

Acceleration by Stepsize Hedging I: Multi-Step Descent and the Silver Stepsize Schedule

Jason M. Altschuler
UPenn
alts@upenn.edu

Pablo A. Parrilo
LIDS - MIT
parrilo@mit.edu

September 15, 2023

Abstract

Can we accelerate convergence of gradient descent without changing the algorithm—just by carefully choosing stepsizes? Surprisingly, we show that the answer is yes. Our proposed *Silver Stepsize Schedule* optimizes strongly convex functions in $\kappa^{\log_\rho 2} \approx \kappa^{0.7864}$ iterations, where $\rho = 1 + \sqrt{2}$ is the silver ratio and κ is the condition number. This is intermediate between the textbook unaccelerated rate κ and the accelerated rate $\kappa^{1/2}$ due to Nesterov in 1983. The non-strongly convex setting is conceptually identical, and standard black-box reductions imply an analogous partially accelerated rate $\varepsilon^{-\log_\rho 2} \approx \varepsilon^{-0.7864}$. We conjecture and provide partial evidence that these rates are optimal among all stepsize schedules.

The Silver Stepsize Schedule is constructed recursively in a fully explicit way. It is non-monotonic, fractal-like, and approximately periodic of period $\kappa^{\log_\rho 2}$. This leads to a phase transition in the convergence rate: initially super-exponential (acceleration regime), then exponential (saturation regime).

The core algorithmic intuition is *hedging* between individually suboptimal strategies—short steps and long steps—since bad cases for the former are good cases for the latter, and vice versa. Properly combining these stepsizes yields faster convergence due to the misalignment of worst-case functions. The key challenge in proving this speedup is enforcing long-range consistency conditions along the algorithm’s trajectory. We do this by developing a technique that recursively glues constraints from different portions of the trajectory, thus removing a key stumbling block in previous analyses of optimization algorithms. More broadly, we believe that the concepts of hedging and multi-step descent have the potential to be powerful algorithmic paradigms in a variety of contexts in optimization and beyond.

This series of papers publishes and extends the first author’s 2018 Master’s Thesis (advised by the second author)—which established for the first time that carefully choosing stepsizes can enable acceleration in convex optimization. Prior to this thesis, the only such result was for the special case of quadratic optimization, due to Young in 1953.

Contents

1	Introduction	3
1.1	Contribution and discussion	5
1.2	Related work	9
1.3	Organization	11
2	Conceptual overview: two-step case ($n = 2$)	11
2.1	Optimal stepsizes for quadratic optimization	12
2.2	Optimal stepsizes for convex optimization	13
3	Silver Stepsize Schedule	16
3.1	Normalized Silver Stepsizes	16
3.2	Silver Stepsizes	17
3.3	Silver Stepsize Schedule	18
3.4	Silver Convergence Rate	19
4	Analysis of the Silver Convergence Rate	19
4.1	Heuristic derivation	20
4.2	Rigorous derivation	21
5	Certificate of the Silver Convergence Rate	22
5.1	Recursive gluing	24
5.2	Certificate verification	25
6	Future work	26
A	Deferred details for §4	28
B	Deferred details for §5	28
B.1	Helper lemma: co-coercivities involving x^*	30
B.2	Helper lemma: identities involving q_i	31
B.3	Recursive gluing as a rational function of z_n, y_{2n}, z_{2n}	33
B.4	Recursive gluing as a succinct quadratic form	33
	References	35

1 Introduction

Gradient descent (GD) is a simple iterative algorithm to minimize an objective function f by producing better and better estimates via the update

$$x_{t+1} = x_t - \alpha_t \nabla f(x_t), \quad \forall t = 0, 1, 2, \dots \quad (1.1)$$

GD dates back nearly two hundred years to the work of Cauchy [14], yet it (and its variants) remain a primary workhorse in modern optimization, engineering, and machine learning due to the practical efficacy, simplicity, and scalability. It is of both theoretical and practical importance to analyze the convergence of GD and moreover to optimize parameters so that this convergence is as fast as possible.

A central fact in convex optimization is that with a prudent choice of the stepsize schedule $\{\alpha_t\}$ —the only¹ parameters of the algorithm—running GD from any initialization x_0 produces iterates which optimize f to arbitrary accuracy. Quantifying this statement leads to two intertwined questions: How fast does x_n converge to a minimizer x^* of f ? And what stepsize choice $\{\alpha_t\}$ leads to the fastest convergence rate?

This series of papers revisits these classical questions in the fundamental setting of smooth² convex optimization. Our overarching goal is to understand how much mileage can be obtained by simply optimizing the stepsize choice for GD.

Note that this is markedly different from the past forty years of literature on accelerating the convergence rate for GD. That literature—starting from Nesterov’s seminal work in 1983 [46]—achieves faster convergence rates by modifying the GD algorithm with extra building blocks such as momentum, auxiliary sequences, or other internal dynamics. See the related work section or the recent survey [30]. In contrast, we investigate the basic question of: can we accelerate convergence without changing the GD algorithm—just by optimizing the stepsizes?

Mainstream approach. The standard analysis of GD uses a constant stepsize schedule, i.e., $\alpha_t = \bar{\alpha}$ for all iterations t ; see e.g. the textbooks [10, 11, 12, 33, 42, 45, 51] among many others. For example, $\bar{\alpha} = 1/M$ in the setting of M -smooth convex objectives, or $\bar{\alpha} = 2/(M+m)$ if the objectives are additionally m -strongly convex. This prescription is based on the following fact:

For one iteration of GD, there is a unique stepsize $\bar{\alpha}$ achieving the fastest convergence rate
(in the worst case over functions and initializations). (1.2)

This is provably correct. For example, in the strongly convex setting, this $\bar{\alpha}$ provides the optimal contraction rate—a larger stepsize $\alpha_t > \bar{\alpha}$ can lead to overshooting the target x^* , and a smaller stepsize $\alpha_t < \bar{\alpha}$ can lead to undershooting x^* .

However, it is well-known that even after optimizing the constant $\bar{\alpha}$, this constant stepsize schedule leads to a slow convergence rate. (Hence the intensive research on accelerated GD.) Moreover, even though many alternative stepsize schedules have been proposed in both theory and practice—e.g., exact line search, Armijo-Goldstein rules, Polyak-type schedules, Barzilai-Borwein-type schedules, etc., see the related work section—none of these alternative schedules have led to an analysis that outperforms the slow “unaccelerated” rate of constant stepsize GD. Conventional wisdom therefore dictates that slow convergence is unavoidable, unless one modifies GD by adding extra building blocks beyond choosing stepsizes, e.g., via momentum.

¹In convex optimization, we typically view the initialization x_0 as part of the problem instance rather than a parameter choice, since $x_0 = 0$ without loss of generality after a possible translation of the objective function f .

²In the non-smooth setting, it is classically known that acceleration is impossible, and moreover GD achieves the minimax-optimal convergence rate with simple monotonically decaying stepsize schedules like $\alpha_t \asymp 1/\sqrt{t}$ [45].

	Quadratic	Convex
Mainstream stepsizes	$\Theta(\kappa)$ by constant stepsizes (folklore)	$\Theta(\kappa)$ by constant stepsizes (folklore)
Additional dynamics	$\Theta(\sqrt{\kappa})$ by Heavy Ball [50]	$\Theta(\sqrt{\kappa})$ by Nesterov Acceleration [46]
Hedged stepsizes	$\Theta(\sqrt{\kappa})$ by Chebyshev Stepsizes [69]	$\Theta(\kappa^{\log \rho/2})$ by Silver Stepsizes (Theorem 1.1)

TABLE 1: Iteration complexity of various approaches for minimizing a κ -conditioned function. The dependence on the accuracy ε is omitted as it is always $\log 1/\varepsilon$. Mainstream stepsize schedules require $\Theta(\kappa)$ iterations; this is the textbook unaccelerated rate. For the special case of quadratics (left), accelerated rates of $\Theta(\sqrt{\kappa})$ can be equivalently achieved via Young’s 1953 Chebyshev Stepsize Schedule [69] or Polyak’s 1964 Heavy Ball Algorithm [50]. For the general case of convex functions (right), this equivalence between internal dynamics and varying stepsizes is false. Acceleration was first achieved by Nesterov’s 1983 Fast Gradient Algorithm [46] and it has long been believed that in the convex setting, any acceleration requires modifying GD by adding internal dynamics, e.g., momentum. We prove that accelerated convex optimization *is* possible by choosing better stepsizes.

Faster convergence via dynamic stepsizes? The premise of this series of papers is that this is wrong. Why might the constant stepsize schedule $\alpha_t = \bar{\alpha}$ be sub-optimal? Certainly it is optimal if GD is only run for $n = 1$ iteration—this is the assertion (1.2). However, it is sub-optimal for n steps of GD, for any $n > 1$. Briefly, this is because the statement for $n = 1$ requires the worst-case problem instance (the objective function f and initialization x_0) to align with the choice of stepsize $\alpha_t \neq \bar{\alpha}$ so that the convergence is slow, and for $n > 1$, the worst-case problem instances for each individual step might not align. This suggests an algorithmic opportunity:

$$\begin{aligned} &\text{Is it possible to combine (individually suboptimal) stepsizes } \alpha_t \neq \bar{\alpha} \\ &\text{to achieve faster convergence for } n > 1 \text{ iterations?} \end{aligned} \tag{1.3}$$

We refer to this algorithmic idea as *hedging* between worst-case problem instances. (See §2 for a fully worked-out example.)

Motivation: the special case of quadratics. Of course, using non-constant stepsizes is not a new idea—for the special case of minimizing *convex quadratics*, it has been known that this enables faster convergence since Young’s seminal paper in 1953 [69]. In particular, for quadratic optimization, the optimal stepsize schedule is not constant, but given by the inverse roots of Chebyshev polynomials; the order of these stepsizes is irrelevant for the convergence rate (assuming exact arithmetic); and the resulting convergence rate is the so-called *accelerated rate* that is optimal among all Krylov-subspace algorithms [44], including even modifications of GD that use momentum or internal dynamics. See the related work section §1.2 for further details.

A longstanding gap between quadratic and convex optimization. However, while the advantage of non-constant stepsizes has been well-understood for quadratic optimization for 70 years (and nowadays is even taught in many introductory optimization courses), it has remained entirely open whether this phenomenon extends to any setting of convex optimization beyond quadratics. In particular, it was unclear whether *any* stepsize schedule could lead to *any* speedup over the textbook GD convergence rate—even by a constant factor.

This gap is due to several reasons. First, many phenomena from the quadratic case are simply false in the setting of general convex optimization: e.g., the stepsize schedule based on roots of the Chebyshev polynomials is provably bad for the convex setting [5, Chapter 8], and the order of the stepsizes dramatically affects the convergence rate in the convex setting [5, Chapter 8]. Second, any approach for establishing the advantage of a non-constant stepsize schedule must track how progress in the current iteration is affected by previous iterations—and this effect of history appeared to

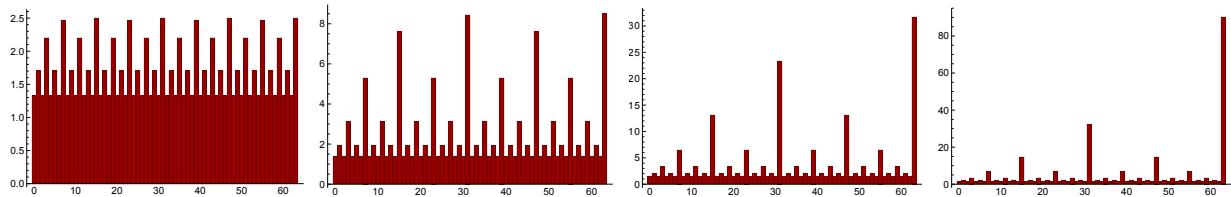


FIGURE 1: Silver Stepsize Schedule, for different condition numbers $\kappa = 4, 16, 64, 256$ – only the first 64 stepsizes are shown. Notice the recursive, fractal behavior and the approximate periodicity with period of size $n^* = \kappa^{\log_2 2}$; details in §1.1.3. Also note the different scales on the vertical axis, since the stepsizes are unnormalized, vs. the normalized stepsizes in Figure 5.

only be explicitly computable in the quadratic setting, essentially since that is the only case in which the GD map is linear (hence tractable to track after repeated iterations).

1.1 Contribution and discussion

In this initial paper, we show that GD can converge faster for smooth convex optimization by using certain time-varying, non-monotonic stepsize schedules. This answers the hedging question (1.3) in the affirmative. This series of papers publishes and extends the first author’s 2018 Master’s Thesis [5] (advised by the second author), which proved such a result for the first time, see the related work section §1.2. In particular, Chapter 8 of the thesis showed for the first time that a constant-factor improvement over the unaccelerated rate is possible in the smooth strongly convex setting, and Chapter 6 of the thesis showed for the first time that an asymptotic acceleration is possible in any setting beyond quadratics. (The latter result proves that arcsine-distributed random stepsizes achieve the fully accelerated rate $\Theta(\sqrt{\kappa} \log 1/\varepsilon)$ if the convex functions are separable; this will be detailed in a forthcoming paper.) Prior to this thesis, the only result for acceleration via choosing stepsizes was for the special case of quadratic optimization, due to Young in 1953.

Conceptually, we deviate from traditional analyses of GD (and other optimization algorithms) by directly analyzing the cumulative progress of all the steps of the algorithm, rather than combining separate bounds for the progress of individual steps. As mentioned above, this global analysis of *multi-step descent* is provably necessary to show any benefit for any deviation from the constant stepsize schedule. Indeed, separately analyzing the progress for each iteration—as done, e.g., in standard GD analyses, in exact line search, or in standard offline-to-online convex optimization reductions—is provably too shortsighted and unavoidably leads to pessimistic, unaccelerated convergence rates. The key difficulty is how to track how different iterations affect progress in other iterations. Previously, this could be accomplished only for the special case of quadratics because then the GD update is linear. We show that this can be accomplished for general convex setting by this by using long-range consistency conditions between the gradients seen along the algorithm’s trajectory. We provide a high-level overview of these new conceptual ideas in §2.

Below, we formally state our main result in §1.1.1, and then discuss the improved convergence rate in §1.1.2, the proposed stepsize schedule in §1.1.3, and the generality of the result in §1.1.4.

1.1.1 Main result: acceleration without momentum

Formalizing this result requires restricting to a function class with controlled curvature. For concreteness, in this first paper we focus on the well-studied setting of strongly convex and smooth f , and we measure progress via distance to the optimum x^* . While smoothness is classically known to be required for acceleration [45], the other choices and assumptions in the theorem statement

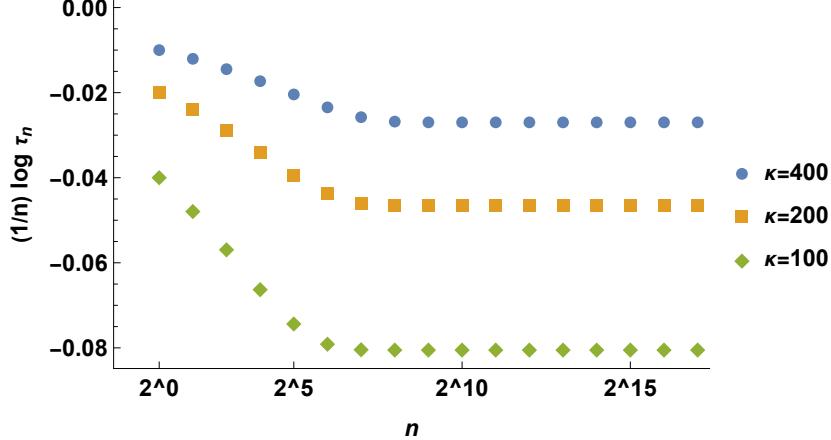


FIGURE 2: Log of the average per-step rate, aka $\frac{1}{n} \log \tau_n$, for varying condition numbers κ . The initial value is the unaccelerated rate $(\frac{\kappa-1}{\kappa+1})^2$. Notice the rate saturation phenomenon that occurs at $n = n^* \asymp \kappa^{\log_\rho 2}$.

are not essential: strong convexity can be relaxed to convexity, and the progress measure can be replaced with other standard desiderata; see the discussion in §1.1.4. Below, let $\rho := 1 + \sqrt{2}$ denote the silver ratio, and assume throughout that f is κ -conditioned, i.e., 1-strongly convex and κ -smooth³—this is without loss of generality after rescaling.

Theorem 1.1. *For any horizon $n \in \mathbb{N}$ that is a power of 2, any dimension d , any κ -conditioned function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, and any initialization x_0 ,*

$$\|x_n - x^*\|^2 \leq \tau_n \|x_0 - x^*\|^2, \quad (1.4)$$

where x^* denotes the unique minimizer of f , x_n denotes the output of n steps of GD using the Silver Stepsize Schedule (defined in §3), and τ_n denotes the n -step Silver Convergence Rate (defined in §3). Moreover, τ_n undergoes the following phase transition at $n^* = \Theta(\kappa^{\log_\rho 2})$:

- Acceleration regime. For $n \leq n^*$,

$$\tau_n = \exp\left(-\Theta\left(\frac{n^{\log_2 \rho}}{\kappa}\right)\right).$$

- Saturation regime. For $n > n^*$,

$$\tau_n = \exp\left(-\Theta\left(\frac{n}{n^*}\right)\right).$$

In particular, in order to achieve a final error $\|x_n - x^*\|^2 \leq \varepsilon$, it suffices to run GD using the Silver Stepsize Schedule for

$$n = \Theta\left(\kappa^{\log_\rho 2} \log \frac{1}{\varepsilon}\right) \approx \Theta\left(\kappa^{0.7864} \log \frac{1}{\varepsilon}\right) \text{ iterations.} \quad (1.5)$$

³Recall that this means f is sandwiched between quadratic lower and upper bounds of curvature 1 and κ , respectively, i.e., $\frac{1}{2}\|v\|^2 \leq f(x+v) - f(x) - \langle v, \nabla f(x) \rangle \leq \frac{\kappa}{2}\|v\|^2$ for all $x, v \in \mathbb{R}^d$. For intuition, this is equivalent to the local curvature bound $I_d \leq \nabla^2 f(x) \leq \kappa I_d$ under the assumption of twice-differentiability (not required by our results).

1.1.2 Discussion of Silver Convergence Rate

Partial acceleration. Our rate $\Theta(\kappa^{\log_\rho 2} \log 1/\varepsilon)$ lies between the textbook rate $\Theta(\kappa \log 1/\varepsilon)$ for GD and the accelerated rate $\Theta(\sqrt{\kappa} \log 1/\varepsilon)$ due to Nesterov in 1983 [46]. We emphasize that before the thesis [5] that this paper is based upon, it was unknown if any improvement over the unaccelerated rate—even a constant factor—was achievable by any stepsize schedule. Our convergence rate is faster than all known GD stepsize schedules for convex optimization, including constant stepsize schedules, Polyak-type schedules, Barzilai-Borwein-type schedules, Goldstein-Armijo-type schedules, exact line search, etc.

Phase transition. A distinctive feature of the Silver Convergence Rate τ_n is that it undergoes a phase transition: τ_n switches from super-exponential to exponential in the horizon n . This transition occurs at $n^* \asymp \kappa^{\log_\rho 2}$, which is the number of iterations required to make the error decrease by a constant factor. See Figure 2. The reason for these two regimes is that beyond n^* , the new stepsizes converge quadratically fast to their stationary value; details in §4.

- Acceleration regime. This regime encapsulates the advantage of multi-step descent: the super-exponentiality of the n -step bound makes it better than composing the 1-step bound n times. This super-exponential regime interpolates the κ dependence between the unaccelerated rate (achieved at $n = 1$) and our partially accelerated rate (achieved at $n \gtrsim n^*$).
- Saturation regime. Here, the benefit of multi-step descent becomes negligible: $\tau_{2n} \approx \tau_n^2$ for $n \geq n^*$.⁴ Briefly, this rate saturation occurs because the Silver Step Schedule is approximately periodic with period n^* , see §1.1.3.

Dimension independence. The convergence rate in Theorem 1.1 is independent of the dimension d and thus can be extended to infinite-dimensional Hilbert space. This is because our analysis only uses consistency conditions for the GD trajectory to arise from a convex function—and these consistency conditions are dimension-independent [54, 63]. This is in common with classical analyses of GD and Nesterov-style acceleration.

Optimality. We conjecture the Silver Step Schedule has the fastest convergence rate among all possible choices of GD stepsize schedules. We prove optimality for the $n = 2$ case of [5, Chapter 8] in §2; this proof readily extends to small n , and we will address the question of optimality for all n in a shortly forthcoming paper.

1.1.3 Discussion of Silver Stepsize Schedule

Recursive construction. The Silver Stepsize Schedule is defined recursively in a fully explicit way. We briefly overview the construction; see §3 for full details. The 1-step schedule $h^{(1)}$ is initialized to the constant $\bar{\alpha} = 2/(1+1/\kappa)$ that is classically known to be optimal for 1-step descent. We then recursively define the $2n$ -step schedule $h^{(2n)}$ as

$$h^{(2n)} := [\tilde{h}^{(n)}, a_{2n}, \tilde{h}^{(n)}, b_{2n}], \quad (1.6)$$

where $\tilde{h}^{(n)}$ is the n -step schedule $h^{(n)}$ with its final stepsize b_n removed, and a_{2n} and b_{2n} are obtained by “splitting” this removed stepsize b_n . Modulo a certain normalizing transformation,

⁴Note that $\tau_{2n} \leq \tau_n^2$; intuitively this amounts to the statement that the optimal $2n$ -step schedule is at least as good as repeating the optimal n -step schedule twice. We call this inequality *rate monotonicity*, see §4. The statement $\tau_{2n} \approx \tau_n^2$ therefore states that this bound is nearly tight.

this splitting produces $a_{2n} < b_n < b_{2n}$ as the roots to a certain quadratic equation in b_n . See §4 for details and closed-form expressions.

Finite-horizon schedule. This recursive construction produces (normalized) stepsize schedules that follow the pattern

$$\begin{aligned} h^{(1)} &= [b_1] \\ h^{(2)} &= [a_2, b_2] \\ h^{(4)} &= [a_2, a_4, a_2, b_4] \\ h^{(8)} &= [a_2, a_4, a_2, a_8, a_2, a_4, a_2, b_8] \end{aligned}$$

See Figure 1 for a visualization.

Infinite-horizon schedule. This schedule simplifies in the limit $n \rightarrow \infty$: the i -th normalized stepsize is given by $a_{B(i)}$, where $B(i)$ denotes the smallest power of 2 in the binary expansion of i . Note that no entries of the b sequence appear.

Fractal order. For the special case of quadratic optimization, the order of the stepsizes is well-known to be irrelevant for the convergence rate. In contrast, in the general setting of convex optimization, the order of the stepsizes provably does matter [5, Chapter 8]. For example, it can be shown that the convergence rate in Theorem 1.1 becomes greater than 1 (i.e., not even contractive) if one reverses the order of the 2-step Silver Stepsize Schedule.

The Silver Stepsize Schedule generates a fractal, see Figure 1. This is due to our recursive construction, and is directly evident from the aforementioned fact that the i -th stepsize depends on the sparsity pattern of the binary expansion of i . This fractal structure aligns with the numerical observations in [19, 32], and is in stark contrast with all classical stepsize schedules which, if time-varying, decay monotonically in the iteration number i , e.g., as $1/i$.

Approximate periodicity. The Silver Stepsize Schedule is not periodic as it is continually changes. However, it is approximately periodic with period $n^* \asymp \kappa^{\log_2 2}$, see Figure 1. This is another facet of the rate saturation phenomenon discussed in §1.1.2. See §4 for details.

Dependence on horizon. Theorem 1.1 is stated for horizons n that are powers of 2. For arbitrary integers n , one can simply run the Silver Stepsize Schedule for the largest power of 2 below n , or better, run for all powers of 2 in the binary expansion of n . This affects the average per-step-rate by only a small constant factor. We moreover conjecture that simply using n steps of the infinite-horizon Silver Stepsize Schedule leads to the same convergence rate modulo a lower-order term. This seems reasonable since only logarithmically many stepsizes are changed, but we have not attempted to prove this. Orthogonally, if the horizon is not set in advance, then one can, e.g., do a “doubling” trick by exploiting the fact that the first $2^i - 1$ stepsizes are identical for all $n \geq 2^i$. Specifically, for each i , decide on iteration $2^i - 1$ whether to stop at $n = 2^i$ iterations, or repeat roughly the same amount of effort and go to $2^{i+1} - 1$ iterations.

1.1.4 Discussion of problem setting

Progress measure. Theorem 4.1 uses distance as the progress measure. This can be replaced by other standard progress measures such as function suboptimality or gradient norm, in the initial or

final condition or both, since these measures are equivalent for κ -conditioned functions. This black-box replacement affects the rate by only a lower-order term. Moreover, this equivalence factor can be avoided by re-doing our analysis in a conceptually identical way for the desired progress measures (possibly also with minor changes to the stepsize schedule; e.g., for gradient norm contraction, it appears that one should reverse the order [5, Chapter 8]).

Smoothness. It is well-known that smoothness is required for acceleration: otherwise, GD cannot be accelerated even with momentum or other internal dynamics [45, Chapter 3].

Convexity. Theorem 1.1 is stated for the strongly convex setting, but this can be relaxed to the non-strongly convex setting. Indeed, all our core conceptual ideas extend: the advantage of time-varying, non-monotonic stepsizes, proving this advantage via multi-step descent rather than iterating the greedy 1-step bound, certifying multi-step descent via recursive gluing, etc. The adaptation requires only minor technical modifications to the stepsize schedule, certificate recursion, and progress measure. These details will appear in a shortly forthcoming paper.

We mention that by standard black-box reductions (see e.g., [3] or [12, page 285]), Theorem 1.1 immediately implies accelerated rates for the (non-strongly) convex setting by running GD with the Silver Stepsize Schedule on a quadratically regularized objective, i.e., $f(\cdot) + \delta \|\cdot - y\|^2$ for appropriate choices of δ and y . This gives an analogous partially accelerated rate of $\varepsilon^{-\log \rho_2} \approx \varepsilon^{-0.7864}$ iterations to obtain ε function suboptimality. This is intermediate between the textbook unaccelerated rate $\Theta(\varepsilon^{-1})$ and Nesterov's accelerated rate $\Theta(\varepsilon^{-1/2})$ from 1983 [46]. This strongly suggests that acceleration in the (non-strongly) convex case surpasses the $\Theta(1/(T \log T))$ conjecture in [32]. The aforementioned forthcoming paper will address this via a direct analysis that bypasses regularization.

1.2 Related work

1.2.1 The special case of quadratic optimization

Three equivalent approaches to acceleration. For quadratic optimization, the GD map becomes linear, which enables three equivalent approaches to acceleration. One approach, taken by Young in 1953 is to choose non-constant stepsizes that are the inverses of the roots of Chebyshev polynomials [69]. A second approach is to use momentum, achieved for example by Hestenes and Stiefel's Conjugate Gradient Method in 1952 [34] and Polyak's Heavy Ball Method in 1964 [50]. This equivalence arises because momentum amounts to a three-term recurrence, which if the coefficients are chosen appropriately, generates the same sequence of Chebyshev polynomials; see e.g. [66, Ch. 5]. A third approach is to use the limiting distribution of the roots of the Chebyshev polynomials: the arcsine distribution [5, 36, 52, 70]. This equivalence is due to the fact that the order of stepsizes does not affect convergence in the quadratic case, thus as the horizon $n \rightarrow \infty$, one might as well draw stepsizes i.i.d. from the equilibrium measure. It is important to emphasize that the elegant equivalences between these three approaches—varying stepsizes, momentum, and equilibrium measures—breaks down beyond the special case of quadratic optimization.

Desiderata beyond fast convergence. The above discussion concerns only the convergence rate, not stability. In settings with noisy gradients or inexact arithmetic, the order of the stepsizes may significantly affect the convergence rate of GD, even for quadratic optimization. This question of stability to roundoff errors was already raised in Young's original paper [69]. In such settings, it is desirable to find permutations of the Chebyshev roots for which GD trajectories are maximally

stable. An effective approach is to interleave the roots of Chebyshev polynomials of increasing degree [40]. This leads to a fractal pattern, superficially similar to our proposed stepsize schedule; see [2] for a recent discussion and additional results. However, we emphasize that this fractal is not only fundamentally different but also arises due to entirely different considerations—stability rather than fast convergence.

Structured quadratics. If the quadratic function’s Hessian has additional spectral structure, then improved results are possible. This is because the different viewpoints discussed above are classically known to extend to this situation via potential theory; see the excellent survey [25] and the references within. This enables further refinements of the methods described above for structured quadratics and sometimes also perturbations away from quadratics; see e.g., [31, 48].

1.2.2 The general case of convex optimization

Unaccelerated GD. For constant stepsize, the optimal convergence rate for GD is $\Theta(\kappa \log 1/\varepsilon)$ in the strongly convex setting, and $\Theta(1/\varepsilon)$ in the convex setting; see, e.g., the textbooks [11, 12, 33, 45, 47, 51]. This is often called the unaccelerated rate for GD. Many alternative stepsize schedules have been proposed in both theory and practice. We highlight several well-studied schedules. One family of well-studied strategies adaptively chooses stepsizes either by minimizing the function value over the line spanned by the gradient. This minimization can be performed exactly via line search [11, 20, 47, 51], or approximately via Goldstein-Armijo-type schedules [45]. Alternatively, it can be done by minimizing the estimated distance to the optimum via Polyak-type schedules [51]. Another family is Barzilai-Borwein-type schedules, which are quasi-Newton methods that approximate the Hessian using the past step’s change in iterate and gradient [7]. None of these strategies are known to accelerate beyond the case of quadratics.

Accelerated GD via internal state. The conventional approach for achieving faster convergence is to consider variations of GD that use auxiliary sequences of iterates and/or different update directions than the gradient. This is of course more powerful than just changing the stepsizes, and can be interpreted from a control theory perspective as adding internal dynamics to the algorithm. Accelerated rates were first shown in Nesterov’s seminal work in 1983 [46], and since then, many other accelerated algorithms and analyses have been proposed [4, 9, 13, 16, 24, 29, 61, 64, 65], as well as fruitful interpretations via continuous-time analysis [23, 43, 56, 57, 60, 67, 68]. These accelerated algorithms require only $\Theta(\sqrt{\kappa} \log 1/\varepsilon)$ iterations, or $\Theta(1/\sqrt{\varepsilon})$ in the convex case, which is known to be minimax-optimal up to a constant for any algorithm that uses only gradient information [44]. Much work has recently sharpened this constant, culminating in exactly matching upper and lower bounds [26, 61]. This recent line work exploits the idea that the worst-case convergence of optimization algorithms can be numerically computed via semidefinite programming (SDP) [27, 28, 30, 37, 38, 41, 62, 63, 65]. This has also enabled using computer-automated SDP-analyses to investigate richer classes of algorithms, such as robust versions of accelerated methods [18], proximal algorithms [6], operator splitting [55], line search [20], biased stochastic gradient methods [35], inexact Newton’s method [21], among many others. This area of research is extremely active and we refer the reader to the excellent recent survey [30] for a comprehensive set of references and a detailed historical account.

Accelerated GD via dynamic stepsizes. Although many time-varying stepsize schedules have been considered for GD, no convergences analyses improved over the textbook unaccelerated rate beyond the quadratic case. In 2018, Altschuler’s MS thesis [5] considered time-varying stepsize

schedules in several settings, all through the unifying lens of hedging and multi-step descent. In Chapter 8 of the thesis, the PESTO framework was used to show for the first time the advantage of using time-varying stepsize schedules for GD beyond the quadratic setting. Explicit solutions were given for $n = 2, 3$ in the strongly convex setting. This showed that a constant-factor improvement over the textbook unaccelerated GD rate was indeed possible. A key difficulty in extending this to larger horizons n is that the search for optimal stepsizes is non-convex. In 2022, Das Gupta et al. [19] combined Branch & Bound techniques with the PESTO SDP to develop algorithms that perform this search numerically, and as an example used this to compute good approximate schedules in the convex setting for larger values of n up to 50. Grimmer [32] very recently developed a technique to round these Branch & Bound solutions to exact rational certificates. This allowed him to extend these approximate stepsize schedules up to $n = 127$ in order to get a larger constant-factor improvement, and conjectured that dynamic stepsizes might lead to an accelerated rate of $O(1/(T \log T))$. By extending a recursive application of the 2-step solution in [5], the present paper rigorously proves acceleration for all horizons n , and in particular obtains the first asymptotic improvements over the textbook unaccelerated GD rate—not just by a constant factor.

1.3 Organization

In §2, we provide an overview of the core conceptual ideas via the key case $n = 2$. §3 formally defines the Silver Step Schedule and the Silver Convergence Rate τ_n , §4 establishes the claimed properties of τ_n , and §5 proves that τ_n is a valid bound on the convergence rate of the Silver Step Schedule. §6 discusses future directions. Some technical details are deferred to the Appendix.

2 Conceptual overview: two-step case ($n = 2$)

This section provides a complete analysis for the minimal non-trivial horizon length: $n = 2$. (No hedging can occur if $n = 1$.) Our goal here is to provide further intuition for the core concepts of hedging and multi-step descent, and explain concretely how these manifest in the design and analysis of the Silver Step Schedule. Indeed, the $n = 2$ case captures most of the core intuition and ideas, and the result for general n is essentially just an amped-up version thereof. These results first appeared in Altschuler’s thesis [5, Chapter 8]; we refer to there for a lengthier treatment.

For simplicity, in this section we denote the stepsizes by α and β , so that the algorithm is

$$x_1 = x_0 - \alpha \nabla f(x_0), \quad x_2 = x_1 - \beta \nabla f(x_1),$$

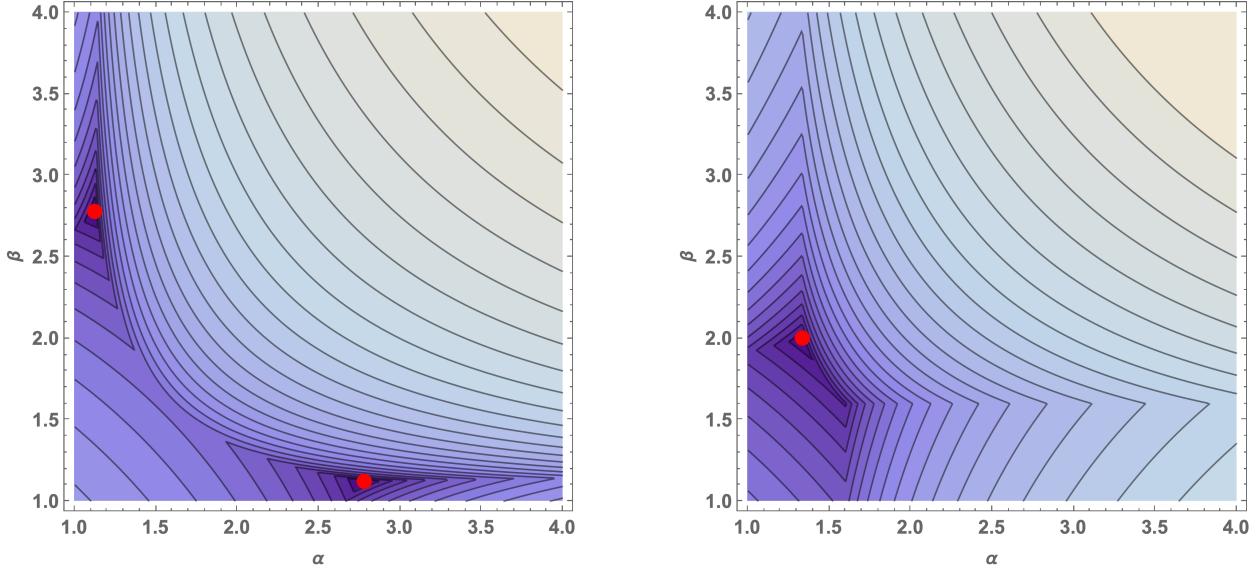
and the worst-case convergence rate over a function class \mathcal{F} is

$$R(\alpha, \beta; \mathcal{F}) := \sup_{f \in \mathcal{F}, x_0 \neq x^*} \frac{\|x_2 - x^*\|}{\|x_0 - x^*\|}.$$

The question of optimal stepsizes is therefore the minimax problem

$$\min_{\alpha, \beta} R(\alpha, \beta; \mathcal{F}). \tag{2.1}$$

To motivate why non-constant stepsizes might be helpful, in §2.1 we first briefly recall the classical result of [69] which solves this for the case of quadratic \mathcal{F} . Then in §2.2, we solve this problem for convex \mathcal{F} by presenting the 2-step Silver Step Schedule from [5, Theorem 8.11], proving its convergence rate via multi-step descent, and proving its optimality via hedging.



(a) Quadratic setting: (α^*, β^*) are the two permutations of $\{1.12339, 2.77905\}$. Details in §2.1.

(b) Convex setting: $(\alpha^*, \beta^*) = (\frac{4}{3}, 2)$. Details in §2.2.

FIGURE 3: Contour plots of worst-case rates, as a function of the two stepsizes α and β , for $m = 1/4$ and $M = 1$. The marked points indicate the global minima. Notice the asymmetry in the convex case (right), due to the non-commutativity of the GD map.

2.1 Optimal stepsizes for quadratic optimization

Young’s argument from 1953. What is the optimal stepsize schedule (α, β) for the class \mathcal{F} of quadratic functions f that are m -strongly convex and M -smooth? Without loss of generality after translating, $f(x) = \frac{1}{2}x^T H x$ where $mI \leq H \leq MI$. By definition of GD, $x_1 = (1 - \alpha H)x_0$ and $x_2 = (1 - \beta H)x_1$, thus

$$x_2 = p(H)x_0, \quad \text{where} \quad p(H) = (1 - \alpha H)(1 - \beta H).$$

Observe that as one ranges over all possible choices of the stepsizes (α, β) , the polynomial p ranges over the set \mathcal{P} of all degree 2 polynomials satisfying the normalizing condition $p(0) = 1$. Therefore finding optimal stepsizes (α, β) is equivalent to finding an optimal polynomial $p \in \mathcal{P}$.

What is the optimal polynomial? By the above display and properties of the spectral norm,

$$R(\alpha, \beta; \mathcal{F}) = \sup_{mI \leq H \leq MI, x_0 \neq 0} \frac{\|p(H)x_0\|}{\|x_0\|} = \sup_{mI \leq H \leq MI} \|p(H)\| = \sup_{m \leq \lambda \leq M} |p(\lambda)|. \quad (2.2)$$

Thus the optimal polynomial $p \in \mathcal{P}_2$ is the one with minimal L_∞ norm over the interval $[m, M]$. It is classically known that this is the (translated and scaled) Chebyshev polynomial of the first kind, see e.g., [53]. Thus the optimal stepsizes (α^*, β^*) are the inverses of the roots $\frac{M+m}{2} \pm \frac{M-m}{2\sqrt{2}}$ of the Chebyshev polynomial, in either order. These are the symmetric marked points in Figure 3, left.

Crucially, observe that these two stepsizes are different—hence the advantage of non-constant schedules in the quadratic setting. We now interpret this phenomenon in two ways that are essential to our intuition for the convex setting. This discussion is based on [5, Chapters 1 and 2].

Interpretation via hedging. Why is $\bar{\alpha} := \frac{2}{M+m}$ suboptimal for 2 steps of GD when it is optimal for 1? Recall that it is optimal for 1 step because GD overshoots when using a longer step $\alpha > \bar{\alpha}$

on the sharp function $f(x) = \frac{M}{2}x^2$, and undershoots when using a shorter step $\alpha < \bar{\alpha}$ on the shallow function $f(x) = \frac{m}{2}x^2$. The algorithmic opportunity is that these worst-case functions are different for short-step GD and long-step GD. This is why using a short step and a long step—each individually suboptimal—can lead to faster overall convergence than using $\bar{\alpha}$ twice. We refer to this misalignment of worst-case functions as *hedging*. See Figure 3, left.

The necessity of multi-step descent. There is a dual interpretation of hedging via multi-step descent. By (2.2), the worst-case rate for 2 steps is

$$R(\alpha, \beta; \mathcal{F}) = \sup_{m \leq \lambda \leq M} |(1 - \alpha\lambda)(1 - \beta\lambda)|. \quad (2.3)$$

Contrast this with the greedy analysis, which bounds the worst-case rate after 2 iterations by the product of the worst-case rates for 1 step with α or β , namely

$$R(\alpha; \mathcal{F}) \cdot R(\beta; \mathcal{F}) = \left(\sup_{m \leq \lambda_\alpha \leq M} |1 - \alpha\lambda_\alpha| \right) \cdot \left(\sup_{m \leq \lambda_\beta \leq M} |1 - \beta\lambda_\beta| \right). \quad (2.4)$$

Observe that the greedy analysis (2.4) is so shortsighted that it not only leads to worse bounds for any given stepsize schedule, but moreover leads to the wrong prescription of stepsizes. Indeed, optimizing this convergence rate (2.4) over (α, β) leads to $\alpha = \beta = \frac{2}{M+m}$ which is the constant schedule. This necessity of multi-step descent explains why the mainstream approach for convex optimization is constant stepsizes: previous approaches were unable to analyze multi-step descent. (This is only tractable in the quadratic setting because the gradient operator is linear, see (2.2).)

2.2 Optimal stepsizes for convex optimization

We now turn to the convex setting. Let \mathcal{F} denote the set of m -strongly convex and M -smooth functions. Young's Chebyshev schedule is then provably bad⁵. What are the optimal 2 stepsizes? Certainly the above discussion of hedging motivates using non-constant stepsizes, but proving this requires multi-step descent, and that has been the longstanding stumbling block preventing progress beyond the quadratic setting.

2.2.1 Silver Stepsize Schedule for $n = 2$

We show below that the 2-step convergence rate $R(\alpha, \beta; \mathcal{F})$ is minimized by the stepsizes (α, β) that are defined by the system of equations

$$(M\alpha - 1)(M\beta - 1) = (1 - \alpha m)(1 - \beta m) = \frac{(1 - m\alpha)(M\beta - 1)}{1 + \alpha(M - m)} \quad (2.5)$$

and moreover the optimal 2-step convergence rate R^* is given by this equalized value.

Remark 2.1. *The equations (2.5) can be solved explicitly, to give the alternative expressions*

$$\alpha^* = \frac{2}{m + S}, \quad \beta^* = \frac{2}{2M + m - S}, \quad R^* = \frac{S - M}{2m + S - M},$$

where $S = \sqrt{M^2 + (M - m)^2}$. These are the formulas given in [5, Thm. 8.10], and is the $n = 2$ case of the Silver Stepsize Schedule and (square-rooted) Silver Convergence Rate defined in §4.

⁵This is not just a failure of analysis techniques: even for mild condition numbers like $\kappa = 10$, using the 2-step Chebyshev Schedule in either order makes GD divergent (i.e., the contraction rate is larger than 1). We are not aware of a reference for this, but it can be shown e.g., by using the SDP-analysis framework of [63].

This $n = 2$ solution showcases the key phenomena that also occur for larger n :

- **Provable advantage of dynamic stepsizes.** Since $R^* < (\frac{M-m}{M+m})^2$, this proves that it is possible to improve over standard GD by dynamically changing the stepsize. (Recall that $\frac{M-m}{M+m}$ is the textbook unaccelerated rate for 1 step of GD.) This mirrors how for quadratics, the optimal 2-step rate (2.3) is better than the squared optimal 1-step rate (2.4).
- **Stepsize splitting.** Since $\alpha^* < \frac{2}{M+m} < \beta^*$, the optimal stepsize $\frac{2}{M+m}$ for $n = 1$ splits into a short step α^* and long step β^* . For general n , the Silver Stepsize Schedule mirrors this splitting at every scale: it splits the largest stepsize into a shorter and longer step.
- **Unique, asymmetric solution.** Unlike the quadratic case, here the stepsize order is essential for fast convergence: the splitting requires the small stepsize to be first.⁶ As a consequence, here the optimal stepsize schedule is unique. See Figure 3, right.
- **Milder splitting.** Even ignoring order, the stepsize values differ from the quadratic case. This occurs because the class of convex functions is richer than the class of quadratics, thus the supremum defining the worst-case rate $R(\alpha, \beta; \mathcal{F})$ is over more functions, thus it is harder to misalign the worst-cases by hedging. The result is less aggressive hedging and partial acceleration: the improvement over the 1-step rate is smaller than in the quadratic case.

We now turn to proving that Theorem 1.1 holds in the case $n = 2$, and moreover that the proposed Silver Stepsize Schedule is optimal among all 2-step schedules.

Theorem 2.2 (Optimal 2-step schedule for strongly convex optimization, Theorem 8.11 of [5]). *Consider any strong-convexity and smoothness parameters $0 < m \leq M < \infty$. The unique optimal 2-step schedule $(\alpha^*, \beta^*) \in \operatorname{argmin}_{\alpha, \beta} R(\alpha, \beta; \mathcal{F})$ and the corresponding optimal 2-step rate R^* are as stated in Remark 2.1.*

The proof has two parts: an upper bound on $R(\alpha^*, \beta^*, \mathcal{F})$ that proves that our 2-step schedule achieves the claimed rate, and a matching lower bound that proves optimality (and in fact uniqueness too). We do this below via multi-step descent and hedging, respectively.

2.2.2 Upper bound: rate certification via multi-step descent

As discussed above, in order to prove any benefit of deviating from the constant stepsizes, we must directly analyze the cumulative multi-step descent of all iterations. This requires capturing how different iterations affect other iterations' progress. We do this by exploiting long-range consistency conditions between the information that GD sees along its trajectory.

Our starting point is a known result on convex interpolability, recalled next. There is a set of consistency conditions that any $f \in \mathcal{F}$ must satisfy at any set of points $\{x_i\}_{i \in \mathcal{I}}$: the co-coercivity

$$Q(x, y) := 2(M - m)(f(x) - f(y)) + 2\langle M\nabla f(y) - m\nabla f(x), y - x \rangle - \|\nabla f(x) - \nabla f(y)\|^2 - Mm\|x - y\|^2$$

must be non-negative for every pair of points $x, y \in \mathcal{I}$. Of particular interest to us is the converse: there are consistency conditions on a set of data $\{(x_i, g_i, f_i)\}_{i \in \mathcal{I}}$ that ensure it is \mathcal{F} -interpolable, i.e., there exists $f \in \mathcal{F}$ satisfying $g_i = \nabla f(x_i)$ and $f_i = f(x_i)$ for each $i \in \mathcal{I}$. Specifically, a celebrated line of work on convex interpolability [54] culminated in a beautiful theorem of [63] which states that $\{(x_i, g_i, f_i)\}_{i \in \mathcal{I}}$ is \mathcal{F} -interpolable if and only if

$$Q_{ij} := 2(M - m)(f_i - f_j) + 2\langle Mg_j - mg_i, x_j - x_i \rangle - \|g_i - g_j\|^2 - Mm\|x - y\|^2 \quad (2.6)$$

⁶We remark that the order may change for different progress measures, see [5, Chapter 8.2].

is non-negative for every pair of indices $i, j \in \mathcal{I}$.

We apply these conditions along the trajectory of GD. Specifically, we take $\mathcal{I} := \{0, 1, \dots, n, *\}$ to index the GD iterates and the optimum, and let $\{(x_i, g_i, f_i)\}_{i \in \mathcal{I}}$ denote the first-order data⁷. The upshot is that this theorem enables replacing the supremum over functions $f \in \mathcal{F}$ by the data $\{(x_i, g_i, f_i)\}_{i \in \mathcal{I}}$ in the definition of the worst-case rate $R(\alpha, \beta; \mathcal{F})$. Note that this replacement is lossless since the interpolability conditions in the theorem are necessary and sufficient.

From the perspective of hedging, these co-coercivity conditions $\{Q_{ij} \geq 0\}_{i \neq j \in \mathcal{I}}$ generate all possible long-range consistency constraints on the objective function given the GD trajectory. From the perspective of multi-step descent, they generate all possible valid inequalities with which one can prove convergence rates for GD. Let us explain how we use this in the case $n = 2$.

Proof of rate upper bound for Theorem 2.2. It suffices to prove the *rate certification identity*

$$R^2 \|x_0 - x^*\|^2 - \|x_2 - x^*\|^2 = \sum_{i \neq j \in \{0, 1, *\}} \lambda_{ij} Q_{ij} \quad (2.7)$$

for some non-negative choice of multipliers λ_{ij} . Indeed, since $Q_{ij} \geq 0$ is non-negative for any objective function $f \in \mathcal{F}$, the rate certification identity implies

$$R^2 \|x_0 - x^*\|^2 - \|x_2 - x^*\|^2 \geq 0, \quad (2.8)$$

which proves the claimed rate. It remains to construct non-negative λ_{ij} for the rate certification identity. This is done in [5, Theorem 8.10]. For completeness, we include the explicit values here, in slightly simpler (but equivalent) form:

$$\lambda = \frac{\alpha^*(\beta^*)^2}{4} \begin{bmatrix} 0 & \frac{(S-m)(S-M)}{(M-m)} & 0 \\ \frac{(S-M)(2M-S-m)}{M-m} & 0 & \frac{2M^2+S^2-2MS-m^2}{M-m} \\ \frac{m^3-m^2S+4M^2S-mS^2-4MS^2+S^3}{M(m+S)} & \frac{2MS-m^2-S^2}{M-m} & 0 \end{bmatrix}. \quad (2.9)$$

Here, the rows and columns are indexed by $0, 1, *$, in that order. \square

Of course, the challenge in such a proof is finding the multipliers λ_{ij} . When we prove our result for general n , we prove that the multipliers for the $2n$ -length Silver Stepsize Schedule are recursively built from repeating the multipliers for the n -length Silver Stepsize Schedule twice, modulo a low rank and sparse correction expressible in closed form. With this recursion, (i) the multipliers for the $n = 2$ case above can be derived formulaically from the textbook proof for $n = 1$, and (ii) the proof for the case of general n mirrors the proof for $n = 2$, at least in spirit.

2.2.3 Lower bound: optimality and uniqueness via hedging

Proof of rate lower bound in Theorem 2.2. We prove the *rate optimality identity*

$$R(\alpha, \beta; \mathcal{F}) \geq \underline{R}(\alpha, \beta), \quad (2.10)$$

for all non-trivial⁸ stepsizes $\alpha, \beta \in [1/M, 1/m]$, where

$$\underline{R}(\alpha, \beta) := \max \left\{ (M\alpha - 1)(M\beta - 1), (1 - \alpha m)(1 - \beta m), (M\alpha - 1)(1 - m\beta), \frac{(1 - m\alpha)(M\beta - 1)}{1 + \alpha(M - m)} \right\}.$$

⁷This is purely an analysis device and does not change the GD algorithm (which neither knows the optimum nor queries function values). Including function values simplifies the interpolability conditions [63] and thus our analysis.

⁸We call such stepsizes non-trivial since if a stepsize is outside this interval, then clipping it to the interval improves convergence. This can be proved by noticing that, of the four hard functions in this proof, all but the fourth apply if $\alpha \geq 1/M$, and all but the third apply if $\alpha \leq 1/m$. By optimizing the resulting analogous bounds (2.10) for the cases $\alpha < 1/M$ and $\alpha > 1/m$, it follows that clipping to the interval $[1/M, 1/m]$ leads to faster convergence.

This suffices since it is straightforward to verify that $\min_{\alpha,\beta} R(\alpha,\beta)$ is minimized uniquely at (α^*, β^*) with value R^* ; this yields the two defining equations in (2.5). Indeed, this verification can be done by hand by case enumeration, or simpler, it can be rigorously proven using standard symbolic computation techniques such as quantifier elimination [15].

It remains to prove (2.10). We do this by exhibiting four “hard-to-optimize” functions $f \in \mathcal{F}$ for which the 2-step convergence rate of GD from initialization $x_0 = 1$ is given by these four values. The first two functions are the quadratics $f(x) = \frac{\lambda}{2}x^2$ for $\lambda \in \{\alpha, \beta\}$, in which case $x_2 = (1 - \lambda\alpha)(1 - \lambda\beta)$. The other two functions are piecewise quadratic. It is perhaps simplest to state these functions via their second derivative since then any function value can be obtained by integrating from the minimum $x^* = 0$. The third function is given by $f''(x) = M$ for $x \geq 0$ and $f''(x) = m$ otherwise, in which case $x_2 = (1 - M\alpha)(1 - m\beta)$. The fourth function is given by $f''(x) = m$ for $x \geq \frac{1-m\alpha}{1+\alpha(M-m)}$ and $f''(x) = M$ otherwise, in which case $x_2 = \frac{(1-m\alpha)(1-M\beta)}{1+\alpha(M-m)}$. This proves the desired identity (2.10). \square

It is insightful to contrast these four hard functions defining the 2-step rate function⁹ with the analog for the quadratic case. Recall from (2.3) that in the quadratic setting, the 2-step rate function $R(\alpha, \beta | \mathcal{F}) = \sup_{m \leq \lambda \leq M} |(1 - \alpha\lambda)(1 - \beta\lambda)|$. Although this seems to require infinitely many λ , it was shown in [5, Chapter 5.2.2] that one can replace the continuum $[m, M]$ with the 3 extrema of Chebyshev polynomials, i.e.,

$$\max_{\lambda \in \{m, M, \frac{M+m}{2}\}} |(1 - \lambda\alpha)(1 - \lambda\beta)|,$$

in the sense that minimizing this over (α, β) yields Young’s 2-step Chebyshev Schedule. These three values of λ correspond to “hard-to-optimize” quadratic functions $f(x) = \frac{\lambda}{2}x^2$. How are they different from the hard functions in the above proof? The first two quadratics are common between the quadratic and convex case, but the remaining functions differ. In particular, the third and fourth functions in the convex case are non-quadratic. In words, the richness of the convex function class enables changing the curvature in different places, which enables more alignment of the bad convergence rates for the individual stepsizes. This makes it provably harder to hedge in the convex setting. Note also that the denominator of the fourth function is singlehandedly responsible for the asymmetry in $\bar{R}(\alpha, \beta)$, and thus in the optimal schedule the convex setting.

This proof can be extended to establish the optimality of the Silver Stepsize Schedule, as will be detailed in a shortly forthcoming paper.

3 Silver Stepsize Schedule

For simplicity of exposition, from here on we restrict to horizons n that are powers of 2 (see §1.1.3 for a discussion of extensions to general n), and we set $m = 1/\kappa$ and $M = 1$ to reduce notational overhead (this is without loss of generality after a possible rescaling).

3.1 Normalized Silver Stepsizes

We construct auxiliary stepsize sequences y_n, z_n , that are normalized in a certain way to lie in the interval $[0, 1]$. The particular normalization (a certain linear fractional transformation defined in §3.2) simplifies the recursive stepsize splitting by making it a quadratic equation.

⁹The rate optimality identity (2.10) actually holds with equality over all non-trivial stepsizes $\alpha, \beta \in [1/M, 1/m]$, although this is unnecessary for our purposes.

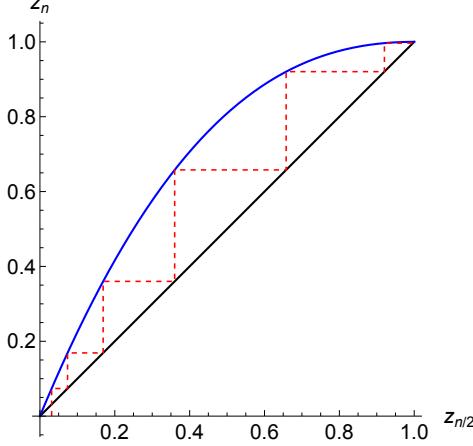


FIGURE 4: Cobweb plot describing the evolution of z_n , under the iteration $z_{n/2} \mapsto z_n$ given in (3.1) and (3.2). The initial condition is $1/\kappa$ (in this plot, $\kappa = 32$). The iterates grow exponentially when z is near zero, and converge quadratically to 1 when z is close to 1.

Explicitly, initialize the sequences $y_1 = z_1 = 1/\kappa$, and define y_n, z_n recursively from $z_{n/2}$ as the solutions to the defining equations

$$y_n z_n = z_{n/2}^2 \quad \text{and} \quad z_n - y_n = 2(z_{n/2} - z_{n/2}^2). \quad (3.1)$$

This is the direct analog of the stepsize splitting detailed for the case $n = 2$ in §2.2. Denoting $\xi = 1 - z_{n/2}$, the explicit solution is

$$y_n = z_{n/2} / (\xi + \sqrt{1 + \xi^2}) \quad \text{and} \quad z_n = z_{n/2} (\xi + \sqrt{1 + \xi^2}). \quad (3.2)$$

The following lemma collect several simple observations about these sequences. See §4 for a detailed discussion of how y_n, z_n both increase to their limits $y_n, z_n \rightarrow 1$, exponentially fast when they are close to 0, and then doubly exponentially fast when they are close to 1.

Lemma 3.1 (Basic properties of the Normalized Silver Stepsizes). *The sequence z_n is monotonically increasing from $z_1 = 1/\kappa$ to $\lim_{n \rightarrow \infty} z_n = 1$. For all n ,*

$$\frac{1}{\kappa} \leq y_n \leq z_n \leq 1. \quad (3.3)$$

Moreover, the above inequality $y_n \leq z_n$ is strict for any $\kappa > 1$.

3.2 Silver Stepsizes

We define the Silver Stepsizes

$$a_n := \psi(y_n) \quad \text{and} \quad b_n := \psi(z_n). \quad (3.4)$$

from the Normalized Silver Stepsizes y_n, z_n via the linear fractional transformation ψ given by

$$\psi : t \mapsto \frac{1 + \kappa t}{1 + t} \quad \text{and} \quad \psi^{-1} : s \mapsto \frac{s - 1}{\kappa - s}. \quad (3.5)$$

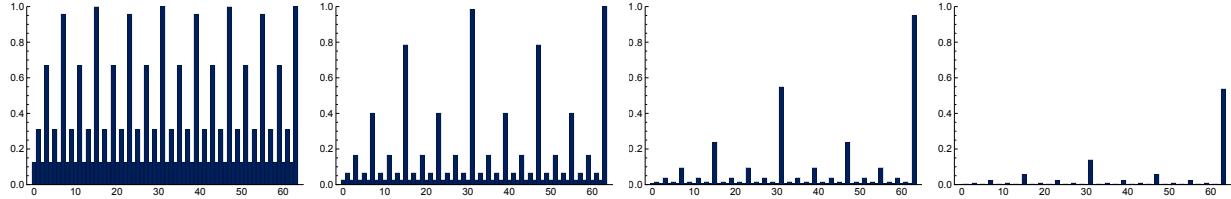


FIGURE 5: Normalized Silver Stepsize Schedule, for different condition numbers $\kappa = 4, 16, 64, 256$. Notice that these are always bounded between 0 and 1. The Silver Stepsize Schedules $h^{(n)}$ shown in Figure 1 are generated by applying ψ to the schedules here.

We remark that this mapping ψ has the following special values

$$\psi(0) = 1, \quad \psi(1/\kappa) = \frac{2}{1+1/\kappa}, \quad \psi(1) = \frac{1+\kappa}{2}, \quad \psi(\infty) = \kappa. \quad (3.6)$$

The significance of the two middle values is that these are the initial stepsizes $a_1 = b_1 = \psi(1/\kappa)$ and the limiting stepsizes $\lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n = \psi(1)$. We remark that these two middle values are the harmonic and arithmetic means of the two extremal values, i.e., $\psi(1/\kappa) = \text{HM}(\psi(0), \psi(\infty))$ and $\psi(1) = \text{AM}(\psi(0), \psi(\infty))$. The looseness in the classical AM-HM inequality therefore quantifies the gap between the initial and limiting stepsizes. The following lemma records this and several other simple observations about these stepsizes.

Lemma 3.2 (Basic properties of the Silver Stepsizes). *The sequence b_n is monotonically increasing from $b_1 = \text{HM}(1, \kappa)$ to $\lim_{n \rightarrow \infty} b_n = \text{AM}(1, \kappa)$. For all n ,*

$$1 \leq \text{HM}(1, \kappa) \leq a_n \leq b_n \leq \text{AM}(1, \kappa) \leq \kappa. \quad (3.7)$$

Moreover, the above inequality $a_n \leq b_n$ is strict for any $\kappa > 1$.

3.3 Silver Stepsize Schedule

Let $h^{(n)}$ denote the Silver Stepsize Schedule of length n . Denote its $n/2$ -th stepsize by a_n and its n -th by b_n . As overviewed briefly in §1.1.3, we recursively construct

$$h^{(n)} := [\tilde{h}^{(n/2)}, a_n, \tilde{h}^{(n/2)}, b_n], \quad (3.8)$$

where $\tilde{h}^{(n/2)}$ denotes everything in $h^{(n/2)}$ except the final step, i.e., everything except $b_{n/2}$. Note that $b_{n/2}$ is in $h^{(n/2)}$, but not in $h^{(n)}$; it is split into a_n and b_n . Note also that a_n, b_n form the largest stepsizes in $h^{(n)}$, with b_n being the largest (Lemma 3.2). For the convenience of the reader, we recall from §1.1.3 that for small n , this pattern is

$$\begin{aligned} h^{(1)} &= [a_1] \\ h^{(2)} &= [a_2, b_2] \\ h^{(4)} &= [a_2, a_4, a_2, b_4] \\ h^{(8)} &= [a_2, a_4, a_2, a_8, a_2, a_4, a_2, b_8] \end{aligned}$$

See Figure 1 for an illustration of this pattern, and see §1.1.3 for a discussion of the emergent fractal, dependence on the horizon, and patterns for small n .

Remark 3.3 (Occupation measure). *For all $i \in \mathbb{N}$ and all sufficiently large horizons $n \geq 2^i$, the stepsize a_{2^i} is used in 2^{-i} fraction of the n -step Silver Stepsize Schedule. For example, for all horizons $n \geq 2$, the smallest stepsize $a_2 = \kappa/(\kappa - 1)$ is used in every other iteration. For the infinite limit of the Silver Stepsize Schedule (see §1.1.3), the occupation measure simplifies to*

$$\sum_{i=1}^{\infty} 2^{-i} \delta_{a_{2^i}}.$$

This can be viewed as a geometric distribution that takes value a_{2^i} with probability 2^{-i} .

3.4 Silver Convergence Rate

We define the Silver Convergence Rate as

$$\tau_n := \left(\frac{1 - z_n}{1 + z_n} \right)^2. \quad (3.9)$$

Of course, from just this definition it is not yet clear why we call τ_n a rate; in §5 we prove that τ_n is the convergence rate of the Silver Stepsize Schedule. Note that since z_n is monotonically increasing (Lemma 3.1), this rate τ_n is monotonically decreasing from the textbook unaccelerated rate $\tau_1 = ((\kappa - 1)/(\kappa + 1))^2$ to $\lim_{n \rightarrow \infty} \tau_n = 0$. In the following section, we provide a complete understanding of exactly how fast τ_n converges to 0.

4 Analysis of the Silver Convergence Rate

Here we prove the bound on the Silver Convergence Rate τ_n in our main result (Theorem 1.1). We restate this bound for convenience.

Theorem 4.1 (Silver Convergence Rate). *Denote $i^* := \lfloor \log_{\rho} \frac{\kappa}{3} \rfloor$ and $n^* := 2^{i^*}$. Then for any n that is a power of 2, we have the following bound on τ_n .*

- Acceleration regime. If $n \leq n^*$, then

$$\tau_n = \exp \left(-\Theta \left(\frac{n^{\log_2 \rho}}{\kappa} \right) \right).$$

- Saturation regime. If $n > n^*$, then

$$\tau_n = \exp \left(-\Theta \left(\frac{n}{n^*} \right) \right).$$

This result establishes $n^* \asymp \kappa^{\log_2 2}$ as the location of a phase transition. There, the Silver Convergence Rate τ_n switches from super-exponential to exponential in the horizon n . See the introduction for a detailed discussion of this phase transition and the intuition behind it in terms of how the Silver Stepsize Schedule is effectively periodic with period of length n^* .

For simplicity, we make no attempt to optimize the constants in the Θ and the choice of i^* (the $1/3$ in the theorem statement is arbitrary). Our proofs make crude constant bounds to ease the exposition, and it is straightforward to tighten these. However, as established by our upper and lower bounds, our proofs are already tight up to reasonable constant factors.

The section is organized as follows. In §4.1, we provide a heuristic derivation of Theorem 4.1 that explains the phase transition via Taylor expanding the dynamics in the two regimes. This gives the central intuition for the result and its proof. In §4.2, we make these Taylor expansions precise to conclude the proof of Theorem 4.1.

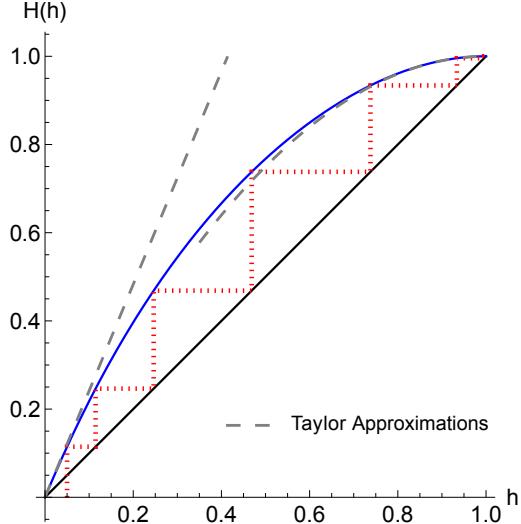


FIGURE 6: Cobweb diagram for the function $H(h)$ in equation (4.3) and its Taylor approximations (4.4) and (4.6). Starting from the initial condition (4.2), iterates grow by a constant multiplicative factor of the Silver Ratio $\rho = 1 + \sqrt{2}$ when h is near zero, and converge quadratically to 1 when h is close to 1.

4.1 Heuristic derivation

The phase transition in $\tau_n = \left(\frac{1-z_n}{1+z_n}\right)^2$ is a consequence of the phase transition in the dynamics of the auxiliary sequence z_n . To explain this, it is convenient to simplify notation by re-indexing $n = 2^i$ so that iterations of the dynamical process are indexed by $i = 0, 1, 2, 3, \dots$ rather than $n = 1, 2, 4, 8, \dots$. It is helpful to also re-parameterize

$$h_i := \Psi(z_{2^i}),$$

where $\Psi : (0, 1) \rightarrow (0, 1)$ is the monotone bijection

$$\Psi(z) := \frac{2z}{1+z}.$$

The significance of this re-parameterization to h_i is that

$$\tau_n = \left(\frac{1-z_n}{1+z_n}\right)^2 = (1-h_i)^2. \quad (4.1)$$

Thus, proving a fast convergence rate amounts to lower bounding h_i .

What do the dynamics of h_i look like? At initialization, $z_1 = 1/\kappa$ (see §3), thus

$$h_0 = \frac{2}{1+\kappa} \asymp \frac{1}{\kappa}. \quad (4.2)$$

Then the iterations of this process increase h_i exponentially fast to 1 when it is sub-constant size, and then doubly-exponentially fast when h_i is of constant size. (This dichotomy is the source of the phase transition in Theorem 4.1.)

To analyze these dynamics, let $H : (0, 1) \rightarrow (0, 1)$ denote the update function sending h_i to h_{i+1} . Then $H = \Psi \circ F \circ \Psi^{-1}$ where $F(z) = z(1 - z + \sqrt{1 + (1 - z)^2})$ is the function that updates z_n to $z_{2n} = F(z_n)$, see Section 3. A direct algebraic computation gives the explicit expression

$$H(h) = \frac{h(2 - 3h + \sqrt{5h^2 - 12h + 8})}{2(1 - h^2)}. \quad (4.3)$$

Taylor expanding H around $h \approx 0$ and $h \approx 1$ illustrates the markedly different dynamics in these two regimes; see Figure 6.

- Acceleration regime. For $h \ll 1$,

$$H(h) \approx \rho h. \quad (4.4)$$

Thus, in this regime, each h_i increases by a factor of roughly ρ , thus $h_i \approx \rho^i h_0 \approx \rho^i / (2\kappa)$, thus the Silver Convergence Rate is roughly

$$\tau_n = (1 - h_i)^2 \approx \exp(-2h_i) \approx \exp(-\rho^i / \kappa) = \exp(-n^{\log_2 \rho} / \kappa). \quad (4.5)$$

This regime lasts for only $i \approx \log_\rho \kappa$ iterations (aka horizon $n = 2^i \approx \kappa^{\log_2 \rho}$) because at that point $h_i \asymp \rho^i / \kappa \asymp 1$ is of constant size. This is the phase transition.

- Saturation regime. For $h \approx 1$,

$$1 - H(h) \approx (1 - h)^2. \quad (4.6)$$

In words, the key phenomenon here is that the average rate $\tau_n^{1/n}$ stays essentially the same as n increases—in contrast to the acceleration regime, in which the average rate improves in n . Indeed, the Taylor expansion (4.6) indicates that in the saturation regime, $\tau_n = (1 - h_i)^2 \approx (1 - h_{i-1})^4 = \tau_{n/2}^2$. By repeating this argument and then using the fact that $\tau_{n^*} = \exp(-\Theta(1))$ which follows from the acceleration regime, we obtain

$$\tau_n \approx (\tau_{n^*})^{n/n^*} = \exp(-\Theta(n/n^*)). \quad (4.7)$$

If the approximations were justified in the above two displays, then this informal argument would lead to a proof of Theorem 4.1. We do this in the following subsection.

4.2 Rigorous derivation

Here we prove Theorem 4.1. We first state two helper lemmas, which formalize the Taylor approximations (4.4) and (4.6) in the acceleration regime and saturation regime, respectively.

Lemma 4.2 (Dynamics in the acceleration regime). *Let $\nu := \frac{3\rho}{2\sqrt{2}} \approx 2.561$. For all $h \geq 0$,*

$$\rho h - \nu h^2 \leq H(h) \leq \rho h.$$

Lemma 4.3 (Dynamics in the saturation regime). *For all $h \geq 0$,*

$$(1 - h)^2 - (1 - h)^4 \leq 1 - H(h) \leq (1 - h)^2.$$

We omit the proofs of these lemmas, since the inequalities are visually obvious from plotting the functions, and can be formally proven in a routine algorithmic way, as they only involve algebraic functions of a single scalar variable. This is done by computing the critical points and using well-known techniques for root isolation; see e.g. [8].

An appealing consequence of Lemma 4.3 is the inequality $\tau_{2n} \leq \tau_n^2$. We call this the *rate monotonicity* property of the Silver Stepsize Schedule, since it amounts to the statement that using the $2n$ -step schedule is at least as good as using the n -step schedule twice.

Corollary 4.4 (Rate monotonicity property for the Silver Stepsize Schedule). *For any n that is a power of 2,*

$$\tau_{2n} \leq \tau_n^2.$$

Proof. By (4.1), then Lemma 4.3, then (4.1) again, we have $\tau_{2n} = (1 - h_{i+1})^2 \leq (1 - h_i)^4 = \tau_n$. \square

Proof of Theorem 4.1. Here we prove the upper bounds (a.k.a., the convergence rates). The matching lower bounds are conceptually identical and deferred to Appendix A for brevity.

Acceleration regime. Suppose $n \leq n^*$. Let $i := \log_2 n \leq i^*$. We bound

$$h_i \geq \rho^i h_0 - \nu \sum_{t=0}^{i-1} \rho^{i-1-t} h_t^2 \geq \rho^i h_0 - \nu h_0^2 \rho^{i-1} \sum_{t=0}^{i-1} \rho^t \geq \rho^i h_0 \left(1 - \frac{3}{4} \rho^i h_0\right) \geq \frac{\rho^i h_0}{2} = \frac{n^{\log_2 \rho}}{2\kappa}. \quad (4.8)$$

Above, the first step is by i applications of the lower bound in Lemma 4.2. The second step is because $h_t \leq \rho^t h_0$ by t applications of the upper bound in Lemma 4.2. The third step is by summing the geometric series, crudely dropping a positive term, and simplifying $\nu/(\rho\sqrt{2}) = 3/4$. The fourth step is because $\rho^i h_0 \leq \rho^{i^*} h_0 \leq 2/3$ by definition of i^* and the initialization upper bound $h_0 \leq 2/\kappa$, see (4.2). The final step is by definition of $i = \log_2 n$ and the initialization lower bound $h_0 \geq 1/\kappa$, see (4.2). This completes the proof since by (4.1),

$$\tau_n = (1 - h_i)^2 \leq \exp(-2h_i) \leq \exp(-n^{\log_2 \rho}/\kappa).$$

Saturation regime. Next, suppose $n > n^*$. By n/n^* applications of Corollary 4.4 and then using the bound on τ_{n^*} proved in the acceleration regime, we have

$$\tau_n \leq (\tau_{n^*})^{n/n^*} \leq \exp\left(-\frac{n}{n^*} \cdot \frac{(n^*)^{\log_2 \rho}}{\kappa}\right).$$

The proof is complete by using the definition of n^* and i^* to bound $(n^*)^{\log_2 \rho} = \rho^{i^*} \geq \frac{\kappa}{3\rho}$. \square

5 Certificate of the Silver Convergence Rate

Here we prove that the Silver Stepsize Schedule has convergence rate τ_n . This is where we establish multi-step descent. For a conceptual overview, we refer the reader to §2.2 for the case of $n = 2$; the proof for general n here mirrors that key case, albeit is more technically involved.

Recall from the discussion there that the proof strategy amounts to finding a *certificate* $\{\lambda_{ij}\}$ for the rate τ_n , by which we mean non-negative multipliers $\{\lambda_{ij}\}_{i,j \in \{0, \dots, n-1, *\}}$ such that

$$\tau_n \|x_0 - x^*\|^2 - \|x_n - x^*\|^2 = \sum_{i,j \in \{0,1,\dots,n,n,*\}} \lambda_{ij} Q_{ij}. \quad (5.1)$$

See §2.2 for a definition of the co-coercivities Q_{ij} . Briefly, these are valid inequalities that generate all possible long-range consistency conditions between the gradients seen along GD's trajectory.

Our proof builds the $2n$ -step certificate by *recursively gluing* two copies of the n -step certificate and adding slight modifications to account for the fact that the $2n$ -step Silver Stepsize Schedule $h^{(2n)}$ differs from $[h^{(n)}, h^{(n)}]$ in two out of the $2n$ stepsizes. Concretely, this recursive gluing can be understood as creating the $(2n+1) \times (2n+1)$ matrix $\{\lambda_{ij}\}_{i,j \in \{0, \dots, 2n-1, *\}}$ from the $(n+1) \times (n+1)$ matrix $\{\sigma_{ij}\}_{i,j \in \{0, \dots, n-1, *\}}$ in three parts: a *tensor product* which glues together two copies of the n -step certificate, a *rank-one correction* which affects the rows indexed by $i \in \{n-1, 2n-1, *\}$, and a *sparse correction* which affects the 6 entries (i, j) where $i \neq j \in \{n-1, 2n-1, *\}$. See Figure 7.

x_0	x_1	\cdots	x_{n-1}	x_n	\cdots	\cdots	x_{2n-1}	x^*
x_0								
x_1								
\vdots								
x_{n-1}				Ξ	Ξ	Ξ	Δ	Δ
x_n								
\vdots								
x_{2n-1}			Δ	Ξ	Ξ	Ξ		Δ
x^*		Δ	Ξ	Ξ	Ξ	Ξ	Δ	

FIGURE 7: Components of the recursively glued certificate in Theorem 5.2, illustrated here for combining two copies of the $n = 4$ certificate (shaded) to create the $2n = 8$ certificate.

This recursive gluing is formally stated in Theorem 5.2 below. To most easily state this result, we first isolate a certain property of the sparsity pattern of the multipliers λ_{ij} that holds by construction in our recursion. This property is technical and eases the proof.

Definition 5.1 ($*$ -sparsity property). *A collection of weights $\{\lambda_{ij}\}_{i,j \in \{0, \dots, n-1, *\}}$ satisfies the $*$ -sparsity property if $\lambda_{i,*}$ for all $i < n - 1$.*

Theorem 5.2 (Recursive gluing for the Silver Stepsize Schedule). *Let $\kappa \in (1, 2) \cup (2, \infty)$. Suppose $\{\sigma_{ij}\}_{i,j \in \{0, \dots, n-1, *\}}$ satisfies $*$ -sparsity and certifies the n -step rate, i.e.,*

$$\tau_n \|x_0 - x^*\|^2 - \|x_n - x^*\|^2 = \sum_{i,j \in \{0, \dots, n-1, *\}} \sigma_{ij} Q_{ij} \quad \text{for stepsize schedule } h^{(n)}. \quad (5.2)$$

*Then there exists $\{\lambda_{ij}\}_{i,j \in \{0, \dots, 2n-1, *\}}$ that satisfies $*$ -sparsity and certifies the $2n$ -step rate, i.e.,*

$$\tau_{2n} \|x_0 - x^*\|^2 - \|x_{2n} - x^*\|^2 = \sum_{i,j \in \{0, \dots, 2n-1, *\}} \lambda_{ij} Q_{ij} \quad \text{for stepsize schedule } h^{(2n)}. \quad (5.3)$$

Moreover, this certificate is explicitly given by

$$\lambda_{ij} := \underbrace{\Theta_{ij}}_{\text{gluing}} + \underbrace{\Xi_{ij}}_{\text{rank-one correction}} + \underbrace{\Delta_{ij}}_{\text{sparse correction}} \quad (5.4)$$

where the “gluing component” Θ is defined as

$$\Theta_{i,j} := \underbrace{\frac{\tau_{2n}}{\tau_n} \sigma_{i,j} \cdot \mathbb{1}_{i,j \in \{0, \dots, n-1, *\}}}_{\text{recurrence for first } n \text{ steps}} + \underbrace{c \sigma_{i-n,j-n} \cdot \mathbb{1}_{i,j \in \{n, \dots, 2n-1, *\}}}_{\text{recurrence for second } n \text{ steps}}, \quad (5.5)$$

*the “rank-one correction” Ξ is zero except $\{\Xi_{ij}\}_{i \in \{n-1, 2n-1, *\}, j \in \{n, \dots, 2n-2\}}$, and the “sparse correction” Δ is zero except $\{\Delta_{ij}\}_{i \neq j \in \{n-1, 2n-1, *\}}$. The explicit values of c , Ξ , Δ are provided in Appendix B.*

While the explicit values of Ξ and Δ are somewhat involved, the key point is that they can be expressed as rational functions in just z_n, y_{2n}, z_{2n} , see Remark B.1. Importantly, since y_{2n}, z_{2n} are explicit algebraic functions of z_n by construction (see §3.1), this turns verifying the claimed identity (5.3) into a straightforward (albeit tedious) algebraic exercise that is rigorously automatable via standard computer algebra techniques [17].

A few minor remarks. First, for simplicity, Theorem 5.2 assumes $\kappa \neq 2$. This allows us to multiply and divide by $\kappa - 2$, which simplifies expressions. The rate for $\kappa = 2$ anyways follows immediately from $\kappa = 2 + \varepsilon$ for $\varepsilon \downarrow 0$. Second, in (5.4) the notational shorthand $i - n$ is understood to be $*$ when $i = *$. Third, when it is said that λ satisfies $*$ -sparsity, it is understood that this property corresponds to the horizon of length $2n$, i.e., $\lambda_{*,i} = 0$ for all $i < 2n - 1$.

Theorem 5.2 immediately implies the convergence rate (1.4) in our main result Theorem 1.1.

Proof of convergence rate in Theorem 1.1. The base case of $n = 1$ is the classical analysis of GD; see, e.g., [5, Chapter 8] for a proof in this language of co-coercivities.¹⁰ The convergence rate is the textbook unaccelerated rate $\tau_1 = (\frac{\kappa-1}{\kappa+1})^2$. By induction, Theorem 2.2 implies that τ_n is a valid convergence rate for the n -step Silver Stepsize Schedule, for all n that are powers of 2. \square

Below, in §5.1, we express the components of the recursively glued certificate as succinct quadratic forms, and then in §5.2, we use this to prove Theorem 5.2.

5.1 Recursive gluing

Proving Theorem 5.2 requires establishing that λ satisfies the identity (5.3). Ignoring presently the linear form in the function values (that term is much simpler and addressed in §5.2), this amounts to showing equality of two quadratic forms. Naïvely, this requires checking equality of *all* coefficients of these quadratic forms—which is painstaking since these are quadratics in all the GD iterates $x_0, \dots, x_{2n-1}, x^*$ and their corresponding gradients $g_0, \dots, g_{2n-1}, g^*$, and moreover are defined over the ideal generated by the GD equations $x_{t+1} = x_t - \alpha_t g_t$. A key observation that removes much of this labor is that *the quadratic forms in our recursive certificate have rank at most 4*. In fact, these quadratic forms are only in the four variables $x_{n-1}, g_{n-1}, x_{2n-1}, g_{2n-1}$. This reduces the number of coefficients to be checked from $\Theta(n^2)$ to a constant number: 10.

This observation is formalized in the following lemma, which expresses the quadratic forms via coefficient matrices as this is convenient for book-keeping. For brevity, just as in Theorem 5.2, the explicit values of these matrices are deferred to the Appendix, but the key point is that each entry can be expressed a rational function of just z_n, y_{2n}, z_{2n} , see Remark B.1. To isolate the quadratic form component of the co-coercivities, let P_{ij} denote Q_{ij} without its linear component $f_i - f_j$, i.e.,

$$P_{ij} := 2\langle g_j - \frac{g_i}{\kappa}, g_j - g_i \rangle - \|g_i - g_j\|^2 - \frac{1}{2(\kappa-1)}\|x_i - x_j\|^2.$$

Lemma 5.3 (Recursive gluing via succinct quadratic forms). *Consider the setup of Theorem 5.2, let $v := [x_{n-1}, g_{n-1}, x_{2n-1}, g_{2n-1}]^T$, and let E, S, L be the 4×4 matrices defined in Appendix B.4.*

- Gluing error: $\tau_{2n}\|x_0\|^2 - \|x_{2n}\|^2 - \sum_{i,j \in \{0, \dots, 2n-1, *\}} \Theta_{ij} P_{ij} = \langle E, vv^T \rangle$
- Sparse correction: $\sum_{ij} \Delta_{ij} P_{ij} = \langle S, vv^T \rangle$
- Rank-one correction: $\sum_{ij} \Xi_{ij} P_{ij} = \langle L, vv^T \rangle$

¹⁰One can also use Theorem 2.2 to take $n = 2$ as the base case. Then this paper's proof is fully self-contained.

For brevity, we defer the proof of the sparse and low-rank corrections to Appendix. However, we provide the proof of the gluing error here to provide intuition for why these quadratic forms have constant rank rather than the a priori upper bound of $\Theta(n)$. In particular, the proof shows how the low rank arises from the recursive construction of the Silver Stepsize Schedule that creates $h^{(2n)}$ from $h^{(n)}$, modulo only changing the n -th and $2n$ -th stepsizes (each increases the rank by 2).

Proof of gluing error for Lemma 5.3. Denote by $\tilde{x}_n := x_{n-1} - b_n g_{n-1}$ and $\tilde{x}_{2n} := x_{2n-1} - b_n g_{2n-1}$ the iterates obtained by running GD with the Silver Stepsize Schedule $h^{(n)}$ from initializations x_0 and x_n , respectively. By definition of σ as a certificate for the n -step rate,

$$\sum_{i,j \in \{0, \dots, n-1, *\}} \sigma_{ij} P_{ij} = \tau_n \|x_0\|^2 - \|\tilde{x}_n\|^2 \quad \text{and} \quad \sum_{i,j \in \{n, \dots, 2n-1, *\}} \sigma_{ij} P_{ij} = \tau_n \|x_n\|^2 - \|\tilde{x}_{2n}\|^2$$

Thus the desired quantity is equal to

$$\begin{aligned} \left(\tau_{2n} \|x_0\|^2 - \|x_{2n}\|^2 \right) - \sum_{i,j \in \{0, \dots, 2n-1, *\}} \Theta_{ij} P_{ij} &= \left(\tau_{2n} \|x_0\|^2 - \|x_{2n}\|^2 \right) - \frac{\tau_{2n}}{\tau_n} \left(\tau_n \|x_0\|^2 - \|\tilde{x}_n\|^2 \right) - c \left(\tau_n \|x_n\|^2 - \|\tilde{x}_{2n}\|^2 \right) \\ &= \left(\frac{\tau_{2n}}{\tau_n} \|\tilde{x}_n\|^2 - c \tau_n \|x_n\|^2 \right) + \left(c \|\tilde{x}_{2n}\|^2 - \|x_{2n}\|^2 \right). \end{aligned}$$

Now by definition of GD, $x_n = x_{n-1} - a_{2n} g_{n-1}$, $x_{2n} = x_{2n-1} - b_{2n} g_{2n-1}$, $\tilde{x}_n = x_{n-1} - b_n g_{n-1}$, and $\tilde{x}_{2n} = x_{2n-1} - b_n g_{2n-1}$. By plugging this into the above display and expanding the square, we see that the discrepancy between the $\|\tilde{x}_n\|^2$ and $\|x_n\|^2$ terms creates a quadratic form in just x_{n-1}, g_{n-1} , and similarly the discrepancy between the $\|\tilde{x}_{2n}\|^2$ and $\|x_{2n}\|^2$ terms creates a quadratic form in just x_{2n-1}, g_{2n-1} . Tracking coefficients completes the proof. \square

5.2 Certificate verification

Proof of Theorem 5.2. The non-negativity and $*$ -sparsity properties of λ are direct from the explicit values of λ ; details in Appendix B.3. It therefore suffices to check the rate certificate (5.3). By definition of the co-coercivity Q_{ij} , this certificate has two components: a linear form in $\{f_i\}_{i \in \{0, \dots, 2n-1, *\}}$ and a quadratic form in $\{x_i, g_i\}_{i \in \{0, \dots, 2n-1, *\}}$. We check these two components below.

Quadratic form in iterates and gradients. By Lemma 5.3, it suffices to show that

$$E - S - L = 0, \tag{5.6}$$

where E, S, L are the matrices defined in Appendix B.4. This amounts to checking the 10 entries on or above the diagonal of these 4×4 matrices—elements below the diagonal need not be checked as the matrices are symmetric. By Lemma B.3, these entries can be expressed as rational functions in z_n, y_{2n}, z_{2n} , which are polynomially related via (3.1). Therefore, checking that these 10 entries vanish amounts to checking that certain polynomials vanish modulo an associated ideal. This verification is rigorously automatable using standard techniques from computational algebraic geometry such as Gröbner bases; see e.g. [17, 39]. A simple script for Mathematica (or other computer algebra systems) that verifies these identities is available at the URL given in the references [1]. We emphasize that this is purely in the interest of brevity: verifying these identities can be done by hand, as it just amounts to straightforward (albeit tedious) algebraic cancellations.

Linear form in function values. Recall that each Q_{ij} contributes $2(M-m)(f_i - f_j)$. Thus, in order to show that all function values vanish in $\sum_{ij} \lambda_{ij} Q_{ij}$, it is equivalent to show that

$$\sum_j \lambda_{ij} = \sum_j \lambda_{ji}, \quad \forall j \in \{0, \dots, 2n-1, *\}. \quad (5.7)$$

That is, the j -th row and column sums of λ must match, for all j . We call refer to these identities as *netflow constraints*. Since σ is a valid certificate, it satisfies the netflow constraints $\sum_j \sigma_{ij} = \sum_j \sigma_{ji}$ for all $j \in \{0, \dots, n-1, *\}$. Thus, by construction of Θ from σ , it follows that Θ satisfies the netflow constraints $\sum_j \Theta_{ij} = \sum_j \Theta_{ji}$ for all $i \in \{0, \dots, 2n-1, *\}$. Therefore, in order to prove (5.7), it is equivalent to prove the netflow constraints for $\Xi + \Delta$; that is,

$$\sum_j (\Xi_{ij} + \Delta_{ij}) = \sum_j (\Xi_{ji} + \Delta_{ji}), \quad \forall i \in \{0, \dots, 2n-1, *\}. \quad (5.8)$$

The cases $i \in \{0, \dots, n-2\}$ are trivial since on these rows and columns, Ξ and Δ are identically zero. The cases $i \in \{n, \dots, 2n-2\}$ are similarly trivial because on these rows and columns, Δ is identically zero and $\sum_j (\Xi_{ji} - \Xi_{ij}) = \Xi_{n-1,i} + \Xi_{2n-1,i} + \Xi_{*,i} = 0$ by construction of Ξ . It remains only to prove (5.8) for $i \in \{n-1, 2n-1, *\}$. By the sparsity patterns of Ξ and Δ , this amounts to showing

$$\sum_{j \in \{n-1, 2n-1, *\} \setminus \{i\}} (\Delta_{ij} - \Delta_{ji}) + \sum_{j=n}^{2n-2} \Xi_{ij} = 0, \quad \forall i \in \{n-1, 2n-1, *\}. \quad (5.9)$$

By Lemma B.3, these quantities can be expressed as rational functions in z_n, y_{2n}, z_{2n} , which are polynomially related via (3.1). Therefore, checking that the three quantities vanish in (5.9) amounts to checking that three polynomials vanish modulo an ideal. As mentioned above, this verification is rigorously automatable using standard computational algebra techniques; see the same URL [1] for a simple script implementing this computation. □

6 Future work

This work removes a key stumbling block in previous analyses of optimization algorithms: we show that directly analyzing *multi-step descent* can lead to improved convergence analyses. This general principle opens up a number of directions in both the design and analysis of optimization algorithms. We list a few here.

Beyond GD. Do these techniques extend to stochastic settings where gradients are noisy or only computed approximately? This is motivated by modern machine learning settings such as empirical risk minimization. What about constrained settings where projections are interleaved? Or other settings where one uses coordinate descent, proximal steps, etc.? What about second-order methods such as Newton or Interior Point methods? The modern optimization toolbox is broad, and the algorithmic opportunity of faster multi-step descent that we establish warrants re-investigating many existing algorithms that use greedy analyses.

Beyond convexity. While our techniques extend to the convex setting (see §1.1.4), it is less clear if extensions to non-convex settings are also possible. In particular, can one prove accelerated rates for converging to an stationary point? Could this justify empirical phenomena observed in neural network training such as super-acceleration from cyclic stepsize schedules [58, 59]?

Faster convergence for restricted function classes. Is faster convergence possible if the objective function is more structured? One well-motivated direction here is low-dimensional objective functions. It is known that faster asymptotic convergence is possible if the dimension d is fixed and the number of iterations $n \rightarrow \infty$, e.g., via cutting planes. Recent work has shown that certain momentum-based modifications to GD can also surpass standard lower bounds [44] for sufficiently large n [49]. Do such phenomena extend to GD with dynamic stepsizes? Altschuler’s thesis [5, Chapter 6] proved that for univariate convex functions (or more generally, separable convex functions), GD achieves the fully accelerated rate $\Theta(\sqrt{\kappa} \log 1/\varepsilon)$ via a certain (random) dynamic choices of stepsizes. Does this extend to higher dimension? What is the fundamental trade-off between n , d , and the convergence rate?

Robustness. The Silver Stepsize Schedule periodically uses extremely large step sizes, which are overly aggressive in isolation, but effective when combined with other short steps. It is natural to wonder if this dependence between iterations makes such strategies more sensitive to model misspecification, noisy gradients, inexact arithmetic, or other considerations in practical implementations. We expect this may occur, since it does for other accelerated algorithms, see e.g., [22].

Acknowledgements. JMA is grateful to his friends for their patience over the past seven years as he continually complained about how hard this problem was.

A Deferred details for §4

Here we prove the matching lower bounds in Theorem 4.1.

Acceleration regime. Suppose $n \leq n^*$. Then

$$\tau_n = (1 - h_i)^2 \geq \exp(-4h_i) \geq \exp(-8\rho^i/\kappa) = \exp(-8n^{\log_2 \rho}/\kappa).$$

Above, the first step is by (4.1). The third step is by the upper bound in Lemma 4.2 and the initialization upper bound $h_0 \leq 2/\kappa$ from (4.2). The fourth step is by definition of $i = \log_2 n$. It remains to argue the second step. This is due to the elementary inequality $1 - h \geq \exp(-2h)$ which holds for $h \in (0, 2/3)$ and is applicable since

$$h_i \leq \rho^i h_0 \leq \rho^{i^*} h_0 \leq 2/3.$$

Here, we used same upper bound in Lemma 4.2, the same initialization upper bound $h_0 \leq 2/\kappa$, and, critically, the fact that $i \leq i^*$ since we are in the acceleration regime.

Saturation regime. Suppose $n > n^*$. Then

$$\tau_{2n} = (1 - h_{i+1})^2 \geq ((1 - h_i)^2 - (1 - h_i)^4)^2 = \tau_n^2 (1 - \tau_n)^2 \geq \frac{\tau_n^2}{16},$$

where the first and third steps are by (4.1), the second step is by the lower bound in Lemma 4.3, and the final step is because $\tau_n \leq \tau_{n^*} \leq \exp(-1/3)$ by the rate monotonicity in Corollary 4.4 and the upper bound we proved for τ_{n^*} in the acceleration regime. By unrolling this recursion from n to n^* , continuing to crudely bound constants for simplicity of exposition, and then plugging in the lower bound $\tau_{n^*} \geq \exp(-8)$ from the acceleration regime, we conclude

$$\tau_n \geq \left(\frac{\tau_{n^*}}{16}\right)^{n/n^*} \geq \exp\left(-\frac{n}{2n^*}\right).$$

B Deferred details for §5

Here we provide the deferred details for the proof of Theorem 5.2. See §5 for a proof overview. Three remarks on notation in this Appendix. First, after a possible translation of both f and x_0 , we assume without loss of generality that $x^* = 0$. Second, it is convenient to define the shorthand

$$q_i := \frac{\alpha_i(1 - \frac{\alpha_{i+1}}{\kappa})}{\alpha_{i+1}},$$

where $\alpha_0, \dots, \alpha_{2n-1}$ index the $2n$ -length Silver Stepsize Schedule $h^{(2n)}$. Third, as is standard convention, products over the empty set such as $\prod_{t=n}^{n-1} q_t$ have value 1.

We begin by explicitly stating the correction components used in the recursive gluing. It is convenient to provide two equivalent versions of these expressions. In the first version (Definition B.2), the low-rank correction Ξ is explicitly defined for every entry, and the sparse correction Δ is typically defined as something minus the gluing component Θ . Such expressions for Δ are convenient for computing the final certificate λ because Θ cancels. In the second version (Lemma B.3), Ξ is given only through its row sums (which is the only way Ξ is needed for the proof of Theorem 5.2), and Δ is given via explicit expressions for the subtracted entries of Θ . The key benefit of this second version is that it provides explicit expressions in terms of z_n, y_{2n}, z_{2n} for all quantities required in the proof of Theorem 5.2. For easy recall, we isolate this important fact in the following remark.

Remark B.1 (Explicit rational functions of z_n, y_{2n}, z_{2n}). While the expressions in Definition B.2 are somewhat involved, the key point is that all the quantities that are required in the proofs in §5 can be expressed as rational functions in z_n, y_{2n}, z_{2n} . This is Lemma B.3. (Note that τ_n, τ_{2n} are by definition rational functions of z_n, z_{2n} , and also note that Ξ is needed only through its row sums.) The upshot is that since z_n, y_{2n}, z_{2n} are polynomially related by construction (see §3.1), these expressions make the rate verification a routine and rigorously automatable algebraic exercise.

Definition B.2 (Corrections to the recursive gluing in Theorem 5.2). The “low-rank correction” Ξ is defined to be zero except that for all $j \in \{n, \dots, 2n-2\}$,

- $\Xi_{n-1,j} := \phi / \prod_{t=n}^{j-1} q_t$
- $\Xi_{2n-1,j} := r\phi / \prod_{t=n}^{j-1} q_t$
- $\Xi_{*,j} := -(1+r)\phi / \prod_{t=n}^{j-1} q_t$

The “sparse correction” Δ is zero everywhere except for the following entries:

- $\Delta_{n-1,2n-1} := \phi \frac{\kappa-2}{\kappa(1-z_n)}$
- $\Delta_{2n-1,n-1} := \phi(\kappa-2) \frac{y_{2n}}{1-z_n}$
- $\Delta_{*,n-1} := \tau_{2n} \frac{1+\kappa y_{2n}}{1-z_n} - \frac{\tau_{2n}}{\tau_n} \sigma_{*,n-1}$
- $\Delta_{*,2n-1} := \frac{1+(\kappa-1)z_{2n}+\kappa z_{2n}^2}{(1+z_{2n})^2} - c\sigma_{*,n-1}$
- $\Delta_{n-1,*} := -\frac{\tau_{2n}}{\tau_n} \sigma_{n-1,*}$
- $\Delta_{2n-1,*} := \frac{2\kappa z_{2n}}{(1+z_{2n})^2} - c\sigma_{n-1,*}$

In the above, we used as shorthand the following special values:

- $r := 1 / \prod_{t=0}^{n-1} q_t$
- $c := \frac{\tau_{2n}}{\tau_n} \left[r + (1+r) \left(\frac{z_{2n}+z_n}{z_{2n}-z_n} \right) \right]$
- $\phi := \tau_{2n} \frac{\kappa}{\kappa-2} \left(\frac{z_{2n}+z_n}{z_{2n}-z_n} \right)$

We now state the alternative expressions discussed in Remark B.1.

Lemma B.3 (Alternative expressions in terms of z_n, y_{2n}, z_{2n}). The following identities hold:

- $\Delta_{n-1,2n-1} = \frac{\tau_{2n}}{1-z_n} \left(\frac{z_{2n}+z_n}{z_{2n}-z_n} \right)$
- $\Delta_{2n-1,n-1} = \frac{\kappa y_{2n} \tau_{2n}}{1-z_n} \left(\frac{z_{2n}+z_n}{z_{2n}-z_n} \right)$
- $\Delta_{*,n-1} = \frac{\tau_{2n}}{1-z_n} (1 + \kappa y_{2n}) - \tau_{2n} \frac{1+(\kappa-1)z_n+\kappa z_n^2}{(1-z_n)^2}$
- $\Delta_{*,2n-1} = \tau_{2n} \frac{1+(\kappa-1)z_{2n}+\kappa z_{2n}^2}{(1-z_{2n})^2} - c\tau_n \frac{1+(\kappa-1)z_n+\kappa z_n^2}{(1-z_n)^2}$
- $\Delta_{n-1,*} = -\frac{2\kappa z_n \tau_{2n}}{(1-z_n)^2}$

- $\Delta_{2n-1,*} = \frac{2\kappa z_{2n} \tau_{2n}}{(1-z_{2n})^2} - c \frac{2\kappa z_n \tau_n}{(1-z_n)^2}$
- $\sum_{j=n}^{2n-2} \Xi_{n-1,j} = \frac{1}{r} \sum_{j=n}^{2n-2} \Xi_{2n-1,j} = -\frac{1}{1+r} \sum_{j=n}^{2n-2} \Xi_{*,j} = \tau_{2n} \left(\frac{\kappa z_n - 1}{1-z_n} \right) \left(\frac{z_{2n} + z_n}{z_{2n} - z_n} \right)$ and is 0 for $n = 1, 2$
- $r = \frac{1-z_n}{1-z_{2n}}$

These equivalent expressions make it straightforward to prove the deferred parts of Theorem 5.2.

Proof of non-negativity and $$ -sparsity in Theorem 5.2.* Checking $*$ -sparsity. Since σ satisfies $*$ -sparsity, it follows immediately that $\lambda_{i,*} = 0$ for all $i < 2n - 1$ except possibly $i = n - 1$. For this remaining case, the definition of the sparse correction Δ ensures $\lambda_{n-1,*} = 0$.

Checking non-negativity. First observe that q_i, r, c, ϕ are all non-negative by the bounds $z_n \leq z_{2n} \leq 1$ in Lemma 3.1 and the bounds $1 \leq a_n, b_n \leq \kappa$ in Lemma 3.2. Next, note that all Θ_{ij} are non-negative since they are a positive multiple of some entry of σ , which is non-negative by assumption of σ being a valid certificate. Thus we need only check non-negativity of $\lambda_{ij} = \Theta_{ij} + \Xi_{ij} + \Delta_{ij}$ on the entries where either the correction Ξ or Δ is non-zero. This non-negativity is clear from the construction for all of the sparsely-corrected entries λ_{ij} where $i \neq j \in \{n-1, 2n-1, *\}$ as well as nearly all of the rank-one-corrected entries λ_{ij} , namely for all $i \in \{n-1, 2n-1\}$ and $j \in \{n, \dots, 2n-2\}$.

It remains only to prove non-negativity of $\lambda_{*,j}$ for $j \in \{n, \dots, 2n-2\}$. Since $\Delta_{*,j} = 0$, this amounts to showing that $\Theta_{*,j} \geq \Xi_{*,j}$, i.e., $c\sigma_{*,j-n} \geq (1+r)\phi / \prod_{t=0}^{j-1} q_t$. This follows by plugging in the explicit formulas for $\sigma_{*,j}$ in Lemma B.3 and the definitions of r, c, ϕ in Theorem 5.2. \square

The rest of this Appendix section is organized as follows. In B.1 and B.2, we provide two helper lemmas. The former explicitly computes the values of all co-coercivity multipliers to/from optimum for the n -step certificate σ . The latter provides useful identities involving sums and products of q_i . We then use these two helper lemmas in B.3 to prove the alternative expressions in Lemma B.3, and in B.4 to prove the succinct quadratic form representations in Lemma 5.3.

B.1 Helper lemma: co-coercivities involving x^*

Here we compute all co-coercivity multipliers $\sigma_{t,*}$ and $\sigma_{*,t}$ between GD iterates x_t and x_* .

Lemma B.4 (Co-coercivity multipliers to/from x^*). *Consider the setup of Theorem 5.2. Then:*

- $\sigma_{j,*} = 0$ for all $j \in \{0, \dots, n-2\}$
- $\sigma_{n-1,*} = \frac{2\kappa z_n}{(1+z_n)^2}$
- $\sigma_{*,j} = \frac{(\prod_{t=j}^{n-3} q_t)(1-z_n)}{(1+z_n)^2}$ for all $j \in \{0, \dots, n-3\}$
- $\sigma_{*,n-2} = \frac{1-z_n}{(1+z_n)^2}$
- $\sigma_{*,n-1} = \frac{1+(\kappa-1)z_n+\kappa z_n^2}{(1+z_n)^2}$

Proof. That $\sigma_{j,*} = 0$ for all $j < n-1$ is trivially due to the assumption that σ satisfies the $*$ -sparsity property (Definition 5.1). The content of the lemma is solving for the other multipliers. To this end, recall that the n -step certificate (5.2) establishes that the two quadratic forms $\tau_n \|x_0\|^2 - \|x_n\|^2$ and $\sum_{i \neq j \in \{0, \dots, n-1, *\}} \sigma_{ij} Q_{ij}$ are equal modulo the ideal generated by the equations $x_{t+1} = x_t - \alpha_t g_t$ for all $t \in \{0, \dots, n-1\}$. Thus if we expand both these quadratic forms by replacing, for every $t > 0$, the iterate x_t with $x_0 - \sum_{s=0}^{t-1} \alpha_s g_s$, then the resulting two quadratic forms (now in the variables $\{x_0, g_0, \dots, g_{n-1}\}$) are equal, and in particular the coefficient of any term must match.

We prove this lemma by solving the equations that come from matching the coefficients for the terms $\|x_0\|^2$ and $\langle x_0, g_i \rangle$ for $i \in \{0, \dots, n-1\}$. By expanding the definition of the co-coercivity, it is evident that only the co-coercivities of the form $Q_{t,*}$ and $Q_{*,t}$ contributes coefficients for these terms. In particular, matching the coefficients for the term $\|x_0\|^2$ gives the equation

$$\tau_n - 1 = -\frac{1}{\kappa} \left(\sigma_{n-1,*} + \sum_{j=0}^{n-1} \sigma_{*,j} \right), \quad (\text{B.1})$$

matching the coefficients for the term $\langle x_0, g_{n-1} \rangle$ gives the equation

$$\alpha_{n-1} = \frac{1}{\kappa} \sigma_{n-1,*} + \sigma_{*,n-1}, \quad (\text{B.2})$$

and matching the coefficients for the other terms $\langle x_0, g_t \rangle$ gives the equations

$$\alpha_t = \frac{\alpha_t}{\kappa} \left(\sigma_{n-1,*} + \sum_{i=t+1}^{n-1} \sigma_{*,i} \right) + \sigma_{*,t}, \quad \forall t \in \{0, \dots, n-2\} \quad (\text{B.3})$$

Intuitively, these equations can be back-solved since they are (essentially) already in triangular form. Below we detail a simple way to do this by hand.

First, we obtain the claimed expression for $\sigma_{n-1,*}$ by combining the netflow equation $\sigma_{n-1,*} = \sum_{j=0}^{n-1} \sigma_{*,j}$ with (B.1), re-arranging, and plugging in the definition of $\tau_n = (\frac{1-z_n}{1+z_n})^2$.

Next, we obtain the claimed expression for $\sigma_{*,n-1}$ by plugging the now-proved value of $\sigma_{n-1,*}$ into (B.2) and using the fact that the Silver Stepsize $\alpha_{n-1} = b_n = (1 + \kappa z_n)/(1 + z_n)$, see §3.

Next, we obtain the claimed expression for $\sigma_{*,n-2}$ by plugging the now-proved values for $\sigma_{*,n-1}$ and $\sigma_{n-1,*}$ into (B.3), for $t = n-2$, and using the fact that Silver Stepsize $\alpha_{n-2} = \kappa/(\kappa - 1)$, see §3.

To solve for the remaining variables $\{\sigma_{*,j}\}_{j \in \{0, \dots, n-3\}}$, we could continue back-solving by plugging into (B.3). However, there is a simpler approach: by subtracting the j -th equation (B.3) from the $(j+1)$ -th equation (B.3), the partial sums telescope. After re-arranging, this gives the recurrence

$$\sigma_{*,j} = q_j \sigma_{*,j+1}, \quad \forall j \in \{0, \dots, n-3\}. \quad (\text{B.4})$$

By plugging in the now-proved value of $\sigma_{*,n-2}$ as the base case for this backwards recurrence, we obtain the claimed expression for $\sigma_{*,j}$ for all $j \in \{0, \dots, n-3\}$. \square

B.2 Helper lemma: identities involving q_i

Here we provide useful identities involving q_i . This enables expressing $r = \prod_{t=0}^{n-1} 1/q_t$ and sums of the form $\sum_{j=n}^{2n-2} \prod_{t=n}^{j-1} 1/q_t$ in terms of the Normalized Silver Stepsizes z_n and z_{2n} . We will use this to prove the final two items of Lemma B.3 in Appendix B.3. Note that for simplicity, we state these identities for $n \geq 4$ since the low-rank component Ξ (which is what this lemma is used to compute) is identically zero for small n .

Lemma B.5 (Identities involving products of q_i). *For $n \geq 4$:*

- $\prod_{t=0}^{n-3} \frac{1}{q_t} = \frac{\kappa-2}{\kappa(1-z_n)}$
- $\prod_{t=0}^{n-1} \frac{1}{q_t} = \frac{1-z_n}{1-z_{2n}} = r$
- $\sum_{j=n}^{2n-2} \prod_{t=n}^{j-1} \frac{1}{q_t} = \frac{(\kappa-2)(\kappa z_n - 1)}{\kappa(1-z_n)}$

The proof exploits the following elementary identity. We remark that this identity has a probabilistic interpretation, although we do not make explicit use of it. This interpretation is for the case $c_t \in [0, 1]$; then c_t can be viewed as the probability that the t -th coin is heads, hence $\sum_{t=0}^T c_t \prod_{i=t+1}^T (1 - c_i) = 1 - \prod_{t=0}^T (1 - c_t)$ are two expressions for the probability that at least one coin is heads. Below, recall the standard convention that the product over an empty set is 1.

Lemma B.6. *For any non-negative integer T and any real numbers c_0, \dots, c_T ,*

$$\sum_{t=0}^T c_t \prod_{i=t+1}^T (1 - c_i) = 1 - \prod_{t=0}^T (1 - c_t).$$

Proof. We prove by induction. The base case $T = 0$ is trivial. For the inductive step, assume true for T ; then the claim holds for $T + 1$ because $\sum_{t=0}^{T+1} c_t \prod_{i=t+1}^{T+1} (1 - c_i) = c_{T+1} + (1 - c_{T+1})(\sum_{t=0}^T c_t \prod_{i=t+1}^T (1 - c_i)) = c_{T+1} + (1 - c_{T+1})(1 - \prod_{t=0}^T (1 - c_t)) = 1 - \prod_{t=0}^{T+1} (1 - c_t)$. \square

Proof of Lemma B.5. Since σ is a valid certificate, it satisfies the netflow constraint $\sum_j \sigma_{*,j} = \sum_j \sigma_{j,*}$. By using the values for these entries (Lemma B.4) and re-arranging, we obtain

$$\sum_{j=0}^{n-2} \prod_{t=j}^{n-3} q_j = \kappa z_n - 1. \quad (\text{B.5})$$

Next, we compute this quantity in a different way. By definition of $q_i = \frac{\alpha_i(1 - \alpha_{i+1}/\kappa)}{\alpha_{i+1}}$, multiplying consecutive q_i yields a telescoping product, namely

$$\prod_{t=j}^{n-3} q_t = \frac{\alpha_j}{\alpha_{n-2}} \prod_{t=j+1}^{n-2} \left(1 - \frac{\alpha_t}{\kappa}\right). \quad (\text{B.6})$$

In particular, this implies that

$$\sum_{j=0}^{n-2} \prod_{t=j}^{n-3} q_j = \frac{\kappa}{\alpha_{n-2}} \left(\sum_{j=0}^{n-2} \frac{\alpha_j}{\kappa} \prod_{t=j+1}^{n-2} \left(1 - \frac{\alpha_t}{\kappa}\right) \right) = (\kappa - 1) \left(1 - \prod_{t=0}^{n-2} \left(1 - \frac{\alpha_t}{\kappa}\right)\right). \quad (\text{B.7})$$

Above, the second step uses Lemma B.6 and the fact that $\alpha_{n-2} = a_2 = \kappa/(\kappa - 1)$ by construction of the Silver Stepsize Schedule (see §3). Now by combining (B.5) and (B.7), we obtain

$$\prod_{t=0}^{n-2} \left(1 - \frac{\alpha_t}{\kappa}\right) = \frac{\kappa(1 - z_n)}{\kappa - 1}.$$

By using again the telescoping property (B.6) and the fact that $\alpha_0 = \alpha_{n-2}$ by construction of the Silver Stepsize Schedule (see §3), we conclude that

$$\prod_{t=0}^{n-3} q_t = \frac{\alpha_0}{\alpha_{n-2}} \prod_{t=1}^{n-2} \left(1 - \frac{\alpha_t}{\kappa}\right) = \frac{\prod_{t=0}^{n-2} (1 - \alpha_t/\kappa)}{1 - \alpha_0/\kappa} = \frac{\kappa(1 - z_n)}{\kappa - 2}. \quad (\text{B.8})$$

This proves the first claim.

For the second claim, observe that

$$q_{n-1} q_{n-2} = \left(1 - \frac{\alpha_n}{\kappa}\right) \left(1 - \frac{\alpha_{n-1}}{\kappa}\right) = \frac{\kappa - 2}{\kappa - 1} \cdot \frac{\kappa - 1}{\kappa(1 + y_{2n})} = \frac{\kappa - 2}{\kappa(1 + y_{2n})},$$

where above we have simplified by using the facts that $\alpha_{n-2} = \alpha_n = a_2 = \kappa/(\kappa - 1)$ and $\alpha_{n-1} = a_{2n}$ by construction of the Silver Stepsize Schedule (see §3), as well as the re-parameterization of y_{2n} in terms of $a_{2n} = \alpha_{n-1}$. Multiplying the above two displays yields

$$\prod_{t=0}^{n-1} q_t = \frac{1 - z_n}{1 + y_{2n}}.$$

The proof of the second claim is then complete by using the identity $(1 - z_n)^2 = (1 - z_{2n})(1 + y_{2n})$ which follows from the recurrence construction of the y_n, z_n sequences in §3.

Finally, for the third claim, divide (B.5) by (B.8) to conclude the desired identity

$$\sum_{j=0}^{n-2} \prod_{t=0}^{j-1} \frac{1}{q_t} = \frac{\sum_{j=0}^{n-2} \prod_{t=j}^{n-3} q_j}{\prod_{t=0}^{n-3} q_t} = \frac{(\kappa - 2)(\kappa z_n - 1)}{\kappa(1 - z_n)}.$$

□

B.3 Recursive gluing as a rational function of z_n, y_{2n}, z_{2n}

Proof of Lemma B.3. The expressions for Δ are immediate from Definition B.2 and Lemma B.4. The expressions for the row sums of Ξ are immediate by definition of Ξ and the final identity in Lemma B.5. The expression for r follows from Lemma B.5. □

B.4 Recursive gluing as a succinct quadratic form

Here we provide details for Lemma 5.3 and its proof. The definitions of the coefficient matrices E, S , and L are as follows. Note that these matrices are symmetric, thus for shorthand we simply write a tilde for their lower-triangular elements.

$$E := \begin{bmatrix} \frac{\tau_{2n}}{\tau_n} - c\tau_n & c\tau_n a_{2n} - \frac{\tau_{2n}}{\tau_n} b_n & 0 & 0 \\ \sim & \frac{\tau_{2n}}{\tau_n} b_n^2 - c\tau_n a_{2n}^2 & 0 & 0 \\ \sim & \sim & c - 1 & b_{2n} - cb_n \\ \sim & \sim & \sim & cb_n^2 - b_{2n}^2 \end{bmatrix}$$

and

$$S := \begin{bmatrix} -\frac{1}{\kappa} \sum_{t \neq n-1} (\Delta_{n-1,t} + \Delta_{t,n-1}) & \sum_{t \neq n-1} \left(\frac{\Delta_{n-1,t}}{\kappa} + \Delta_{t,n-1} \right) & \frac{1}{\kappa} (\Delta_{n-1,2n-1} + \Delta_{2n-1,n-1}) & -\Delta_{n-1,2n-1} - \frac{\Delta_{2n-1,n-1}}{\kappa} \\ \sim & -\sum_{t \neq n-1} (\Delta_{n-1,t} + \Delta_{t,n-1}) & -\Delta_{2n-1,n-1} - \frac{\Delta_{n-1,2n-1}}{\kappa} & \Delta_{n-1,2n-1} + \Delta_{2n-1,n-1} \\ \sim & \sim & -\frac{1}{\kappa} \sum_{t \neq 2n-1} (\Delta_{2n-1,t} + \Delta_{t,2n-1}) & \sum_{t \neq 2n-1} \left(\frac{\Delta_{2n-1,t}}{\kappa} + \Delta_{t,2n-1} \right) \\ \sim & \sim & \sim & -\sum_{t \neq 2n-1} (\Delta_{2n-1,t} + \Delta_{t,2n-1}) \end{bmatrix}$$

and $L := \phi(L^{(n-1)} + rL^{(2n-1)})$, where

$$L^{(n-1)} := \frac{\kappa - 2}{\kappa(1 - z_n)} \begin{bmatrix} z_n - 2 + \frac{1}{\kappa} & a_{2n}(1 - z_n) + \frac{\kappa - 1}{\kappa} & 1 - \frac{1}{\kappa} & 0 \\ \sim & 2a_{2n}(z_n - 1) - (\kappa z_n - 1) & -1 + \frac{1}{\kappa} & 0 \\ \sim & \sim & 0 & 0 \\ \sim & \sim & \sim & 0 \end{bmatrix}$$

$$L^{(2n-1)} := \frac{\kappa - 2}{\kappa(1 - z_n)} \begin{bmatrix} 0 & 0 & -1 + z_n & 1 - z_n \\ \sim & 0 & a_{2n}(1 - z_n) & -a_{2n}(1 - z_n) \\ \sim & \sim & 2 - z_n - \frac{1}{\kappa} & -1 + z_n \\ \sim & \sim & \sim & 1 - \kappa z_n \end{bmatrix}$$

and is zero for $n = 1, 2$.

Proof of remaining parts of Lemma 5.3. The identity for the sparse correction is immediate by plugging in the definition of P_{ij} , simplifying $x^* = g^* = 0$, and collecting terms. It remains to show the identity for the low-rank correction. Suppose $n \geq 4$, else Ξ is identically zero and the claim is trivial. We claim that

$$\sum_{j=n}^{2n-2} \prod_{t=n}^{j-1} \frac{1}{q_j} (P_{n-1,j} - P_{*,j}) = \langle L^{(n-1)}, vv^T \rangle \quad (\text{B.9})$$

$$\sum_{j=n}^{2n-2} \prod_{t=n}^{j-1} \frac{1}{q_j} (P_{2n-1,j} - P_{*,j}) = \langle L^{(2n-1)}, vv^T \rangle. \quad (\text{B.10})$$

These identities suffice because by the definition of Ξ and L , we then have

$$\sum_{ij} \Xi_{ij} P_{ij} = \phi \sum_{j=n}^{2n-2} \prod_{t=n}^{j-1} \frac{1}{q_j} ((P_{n-1,j} - P_{*,j}) - r(P_{2n-1,j} - P_{*,j})) = \langle L, vv^T \rangle.$$

We prove (B.9); the proof of (B.10) is entirely analogous and thus omitted for brevity. By expanding the squares in the definition of P_{ij} , simplifying $x^* = g^* = 0$, and collecting terms,

$$P_{n-1,j} - P_{*,j} = -\frac{1}{\kappa} \|x_{n-1}\|^2 - \|g_{n-1}\|^2 + \frac{2}{\kappa} \langle x_{n-1}, g_{n-1} \rangle + 2 \langle x_{n-1} - g_{n-1}, \frac{x_j}{\kappa} - g_j \rangle.$$

By expanding $x_j = x_{n-1} - \sum_{i=n-1}^{j-1} \alpha_i g_i$ using the definition of GD, and then collecting terms,

$$P_{n-1,j} - P_{*,j} = \frac{1}{\kappa} \|x_{n-1}\|^2 + \left(\frac{2\alpha_{n-1}}{\kappa} - 1 \right) \|g_{n-1}\|^2 - \frac{2\alpha_{n-1}}{\kappa} \langle x_{n-1}, g_{n-1} \rangle - 2 \langle x_{n-1} - g_{n-1}, g_j + \frac{1}{\kappa} \sum_{i=n}^{j-1} \alpha_i g_i \rangle.$$

Since the first three of these four summands are independent of j , we conclude

$$\begin{aligned} \sum_{j=n}^{2n-2} \prod_{t=n}^{j-1} \frac{1}{q_j} (P_{n-1,j} - P_{*,j}) &= \left(\sum_{j=n}^{2n-2} \prod_{t=n}^{j-1} \frac{1}{q_j} \right) \left(\frac{1}{\kappa} \|x_{n-1}\|^2 + \left(\frac{2\alpha_{n-1}}{\kappa} - 1 \right) \|g_{n-1}\|^2 - \frac{2\alpha_{n-1}}{\kappa} \langle x_{n-1}, g_{n-1} \rangle \right) \\ &\quad - 2 \left\langle x_{n-1} - g_{n-1}, \sum_{j=n}^{2n-2} \prod_{t=n}^{j-1} \frac{1}{q_j} \left(g_j + \frac{1}{\kappa} \sum_{i=n}^{j-1} \alpha_i g_i \right) \right\rangle. \end{aligned} \quad (\text{B.11})$$

For the first term, use Lemma B.5 and the fact that $q_j = q_{j-n}$ for $j \in \{n, \dots, 2n-2\}$ to obtain

$$\sum_{j=n}^{2n-2} \prod_{t=n}^{j-1} \frac{1}{q_j} = \sum_{j=0}^{n-2} \prod_{t=0}^{j-1} \frac{1}{q_j} = \frac{(\kappa-2)(\kappa z_n - 1)}{\kappa(1-z_n)}. \quad (\text{B.12})$$

For the second term, observe that

$$\begin{aligned}
\sum_{j=n}^{2n-2} \prod_{t=n}^{j-1} \frac{1}{q_j} \left(g_j + \frac{1}{\kappa} \sum_{i=n}^{j-1} \alpha_i g_i \right) &= \frac{1}{\alpha_n} \sum_{j=n}^{2n-2} \alpha_j \prod_{t=n+1}^j \frac{1}{(1 - \alpha_t/\kappa)} \left(g_j + \frac{1}{\kappa} \sum_{i=n}^{j-1} \alpha_i g_i \right) \\
&= \frac{1}{\alpha_n} \sum_{i=n}^{2n-2} \alpha_i g_i \left[\frac{1}{\prod_{t=n+1}^i (1 - \alpha_t/\kappa)} + \sum_{j=i+1}^{2n-2} \frac{\alpha_j}{\kappa \prod_{t=n+1}^j (1 - \alpha_t/\kappa)} \right] \\
&= \frac{1}{\alpha_n \prod_{t=n+1}^{2n-2} (1 - \alpha_t/\kappa)} \sum_{i=n}^{2n-2} \alpha_i g_i \left[\prod_{t=i+1}^{2n-2} (1 - \alpha_t/\kappa) + \sum_{j=i+1}^{2n-2} \frac{\alpha_j}{\kappa} \prod_{t=j+1}^{2n-2} (1 - \alpha_t/\kappa) \right] \\
&= \frac{1}{\alpha_n \prod_{t=n+1}^{2n-2} (1 - \alpha_t/\kappa)} \sum_{i=n}^{2n-2} \alpha_i g_i \\
&= \frac{1}{\alpha_n \prod_{t=n+1}^{2n-2} (1 - \alpha_t/\kappa)} \left[x_{n-1} - x_{2n-1} - \alpha_{n-1} g_{n-1} \right] \\
&= \frac{(\kappa - 1)(\kappa - 2)}{\kappa^2(1 - z_n)} \left[x_{n-1} - x_{2n-1} - \alpha_{n-1} g_{n-1} \right]. \tag{B.13}
\end{aligned}$$

Above, the first step is by definition of q_j and telescoping. The second step is by re-arranging sums. The third step is by factoring out the product. The fourth step is because $\sum_{j=i+1}^{2n-2} \frac{\alpha_j}{\kappa} \prod_{t=j+1}^{2n-2} (1 - \frac{\alpha_t}{\kappa}) = 1 - \prod_{t=i+1}^{2n-2} (1 - \frac{\alpha_t}{\kappa})$ by Lemma B.6. The fifth step is by definition of GD. The final step uses $\alpha_n \prod_{t=n+1}^{2n-2} (1 - \frac{\alpha_t}{\kappa}) = \alpha_{2n-2} \prod_{t=n}^{2n-3} q_t$, Lemma B.5, and the fact that $\alpha_{2n-2} = a_2 = \kappa/(\kappa - 1)$.

By combining (B.11), (B.12), and (B.13), we conclude that the desired quantity is equal to

$$\begin{aligned}
\sum_{j=n}^{2n-2} \prod_{t=n}^{j-1} \frac{1}{q_j} (P_{n-1,j} - P_{*,j}) &= \frac{\kappa - 2}{\kappa^2(1 - z_n)} \left[(\kappa z_n - 1) \left(\|x_{n-1}\|^2 - 2\alpha_{n-1} \langle x_{n-1}, g_{n-1} \rangle + (2\alpha_{n-1} - \kappa) \|g_{n-1}\|^2 \right) \right. \\
&\quad \left. - 2(\kappa - 1) \langle x_{n-1} - g_{n-1}, x_{n-1} - x_{2n-1} - \alpha_{n-1} g_{n-1} \rangle \right].
\end{aligned}$$

Expanding the inner product and canceling terms completes the proof of (B.9). \square

References

- [1] Acceleration by stepsize hedging – verification of identities computer algebra script. <https://github.io/AccelerationByStepsizeHedging>.
- [2] Naman Agarwal, Surbhi Goel, and Cyril Zhang. Acceleration via fractal learning rate schedules. In *International Conference on Machine Learning*, pages 87–99. PMLR, 2021.
- [3] Zeyuan Allen-Zhu and Elad Hazan. Optimal black-box reductions between optimization objectives. *Advances in Neural Information Processing Systems*, 29, 2016.
- [4] Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv preprint arXiv:1407.1537*, 2014.
- [5] Jason M. Altschuler. Greed, hedging, and acceleration in convex optimization. Master’s thesis, Massachusetts Institute of Technology, 2018.
- [6] Mathieu Barré, Adrien B Taylor, and Francis Bach. Principled analyses and design of first-order methods with inexact proximal operators. *Mathematical Programming*, 201(1-2):185–230, 2023.

- [7] Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.
- [8] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in real algebraic geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 2003. ISBN 3-540-00973-6.
- [9] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.
- [10] D.P. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.
- [11] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.
- [12] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.
- [13] Sébastien Bubeck, Yin Tat Lee, and Mohit Singh. A geometric alternative to Nesterov’s accelerated gradient descent. *arXiv preprint arXiv:1506.08187*, 2015.
- [14] Augustin Cauchy. Méthode générale pour la résolution des systèmes d’équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.
- [15] Bob F Caviness and Jeremy R Johnson. *Quantifier elimination and cylindrical algebraic decomposition*. Springer Science & Business Media, 2012.
- [16] Michael B Cohen, Aaron Sidford, and Kevin Tian. Relative lipschitzness in extragradient methods and a direct recipe for acceleration. *arXiv preprint arXiv:2011.06572*, 2020.
- [17] David Cox, John Little, and Donal O’Shea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.
- [18] Saman Cyrus, Bin Hu, Bryan Van Scyoc, and Laurent Lessard. A robust accelerated optimization algorithm for strongly convex functions. *arXiv preprint arXiv:1710.04753*, 2017.
- [19] Shuvomoy Das Gupta, Bart PG Van Parys, and Ernest K Ryu. Branch-and-bound performance estimation programming: A unified methodology for constructing optimal optimization methods. *arXiv preprint arXiv:2203.07305*, 2022.
- [20] Etienne De Klerk, François Glineur, and Adrien B Taylor. On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions. *Optimization Letters*, 11:1185–1199, 2017.
- [21] Etienne De Klerk, Francois Glineur, and Adrien B Taylor. Worst-case convergence analysis of inexact gradient and Newton methods through semidefinite programming performance estimation. *SIAM Journal on Optimization*, 30(3):2053–2082, 2020.
- [22] Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146:37–75, 2014.
- [23] Jelena Diakonikolas and Michael I Jordan. Generalized momentum-based methods: A Hamiltonian perspective. *SIAM Journal on Optimization*, 31(1):915–944, 2021.
- [24] Jelena Diakonikolas and Lorenzo Orecchia. Accelerated extra-gradient descent: A novel accelerated first-order method. *arXiv preprint arXiv:1706.04680*, 2017.
- [25] Tobin A Driscoll, Kim-Chuan Toh, and Lloyd N Trefethen. From potential theory to matrix iterations in six steps. *SIAM Review*, 40(3):547–578, 1998.
- [26] Yoel Drori and Adrien Taylor. On the oracle complexity of smooth strongly convex minimization. *Journal of Complexity*, 68:101590, 2022.
- [27] Yoel Drori and Adrien B Taylor. Efficient first-order methods for convex minimization: a constructive approach. *arXiv preprint arXiv:1803.05676*, 2018.

- [28] Yoel Drori and Marc Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1-2):451–482, 2014.
- [29] Dmitriy Drusvyatskiy, Maryam Fazel, and Scott Roy. An optimal first order method based on optimal quadratic averaging. *SIAM Journal on Optimization*, 28(1):251–271, 2018.
- [30] Alexandre d’Aspremont, Damien Scieur, and Adrien Taylor. Acceleration methods. *Foundations and Trends® in Optimization*, 5(1-2):1–245, 2021.
- [31] Baptiste Goujaud, Damien Scieur, Aymeric Dieuleveut, Adrien B Taylor, and Fabian Pedregosa. Super-acceleration with cyclical step-sizes. In *International Conference on Artificial Intelligence and Statistics*, pages 3028–3065. PMLR, 2022.
- [32] Benjamin Grimmer. Provably faster gradient descent via long steps. *arXiv preprint arXiv:2307.06324*, 2023.
- [33] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.
- [34] Magnus R Hestenes, Eduard Stiefel, et al. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.
- [35] Bin Hu, Peter Seiler, and Laurent Lessard. Analysis of approximate stochastic gradient using quadratic constraints and sequential semidefinite programs. *arXiv preprint arXiv:1711.00987*, 2017.
- [36] Zdeněk Kalousek. Steepest descent method with random step lengths. *Foundations of Computational Mathematics*, 17(2):359–422, 2017.
- [37] Donghwan Kim and Jeffrey A Fessler. Optimized first-order methods for smooth convex minimization. *Mathematical Programming*, 159(1-2):81–107, 2016.
- [38] Donghwan Kim and Jeffrey A Fessler. On the convergence analysis of the optimized gradient method. *Journal of Optimization Theory and Applications*, 172(1):187–205, 2017.
- [39] M. Kreuzer and L. Robbiano. *Computational commutative algebra. 1.* Springer-Verlag, Berlin, 2000. ISBN 3-540-67733-X.
- [40] VI Lebedev and SA Finogenov. Ordering of the iterative parameters in the cyclical Chebyshev iterative method. *USSR Computational Mathematics and Mathematical Physics*, 11(2):155–170, 1971.
- [41] Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.
- [42] David G Luenberger and Yinyu Ye. *Linear and nonlinear programming*, volume 2. Springer, 1984.
- [43] Céline Moucer, Adrien B. Taylor, and Francis Bach. A systematic approach to Lyapunov analyses of continuous-time models in convex optimization. *SIAM Journal on Optimization*, 2023.
- [44] Arkadii Nemirovskii and David Borisovich Yudin. *Problem complexity and method efficiency in optimization*. Wiley, 1983.
- [45] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 1998.
- [46] Yurii E. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Math. Dokl.*, 27(2):372–376, 1983.
- [47] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.
- [48] Samet Oymak. Provable super-convergence with a large cyclical learning rate. *IEEE Signal Processing Letters*, 28:1645–1649, 2021.
- [49] Weibin Peng and Tianyu Wang. The Nesterov-Spokoiny acceleration: $o(1/k^2)$ convergence without proximal operations. *arXiv preprint arXiv:2308.14314*, 2023.

- [50] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.
- [51] Boris T. Polyak. *Introduction to optimization*. Optimization Software, Inc., 1987.
- [52] Luc Pronzato and Anatoly Zhigljavsky. Gradient algorithms for quadratic optimization with fast convergence rates. *Computational Optimization and Applications*, 50(3):597–617, 2011.
- [53] Theodore J Rivlin. *An introduction to the approximation of functions*. Courier Corporation, 1981.
- [54] Ralph Tyrell Rockafellar. *Convex analysis*. Princeton University Press, 1970.
- [55] Ernest K Ryu, Adrien B Taylor, Carolina Bergeling, and Pontus Giselsson. Operator splitting performance estimation: Tight contraction factors and optimal parameter selection. *SIAM Journal on Optimization*, 30(3):2251–2271, 2020.
- [56] Bin Shi, Simon S Du, Weijie Su, and Michael I Jordan. Acceleration via symplectic discretization of high-resolution differential equations. *Advances in Neural Information Processing Systems*, 32, 2019.
- [57] Bin Shi, Simon S Du, Michael I Jordan, and Weijie J Su. Understanding the acceleration phenomenon via high-resolution differential equations. *Mathematical Programming*, pages 1–70, 2021.
- [58] Leslie N Smith. Cyclical learning rates for training neural networks. In *IEEE Winter Conference on Applications of Computer Vision*, pages 464–472. IEEE, 2017.
- [59] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, pages 369–386. SPIE, 2019.
- [60] Weijie Su, Stephen Boyd, and Emmanuel J Candès. A differential equation for modeling Nesterov’s accelerated gradient method: Theory and insights. *The Journal of Machine Learning Research*, 17(1):5312–5354, 2016.
- [61] Adrien Taylor and Yoel Drori. An optimal gradient method for smooth strongly convex minimization. *Mathematical Programming*, 199(1-2):557–594, 2023.
- [62] Adrien B Taylor. *Convex interpolation and performance estimation of first-order methods for convex optimization*. PhD thesis, Catholic University of Louvain, Louvain-la-Neuve, Belgium, 2017.
- [63] Adrien B Taylor, Julien M Hendrickx, and François Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. *Mathematical Programming*, 161(1-2):307–345, 2017.
- [64] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. URL <http://www.math.washington.edu/tseng/papers/apgm.pdf>, 2008.
- [65] Bryan Van Scoy, Randy A Freeman, and Kevin M Lynch. The fastest known globally convergent first-order method for minimizing strongly convex functions. *IEEE Control Systems Letters*, 2(1):49–54, 2018.
- [66] Richard S Varga. *Matrix Iterative Analysis*, volume 27 of *Springer Series in Computational Mathematics*. Springer, 2000.
- [67] Andre Wibisono, Ashia C Wilson, and Michael I Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.
- [68] Ashia C Wilson, Benjamin Recht, and Michael I Jordan. A Lyapunov analysis of momentum methods in optimization. *arXiv preprint arXiv:1611.02635*, 2016.
- [69] David Young. On Richardson’s method for solving linear systems with positive definite matrices. *Journal of Mathematics and Physics*, 32(1-4):243–255, 1953.
- [70] Anatoly Zhigljavsky, Luc Pronzato, and Elena Bukina. An asymptotically optimal gradient algorithm for quadratic optimization with low computational cost. *Optimization Letters*, 7(6):1047–1059, 2013.