2021, Definition 1)	$(h_0, \ldots, h_{K-1})$ (See Algorithm 1)
Optimal among chosen scheme	Not proven / discussed	Yes
Spectral assumption	Bimodal (2 intervals)	Any number of intervals
Spectral assumption in bimodal case	Rate depends on $\frac{L_2}{\mu_2}$ and $\frac{L_1}{\mu_1}$	Rate depends on $L_2 - \mu_2$ and $L_1 - \mu_1$
Typical application case in bimodal	Very strong assumption on $L_1 - \mu_1$ and very weak assumption on $L_2 - \mu_2$.	Weak assumption on both $L_2 - \mu_2$ and $L_1 - \mu_1$, more in line with empirical observations (Papyan, 2018; Ghorbani et al., 2019).
Spectrum is a single interval	Does not recover original rate	Recovers rate and optimal method
Convergence beyond quadratic objectives	Yes (with extra assumptions)	Local convergence

We emphasize the following points.

- While we provide Theorem 4.8 which described the convergence rate of any cycling heavy ball (for any cycle), Oymak (2021) only studied gradient descent method (without momentum) for a particular cycle (for which cycle length is not a parameter, but fixed by the eigenvalues).
- Moreover, our Theorem 4.9 provides the optimal cycle to use for any choice of a cycle length, while optimality is not discussed in Oymak (2021) and the cycle uses only two different step-sizes, which is somewhat arbitrary.
- Furthermore, our work highlights acceleration under assumptions that seem more aligned with empirical

observation: Oymak (2021) shows that $L_1 - \mu_1$ needs to be very small for his strategy to be worthwhile, while this is not really the case in our experiments (see Figure 1)

- Finally, in the general case in which we cannot assume any gap in the spectrum, we naturally recover the classical optimal method and rate. This is not the case in Oymak (2021) which is suboptimal in this setup.

In this work, we focus on quadratic minimization and give some local convergence guarantee beyond quadratics. On the other hand, Oymak (2021) provides guarantee beyond this setup, at the cost of very restrictive assumptions.