

Accelerating Proximal Gradient Descent via Silver Stepsizes

Jinho Bok Jason M. Altschuler

Abstract

Surprisingly, recent work has shown that gradient descent can be accelerated without using momentum—just by judiciously choosing stepsizes. An open question raised by several papers is whether this phenomenon of stepsize-based acceleration holds more generally for constrained and/or composite convex optimization via projected and/or proximal versions of gradient descent. We answer this in the affirmative by proving that the silver stepsize schedule yields analogously accelerated rates in these settings. These rates are conjectured to be asymptotically optimal among all stepsize schedules, and match the silver convergence rate of vanilla gradient descent [3, 5], namely $O(\varepsilon^{-\log_\rho 2})$ for smooth convex optimization and $O(\kappa^{\log_\rho 2} \log \frac{1}{\varepsilon})$ under strong convexity, where ε is the precision, κ is the condition number, and $\rho = 1 + \sqrt{2}$ is the silver ratio. The key technical insight is the combination of recursive gluing—the technique underlying all analyses of gradient descent accelerated with time-varying stepsizes—with a certain Laplacian-structured sum-of-squares certificate for the analysis of proximal point updates.

1 Introduction

First, consider the fundamental setting of unconstrained smooth convex optimization $\min_x f(x)$, where without loss of generality f is 1-smooth (i.e., ∇f is 1-Lipschitz). Until recently, folklore wisdom argued that the gradient descent algorithm (GD)

$$x_{t+1} = x_t - \alpha_t \nabla f(x_t) \tag{1.1}$$

requires $O(\varepsilon^{-1})$ iterations to converge to an ε -optimal solution, and moreover that any asymptotically faster convergence rate (i.e., “accelerated” rate) requires changing GD by adding momentum, internal dynamics, or other additional building blocks beyond just modifying the stepsizes. See, for example, the textbooks [9, 12, 23, 38] for a further discussion of mainstream approaches for accelerating GD.

Surprisingly, a recent line of work has shown that GD can be accelerated simply by using a judicious choice of stepsizes [2, 3, 5, 16, 24, 25, 26, 27, 28, 29, 52, 56, 57]. In particular, [5] showed that GD can converge at the silver convergence rate $O(\varepsilon^{-\log_\rho 2}) \approx O(\varepsilon^{-0.7864})$ —conjectured to be asymptotically optimal among all possible stepsize schedules—by using the silver stepsize schedule

$$\alpha_t = 1 + \rho^{\nu(t+1)-1}, \tag{1.2}$$

where $\rho := 1 + \sqrt{2}$ denotes the silver ratio, and $\nu(i)$ denotes the 2-adic valuation of i , i.e., the largest integer j such that 2^j divides i . The silver stepsize schedule deviates qualitatively from mainstream stepsize schedules: it is time-varying, nonmonotone, fractal-like, and uses arbitrarily large stepsizes that are in particular larger than 2 (the threshold at which constant stepsize schedules make GD divergent). These results extend in a conceptually identical way if f is additionally strongly convex: the silver stepsize schedule similarly accelerates the convergence of GD from the classical rate $O(\kappa \log \frac{1}{\varepsilon})$ for constant stepsize schedules, to $O(\kappa^{\log_\rho 2} \log \frac{1}{\varepsilon})$, where κ denotes the condition number [3].

In just the past year, there has been an exciting flurry of work that further investigates this phenomenon of stepsize-based acceleration. In particular, [27, 28, 52, 56] improved the hidden constant in the aforementioned rate $O(\varepsilon^{-\log_\rho 2})$ and provided extensions to different performance metrics, and [57] extended final-iterate convergence to anytime convergence. See the related work in §1.3 for a detailed discussion of these results.

The authors are in the Department of Statistics and Data Science, University of Pennsylvania. Work partially supported by a Seed Grant Award from Apple. Email: {jinhobok, alts}@upenn.edu.

Generality of stepsize-based acceleration? However, these results focus primarily on the setting of unconstrained, smooth, (strongly) convex optimization. How general is this phenomenon? In particular, recent papers [3, 25] asked whether stepsize-based acceleration holds in the following two fundamental settings:

- **Constrained optimization via projected GD?** For constrained optimization $\min_{x \in \mathcal{K}} f(x)$, the standard analog of GD is *projected GD*

$$x_{t+1} = \Pi_{\mathcal{K}}(x_t - \alpha_t \nabla f(x_t)) , \quad (1.3)$$

where $\Pi_{\mathcal{K}}$ denotes the projection operator onto the constraint set \mathcal{K} , which is typically assumed to be convex and closed. Can projected GD be similarly accelerated by judiciously choosing stepsizes?

- **Composite optimization via proximal GD?** For composite convex optimization

$$\min_x F(x) := f(x) + h(x)$$

where f is convex and smooth, and h is convex (but potentially non-smooth) and accessible through its proximal operator $\text{prox}_{\alpha h}(x) = \text{argmin}_z h(z) + \frac{1}{2\alpha} \|z - x\|^2$, the standard analog of GD is *proximal GD* (also known as composite GD or backward-forward splitting)

$$x_{t+1} = \text{prox}_{\alpha_t h}(x_t - \alpha_t \nabla f(x_t)) . \quad (1.4)$$

The point of proximal GD is that it enables better convergence than directly applying (sub)gradient descent on F . Indeed, since F is non-smooth, GD on F might not converge and even when it does, the convergence rate is slow; for example, under standard additional assumptions such as bounded gradients, it converges at the rate $O(\varepsilon^{-2})$, whereas proximal GD enjoys a faster convergence rate $O(\varepsilon^{-1})$ using constant stepsizes (see e.g., [9, 22, 55]). Can this be accelerated further to the silver convergence rate using time-varying stepsizes?

Both of these settings are more general than the setting of previous results for stepsize-based acceleration. Indeed, projected GD recovers vanilla GD by taking $\mathcal{K} = \mathbb{R}^d$, and proximal GD recovers vanilla GD by taking $h \equiv 0$. (Note also that proximal GD recovers projected GD by taking h to be the indicator function of the constraint set.) Thus, we cannot hope to accelerate projected GD or proximal GD beyond what is possible for vanilla GD—but can we at least match it? Several recent works [35, 43, 45] have empirically observed that the silver stepsizes (or other time-varying schedules with large steps) can be effective for both settings; however, theoretical guarantees are unknown beyond vanilla GD.

Challenges. At first glance, these extensions might seem straightforward since classical analyses for vanilla, projected, and proximal GD parallel each other closely and lead to nearly identical $O(\varepsilon^{-1})$ convergence rates for constant stepsize schedules. For example, at least intuitively, the classical analysis of projected GD reduces to the analysis of vanilla GD by arguing that since the projection operator is 1-Lipschitz, it cannot worsen the progress towards the minimizer; see e.g., [7, §3.1.2]. However, for stepsize-based acceleration it is unclear how to apply any such arguments. This is because existing analyses for stepsize-based acceleration use more complex dependencies between the iterates, in contrast to the modular argument of combining “GD makes progress” and “interleaving projections does not worsen progress” at each step; see the techniques section §1.2 for a more detailed discussion.

This is not just a failure of current analyses. Interleaving projection and/or proximal operators can change the behavior of GD in fundamental ways when using time-varying stepsizes. For example, [25, §4] numerically investigated stepsize-based acceleration for $n = 2, 3$ steps and observed that projected GD has strictly worse rates than vanilla GD, and moreover proximal GD has strictly worse rates than projected GD. This suggests that as the level of generality increases, the algorithms may have to take strictly less aggressive stepsizes. As another example, even for the seemingly simple setting of quadratic objectives f —a setting for which stepsize-based acceleration has been known for 70+ years [54]—convergence results do not extend in a straightforward way from vanilla GD to projected GD. This is due to fundamental issues beyond just a failure of existing analyses: for example, it can be shown that the optimal $n = 2$ stepsizes for unconstrained quadratic optimization (based on Chebyshev polynomials) provably provide a *divergent* rate if a projection or proximal operation is interleaved, as will be elaborated upon in future work [44].

Hence, it remained unclear whether stepsize-based acceleration extends to more general settings of constrained and/or composite convex optimization via projected and/or proximal versions of GD.

	Convex opt	Constrained convex opt	Composite convex opt
Problem	$\min_x f(x)$	$\min_{x \in \mathcal{K}} f(x)$	$\min_x f(x) + h(x)$
Algorithm	GD	Projected GD	Proximal GD
Update rule	$x_{t+1} = x_t - \alpha_t \nabla f(x_t)$	$x_{t+1} = \Pi_{\mathcal{K}}(x_t - \alpha_t \nabla f(x_t))$	$x_{t+1} = \text{prox}_{\alpha_t h}(x_t - \alpha_t \nabla f(x_t))$
Rate for constant stepsizes	ε^{-1} (folklore)	ε^{-1} (folklore)	ε^{-1} (folklore)
Rate for silver stepsizes	$\varepsilon^{-\log_{\rho} 2}$ [5]	$\varepsilon^{-\log_{\rho} 2}$ (Theorem 1.1)	$\varepsilon^{-\log_{\rho} 2}$ (Theorem 1.1)

TABLE 1: The mainstream convergence rate for GD and its variants is $O(\varepsilon^{-1})$, achieved by constant stepsize schedules. [5] recently showed that GD can be accelerated to $O(\varepsilon^{-\log_{\rho} 2}) \approx O(\varepsilon^{-0.7864})$ for unconstrained smooth convex optimization (left) by using the silver stepsize schedule, conjectured to be asymptotically optimal among all stepsize schedules. It remained open if this phenomenon of stepsize-based acceleration holds in the more general settings of constrained (middle) and composite (right) convex optimization. We show that the answer is yes. In this table, ε is the precision, $\rho = 1 + \sqrt{2}$ is the silver ratio, f and h are convex, and f is 1-smooth (wlog by rescaling). If f is also strongly convex, then the rates improve similarly from $O(\kappa \log 1/\varepsilon)$ for constant stepsizes, to $O(\kappa^{\log_{\rho} 2} \log 1/\varepsilon)$ for the silver stepsize schedule, where κ is the condition number; see Corollary 1.2.

1.1 Contribution

In this paper, we answer these questions in the affirmative. In particular, we show that by using the silver stepsize schedule, projected and proximal GD converge at the silver convergence rate. This was previously only known for vanilla GD. See Table 1 for a summary.

We state our main result for proximal GD (since that generalizes projected GD) and for general M -smoothness (since that is equivalent after normalizing the silver stepsizes by $1/M$). For context, note that the tight rate for proximal GD using constant stepsize schedule $\alpha_t \equiv 1$ is $F(x_n) - F(x_*) \leq \frac{1}{4n} M \|x_0 - x_*\|^2$ where x_* denotes any minimizer of F [50, Theorem 7].

Theorem 1.1 (Main result). *For any horizon $n = 2^k - 1$, any M -smooth convex function f , any convex function h , and any initialization x_0 ,*

$$F(x_n) - F(x_*) \leq \frac{\rho}{4\sqrt{2}n^{\log_2 \rho}} M \|x_0 - x_*\|^2, \quad (1.5)$$

where x_* denotes any minimizer of $F = f + h$, and x_n denotes the n -th iterate of proximal GD using the silver stepsize schedule. In particular, in order to achieve error $F(x_n) - F(x_*) \leq \varepsilon$, it suffices to run proximal GD for some number of iterations n , where

$$n \lesssim \left(\frac{M \|x_0 - x_*\|^2}{\varepsilon} \right)^{\log_{\rho} 2} \approx \left(\frac{M \|x_0 - x_*\|^2}{\varepsilon} \right)^{0.7864}.$$

We make four remarks. First, by standard black-box reductions (see, e.g., [5, Footnote 2]), this acceleration immediately extends to the strongly convex setting. This improves the tight iteration complexity $O(\kappa \log 1/\varepsilon)$ for proximal GD when using constant stepsizes [49], to match the silver convergence rate $O(\kappa^{\log_{\rho} 2} \log 1/\varepsilon)$ that was previously only known for vanilla GD.

Corollary 1.2. *Assume further that f is m -strongly convex. There exists a stepsize schedule such that $\|x_n - x_*\| \leq \varepsilon$ after*

$$n \lesssim \kappa^{\log_{\rho} 2} \log \left(\frac{\|x_0 - x_*\|}{\varepsilon} \right) \approx \kappa^{0.7864} \log \left(\frac{\|x_0 - x_*\|}{\varepsilon} \right)$$

iterations, where $\kappa := M/m$ denotes the condition number.

Second, these acceleration results readily apply to all special cases of proximal GD, e.g.:

- Setting $h \equiv 0$ recovers the silver convergence rate for vanilla GD [3, 5].
- Setting h to be the indicator of a convex set proves the silver convergence rate for projected GD.
- Setting h to be the ℓ_1 norm proves the silver convergence rate for ISTA [17].

For the latter two settings, these are the first results showing stepsize-based acceleration.

Third, we remark that for simplicity, these results are stated for horizons of the form $n = 2^k - 1$. However, the asymptotic rates extend to arbitrary n by standard tricks, e.g., output the best iterate rather than the final iterate, or run for the largest $n' = 2^{k'} - 1$ less than n . Finally, we remark about the tightness of our bounds.

Remark 1.3 (Optimality of the bounds). *In our rates $O(\varepsilon^{-\log_\rho 2})$ and $O(\kappa^{\log_\rho 2} \log 1/\varepsilon)$, the exponent $\log_\rho 2$ is tight for the silver stepsize schedule (proven in Appendix A.1), and is conjecturally optimal among all possible stepsize schedules (since that is conjectured even for the special case of GD [3, 5, 27, 28]).*

These rates for stepsize-based acceleration are weaker than traditional momentum-based acceleration, which changes (vanilla, projected, proximal) GD beyond just modifying stepsizes and achieves rates $O(\varepsilon^{-1/2})$ and $O(\kappa^{1/2} \log 1/\varepsilon)$ for the convex and strongly convex settings, respectively, both of which are information-theoretically optimal among first-order methods [23].

1.2 Overview of techniques

The overarching rationale for stepsize-based acceleration is that although short steps and long steps are individually suboptimal—since short steps can undershoot on flat objectives, and long steps can overshoot on steep objectives—judiciously combining them can lead to overall faster convergence because these worst-cases cannot align. See [2, 3] for an interpretation of this phenomenon in terms of hedging. Notably, proving stepsize-based acceleration precludes using classical analyses (such as those for constant stepsize schedules, see e.g., [7]) which bound the progress of each iteration separately and then sum these bounds. Indeed, any such analysis is unable to prove any benefit of deviating from constant stepsize schedules.

Instead, in order to prove stepsize-based acceleration, it is necessary to argue holistically about the cumulative progress—i.e., the *multi-step descent*—of the algorithm. Let us briefly describe how prior work accomplishes this for GD, before moving to the present setting of proximal GD.

Stepsize acceleration for GD. Suppose one seeks to prove that n steps of GD converges at a rate τ_n , i.e., an inequality $f(x_n) - f(x_*) \leq \tau_n \|x_0 - x_*\|^2$. All existing analyses of stepsize-based acceleration establish this by proving an identity of the form

$$f(x_*) - f(x_n) + \tau_n \|x_0 - x_*\|^2 = \sum_{i,j} \lambda_{ij} Q_{ij} + (\text{SOS}), \quad (1.6)$$

where SOS is a sum-of-squares polynomial, $\lambda_{ij} \geq 0$, and $Q_{ij} \geq 0$ are co-coercivities (i.e., certain polynomials in the iterates x_i and x_j , gradients $\nabla f(x_i)$ and $\nabla f(x_j)$, and function values $f(x_i)$ and $f(x_j)$; see Definition 2.1). The identity (1.6) constitutes a *certificate* of the desired multi-step descent since the right hand side is obviously nonnegative.

In words, the co-coercivity inequality $Q_{ij} \geq 0$ is a long-range consistency condition regarding information about the objective function f at different iterates along the GD trajectory. Crucially, these inequalities are valid not only for consecutive iterates but for *all* x_i and x_j , enabling a holistic argument that captures how different iterations affect each other. In the right hand side of (1.6), $\lambda_{ij} \geq 0$ should be interpreted as multipliers that combine the inequalities $Q_{ij} \geq 0$, and the SOS term should be interpreted as a way to add any other possible valid inequalities. Indeed, although we will not use the following fact in this paper, it turns out that identities of the form (1.6) provide a complete proof system in the sense that any valid convergence rate τ_n admits a proof via such an identity [48].

There are two challenges for proving an identity of the form (1.6): 1) providing explicit expressions for the multipliers $\{\lambda_{ij}\}$ and the SOS term, and 2) verifying the identity, which amounts to checking that the coefficients match on both sides of this polynomial identity. In particular, the issue is that the complexity of both tasks ostensibly increases in the number of iterations n .

The key technique that makes both steps tractable is *recursive gluing*. This technique was introduced in [3, 5] and is at the heart of all analyses in subsequent works on accelerating GD [27, 28, 52, 56]. The underlying idea is that for increasing n , the complexity of proving identities such as (1.6) can be controlled by using the recursive structures of the $\{\lambda_{ij}\}$ and SOS terms that are inherited from the recursive pattern of the stepsizes. In particular, this allows analyzing the performance of a longer stepsize schedule that is created

by concatenating two shorter stepsize schedules—possibly with additional modifications such as changing [3] or adding [5, 27, 28, 52, 56] a few stepsizes—by essentially composing the progress of the shorter stepsize schedules. More precisely, this provides a new identity certifying multi-step descent for the longer stepsize schedule by essentially combining the identities for the shorter schedules. The multipliers $\{\lambda_{ij}\}$ for the new identity are comprised of three components in a formulaic manner: a recursive component (that combines the identities for the shorter subsequences), a sparse correction (that affects only $O(1)$ entries of $\{\lambda_{ij}\}$), and a low-rank correction (that affects only $O(1)$ rows of $\{\lambda_{ij}\}$)—see §3 for details. The upshot is that the majority of the analysis for the longer sequence is inherited from analyses for the shorter sequences, letting one analyze large number of iterations in an inductive or recursive manner. Concretely, this reduces proving an n -step identity of the form (1.6) to verifying only $O(1)$ coefficients.

Stepsize acceleration for proximal GD. We analyze proximal GD by proving a multi-step descent identity analogous to (1.6), but now with a composite objective $F = f + h$, with co-cocoercivity inequalities $Q_{ij}^f \geq 0$ and $Q_{ij}^h \geq 0$ which respectively encode consistency conditions for f and h , and with two sets of corresponding multipliers $\{\lambda_{ij}\}$ and $\{\mu_{ij}\}$. The identity is now of the form

$$F(x_*) - F(x_n) + \tau_n \|x_0 - x_*\|^2 = \sum_{i,j} \lambda_{ij} Q_{ij}^f + \sum_{i,j} \mu_{ij} Q_{ij}^h + (\text{SOS}). \quad (1.7)$$

Our starting point is the observation that we can reuse the same multipliers $\{\lambda_{ij}\}$ from the analysis of GD. However, we still need explicit expressions for $\{\mu_{ij}\}$ and the SOS term. While the multipliers $\{\mu_{ij}\}$ are new and different from $\{\lambda_{ij}\}$, we show that these can also be constructed using the technique of recursive gluing. Combined with $\{\lambda_{ij}\}$, this reduces the complexity of proving the n -step identity into verifying $O(1)$ coefficients as described above.

Conceptually, the SOS term poses a core new difficulty that arises for proximal GD but not GD. Given $\{\lambda_{ij}\}$ and $\{\mu_{ij}\}$, this term can be computed as the residual in the identity (1.7); the challenge is proving that the resulting quadratic form is actually a sum of squares. For GD, this is easy as the term is just a single square [5].¹ In contrast, for proximal GD this term is of *high rank*, implying that any decomposition as a sum of squares (if one exists) requires $\Omega(n)$ terms. Certifying that this term is SOS by finding such a decomposition therefore appears to bring us back to the original challenge of proving multi-step descent identities: the complexity increases in n .

We approach this by combining certain aspects of the analysis of both components of proximal GD: vanilla GD and the proximal point method. This is most easily stated in terms of the coefficient matrix corresponding to this quadratic form; establishing that the term is SOS is then equivalent to showing that this matrix is positive semidefinite. In the case of GD, this matrix has rank 1; in the case of the proximal point method [47], while the matrix has rank $\Omega(n)$, it has an elegant Laplacian structure which implies positive semidefiniteness. It turns out that modulo a Schur complement, the matrix in our analysis of proximal GD is precisely a sum of a rank-1 matrix and a Laplacian matrix that can be constructed recursively. This provides a simple certificate of positive semidefiniteness and thus provides the final missing piece—verifying that the term is indeed SOS—for the recursive gluing argument to prove the desired identity (1.7). We remark that such a combination of structures is novel to analyses of stepsize acceleration, and we are hopeful that such ideas might lead to new possibilities in this line of work.

1.3 Related work

In addition to the literature described above, here we provide further context about related work.

Stepsize-based acceleration. For the special case of quadratic objectives, Young showed in 1953 that GD can be accelerated using non-constant stepsizes, given explicitly as the inverses of the roots of Chebyshev polynomials [54]. However, until recently it was believed that this phenomenon of stepsize-based acceleration was limited to quadratic objectives. Indeed, despite the investigation of many stepsize schedules—including even adaptive strategies such as exact line search [11, 19, 39], Armijo-Goldstein schedules [38], Polyak-type

¹In fact, in the strongly convex case, the SOS term can be made to be zero for GD [3].

schedules [42], and Barzilai-Borwein-type schedules [8]—there had been no analysis showing that any stepsize schedule improves over constant stepsizes in any setting beyond quadratics.

Beginning with [2] in 2018, a line of work showed that such results are in fact possible. [2] showed that time-varying stepsizes can improve over constant stepsize schedules in several settings. Most relevant to the setting of this paper, [2, Chapter 8] provides optimal stepsizes for $n = 2, 3$ iterations for smooth, strongly convex optimization. Cycling through these schedules provides a constant factor improvement over the textbook unaccelerated rate for constant stepsizes. These results were proved via multi-step descent identities of the form (1.6) by leveraging the performance estimation problem (PEP) framework of [21, 48], which uses semidefinite programming to numerically compute the optimal convergence rate of a *given* stepsize schedule. A key difficulty for PEP is that the *search* for optimal stepsizes is nonconvex and computationally difficult as n increases. [16] in 2019 and [24] in 2022 used similar techniques to compute optimal stepsizes for $n = 2, 3$ iterations for the non-strongly convex setting and for different performance metrics. In 2022, [29] developed a branch-and-bound framework for PEP in order to numerically perform this nonconvex search for optimal stepsizes, and used this to compute schedules up to $n = 50$ for the non-strongly convex setting. In July 2023, [25] obtained a constant-factor improvement over constant stepsizes for the non-strongly convex setting by cycling through approximate schedules of length $n = 127$.

In September 2023, two concurrent lines of work showed asymptotic acceleration results for GD: [3, 5] and [26]. The papers [3, 5] proved that GD can accelerate at the silver convergence rate $O(\varepsilon^{-\log_\rho 2}) \approx O(\varepsilon^{-0.7864})$, by introducing the silver stepsize schedule and developing the recursive gluing technique to analyze it. The results are entirely analogous for the strongly convex setting, yielding the rate $O(\kappa^{\log_\rho 2} \log \frac{1}{\varepsilon}) \approx O(\kappa^{0.7864} \log \frac{1}{\varepsilon})$. These rates are conjectured to be asymptotically optimal among all possible stepsize schedules. The other paper [26] proposed a nonperiodic stepsize schedule similar to but slightly different than the silver stepsize schedule, and showed that it accelerates GD at the rate $O(\varepsilon^{-0.9467})$ for smooth convex optimization, and $O(\kappa^{0.9467} \log 1/\varepsilon)$ for the strongly convex setting.

In the year since, a flurry of work has built upon these results in order to further investigate this phenomenon of stepsize acceleration. One thrust has refined the hidden constant in the asymptotic rate $O(\varepsilon^{-\log_\rho 2})$ of [5], by a factor of 2 for the silver stepsize schedule [52], by a factor of roughly 2.32 using a similar stepsize schedule [27], and most recently by a conjecturally optimal factor of roughly 2.37 using highly optimized schedules [28, 56]. A second thrust of these papers [27, 28, 52, 56] extended convergence results from function suboptimality to performance metrics related to the gradient norm. For these results, [28, 56] identified certain families of stepsize schedules that allow structured formulae for concatenating them (and adding a stepsize in the middle), in this way refining and simplifying the technique of recursive gluing. Leveraging these ideas, [57] answered an open question of [34] by concatenating silver stepsize schedules of different lengths in a careful way, in order to establish “anytime convergence” results which ensure accelerated convergence for GD at every iteration rather than just at the final iteration (albeit at a rate $O(\varepsilon^{-0.9672})$ rather than the silver convergence rate). It seems plausible that similar results extend to proximal GD.

All these papers² focus on unconstrained minimization of smooth and (strongly) convex objectives; the purpose of the present paper is to go beyond this setting.

First-order methods for composite convex optimization. Minimizing composite objectives is a fundamental problem with many applications, and using the proximal operator of the nonsmooth function h is a central component in the design and analysis of first-order methods; for an overview, see the survey [41]. A prominent example is where h is the ℓ_1 norm, in which case proximal GD is known as ISTA [17]. The seminal work [10] proposed the FISTA algorithm and showed an $O(\varepsilon^{-1/2})$ accelerated convergence rate. Since then, many algorithms have been proposed; see for example the textbook [9, Chapter 10]. A recent line of work has used PEP not only to obtain tight analyses of existing algorithms [23, 36, 47, 49], but also to develop an optimal algorithm with rate $O(\varepsilon^{-1/2})$ that has the best possible constant factor and an exactly matching lower bound [31]. As is the case for GD, acceleration results in the composite setting are obtained

²The one exception is the recent paper [52] which shows that the relaxed proximal point algorithm (RPPA) can achieve the silver convergence rate for unconstrained convex (but possibly nonsmooth) f . However, this does not imply our results as it requires access to a stronger oracle: the proximal operator for the entire objective. Indeed, as detailed in [52, §1.1], RPPA on f is equivalent to GD on the (smooth and convex) Moreau envelope of f , which means that its setting is in a formal sense equivalent to unconstrained smooth convex optimization—the setting of previous stepsize-based acceleration results for GD. See also the discussion around [52, Theorem 5.3].

by changing proximal GD beyond just changing modifying, e.g., via momentum.

Time-varying and/or large stepsizes in other contexts. Time-varying and/or large stepsizes have also been investigated in other contexts, such as accelerating GD when the objective has Hessians with bimodal [40] or multiscale [32] spectral structure, when the objective is separable [2, 4], when the objective has fourth order growth [18], as well as for logistic regression with separable data [6, 53], parametric convex optimization [43, 45], and perhaps most famously, training of deep neural networks [30, 46]. It is well-documented that large steps for (stochastic) GD can lead to phenomena such as the Edge of Stability [15], implicitly biased convergence towards flat minima [33], and chaos or catapults along the trajectory [13, 37, 51].

Interestingly, recent empirical results have documented that the silver stepsize schedule performs well in constrained settings [43]; also, across several benchmarks, hyperparameter tuning algorithms empirically learn time-varying stepsizes schedules for projected and proximal GD that qualitatively resemble the silver stepsize schedule in certain aspects such as the use of large but rare stepsizes, and fractal-like or near-periodic structure [35, 43]. This paper provides theoretical justification for these empirical phenomena.

2 Preliminaries and notations

2.1 Notation

For shorthand, we denote the sum of a finite sequence $a = \{a_i\}$ by $\sum a$. For vectors a, b of the same length, the notation $a \geq b$ means $a_i \geq b_i$ for all i . Vectors are assumed to be vertical. We use e_1, e_2, \dots to denote the standard basis vectors; the ambient dimension will always be clear from context.

For simplicity, throughout we assume $M = 1$; this is without loss of generality after rescaling the functions f and h by $1/M$. Henceforth, with a slight abuse of notation, $\{\alpha_i\}$ denotes the (normalized) silver stepsizes.

2.2 Co-coercivities

For shorthand, we denote $f_i := f(x_i)$, $g_i := \nabla f(x_i)$, $h_i := h(x_i)$, and s_i to be a subgradient of h at x_i , for all $i \in \{0, 1, \dots, n, *\}$. In this notation, proximal GD (1.4) has the update

$$x_{t+1} = x_t - \alpha_t(g_t + s_{t+1}),$$

and the optimality condition is $g_* + s_* = 0$.

As mentioned in §1.2, our analysis uses certain inequalities—called *co-coercivities*—between pairs of iterates. These inequalities play a central role in the performance estimation framework of [20], and are necessary and sufficient conditions for the interpolability of a (smooth) convex function based only on function and gradient evaluations on a finite set of points [48].

Definition 2.1 (Co-coercivities). *Define*

$$\begin{aligned} Q_{ij}^f &:= f_i - f_j - \langle g_j, x_i - x_j \rangle - \frac{1}{2} \|g_i - g_j\|^2, \\ Q_{ij}^h &:= h_i - h_j - \langle s_j, x_i - x_j \rangle. \end{aligned}$$

Lemma 2.2 ([48, Theorem 4]). *Let f be convex and 1-smooth, and let h be convex. Then $Q_{ij}^f \geq 0$ and $Q_{ij}^h \geq 0$ for all i, j .*

2.3 Helper lemmas

For each $k \in \mathbb{N}$ and $n = 2^k - 1$, we denote by $\pi^{(k)} := [\alpha_0, \dots, \alpha_{n-1}]$ the first n entries of the silver stepsize schedule. We make use of its following basic properties from [5].

Lemma 2.3 (Properties of the silver stepsize schedule $\pi^{(k)}$). *For all $k \in \mathbb{N}$:*

(a) *Sum.* $\sum \pi^{(k)} = \rho^k - 1$.

(b) Recursive construction. $\pi^{(k+1)} = [\pi^{(k)}, \rho^{k-1} + 1, \pi^{(k)}]$.

In order to analyze the silver stepsize schedule for proximal GD, it is helpful to introduce the following auxiliary sequence $c^{(k)}$. Roughly speaking, its first half is identical to the silver stepsize schedule, whereas the second half resembles a more irregular fractal and attains a large maximum at the end.

Definition 2.4. For each $k \in \mathbb{N}$ and $n = 2^k - 1$, the sequence $c^{(k)} \in \mathbb{R}^n$ is recursively defined as

$$\begin{aligned} c^{(1)} &:= 2(\rho - 1), \\ c^{(k+1)} &:= [\pi^{(k)}, (1 + \frac{1}{\rho^k})(\rho^{k-1} + 1), \rho c^{(k)} - (\rho - 1 - \frac{1}{\rho^k})\pi^{(k)}]. \end{aligned}$$

In our analysis, we make repeated use of the following simple properties of $c^{(k)}$.

Lemma 2.5 (Properties of $c^{(k)}$). Let $c^{(k)}$ be as in Definition 2.4.

(a) Sum. $\sum c^{(k)} = 2(\rho^k - 1)$.

(b) Domination. $c^{(k)} \geq \pi^{(k)}$.

Proof. For (a), by induction, $\sum c^{(k+1)} = (\rho^k - 1) + (1 + \frac{1}{\rho^k})(\rho^{k-1} + 1) + 2\rho(\rho^k - 1) - (\rho - 1 - \frac{1}{\rho^k})(\rho^k - 1) = 2(\rho^{k+1} - 1)$ from $\sum \pi^{(k)} = \rho^k - 1$ (Lemma 2.3). For (b), this is straightforward by induction and the recursive construction of $\pi^{(k)}$ (Lemma 2.3). \square

3 Proof of the main result

In this section, we prove the main result (Theorem 1.1) by constructing nonnegative multipliers and sum-of-squares terms such that the following multi-step descent identity holds. These terms capture how different iterations affect other iteration's progress, and in particular capture effects beyond consecutive iterations. See §1.2 for a further discussion and a high-level overview of our analysis techniques.

Theorem 3.1 (Certificate of multi-step descent). Let $k \in \mathbb{N}$ and $n = 2^k - 1$. There exist $\lambda = \lambda^{(k)}$ (Definition 3.2), $\mu = \mu^{(k)}$ (Definition 3.3), and $S = S^{(k)}$ (Definition 3.5) such that the following hold:

(i) Nonnegativity of multipliers. $\lambda_{i,j}, \mu_{i,j} \geq 0$ for all $i \neq j$.

(ii) Positive semidefiniteness of slack. $S \succeq 0$.

(iii) Multi-step descent identity.

$$\sum_{i,j} \lambda_{i,j} Q_{ij}^f + \sum_{i,j} \mu_{i,j} Q_{ij}^h = (2\rho^k - 1)(F_* - F_n) + \frac{\rho}{2\sqrt{2}} \|x_0 - x_*\|^2 - \frac{1}{2} (\|u\|^2 + \text{Tr}(VSV^T)), \quad (3.1)$$

where for shorthand $u := x_0 - x_* - \sum_{i=0}^{n-1} \alpha_i g_i - \rho^k g_n - \sum_{j=1}^n c_j s_j - s_*$ with $c := c^{(k)}$ (Definition 2.4), and $V := [x_0 - x_* \mid s_1 \mid \dots \mid s_n \mid s_*]$ denotes the matrix obtained by concatenating these column vectors.

Theorem 3.1 immediately implies the main result (Theorem 1.1).

Proof of Theorem 1.1. The left hand side of the multi-step descent identity (3.1) is nonnegative since for all $i \neq j$, the co-coercivities Q_{ij}^f, Q_{ij}^h are nonnegative (Lemma 2.2) and so are the multipliers $\lambda_{i,j}, \mu_{i,j}$ (by item (i)). The term $\|u\|^2 + \text{Tr}(VSV^T)$ on the right hand side is also nonnegative as a sum of squares, since $S = S^{(k)}$ is positive semidefinite (by item (ii)). Therefore, by rearranging, (3.1) implies the desired bound

$$F_n - F_* \leq \frac{\rho}{\sqrt{2}(4\rho^k - 2)} \|x_0 - x_*\|^2 \leq \frac{\rho}{4\sqrt{2}n^{\log_2 \rho}} \|x_0 - x_*\|^2.$$

\square

The rest of the section is devoted to establishing Theorem 3.1. In §3.1 we define the multipliers λ, μ and prove their nonnegativity (item (i)). In §3.2 we define the coefficient matrix S for the sum-of-squares term and prove that S is positive semidefinite (item (ii)). In §3.3 we turn to proving the multi-step descent identity (item (iii)). The definitions are recursive in k and the proofs follow by induction on k ; the base case $k = 1$ for Theorem 3.1 amounts to an elementary identity and is deferred to Appendix B.1.

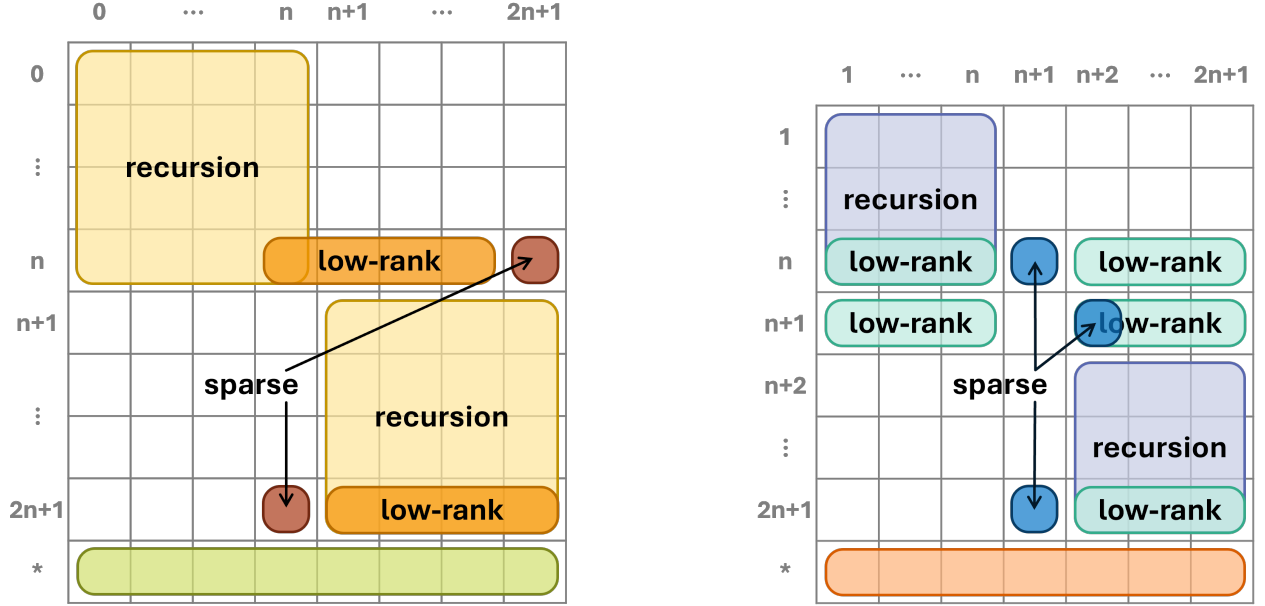


FIGURE 1: Illustration of the structure of $\lambda^{(k+1)}$ (left) and $\mu^{(k+1)}$ (right). Both are constructed via the *recursive gluing* technique, which combines three components. The recursion component glues two copies from $\lambda^{(k)}$ or $\mu^{(k)}$, respectively. The sparse component consists of $O(1)$ nonzero entries. The low-rank component consists of linear combinations of $\pi^{(k)}$ and $c^{(k)}$. Intuitively, the latter two components enable “gluing” the two recursive components by controlling the long-range effect of the large step in the middle. Altogether, this allows inductively establishing the desired convergence rate for $2n + 1 = 2^{k+1} - 1$ given $n = 2^k - 1$.

3.1 Construction of multipliers via recursive gluing

First, we define the multipliers for the co-coercivities Q_{ij}^f associated with f . Although we are considering the more general setting of proximal GD for composite optimization, it turns out that we can use the same multipliers from previous works on GD for single-objective convex optimization. Common to all such analyses, $\lambda^{(k+1)}$ is constructed via recursive gluing as a sum of three components: a recursion component which glues two copies from $\lambda^{(k)}$, a sparse correction which has $O(1)$ nonzero entries, and a low-rank correction which affects only $O(1)$ rows. The original recursive gluing construction of [5] was refined recently in [27, 52], and we use this version here; in particular, the sparse correction has exactly two nonzero entries, and the low-rank correction contains two rows that are multiples of $\pi^{(k)}$. For bookkeeping purposes, it is convenient to isolate the bottom row $\{\lambda_{*,j}\}$ of λ into a separate vector $\underline{\lambda}$. See Figure 1, left, for an illustration.

Definition 3.2 (Multipliers for f). *For $k \in \mathbb{N}$ and $n = 2^k - 1$, define $\lambda_{i,j}^{(k)}$, $i \in \{0, 1, \dots, n, *\}$, $j \in \{0, 1, \dots, n\}$ as*

$$\lambda_{i,j}^{(k)} := \bar{\lambda}_{i,j}^{(k)} \mathbf{1}_{\{i \neq *\}} + \underline{\lambda}_j^{(k)} \mathbf{1}_{\{i = *\}},$$

where $\underline{\lambda}^{(k)} := [\pi^{(k)}, \rho^k]$ and

$$\bar{\lambda}^{(1)} := \begin{matrix} 0 & 1 \\ 0 & \rho \\ 1 & 0 \end{matrix},$$

$$\bar{\lambda}^{(k+1)} := \underbrace{\bar{\lambda}^{(k+1),\text{rec}}}_{\text{recursion}} + \underbrace{\bar{\lambda}^{(k+1),\text{sp}}}_{\text{sparse correction}} + \underbrace{\bar{\lambda}^{(k+1),\text{lr}}}_{\text{low-rank correction}}.$$

The multipliers $\bar{\lambda}^{(k+1),\text{rec}}, \bar{\lambda}^{(k+1),\text{sp}}, \bar{\lambda}^{(k+1),\text{lr}}$ are defined as

$$\begin{aligned}\bar{\lambda}_{i,j}^{(k+1),\text{rec}} &:= \bar{\lambda}_{i,j}^{(k)} \mathbf{1}_{\{0 \leq i, j \leq n\}} + \rho^2 \bar{\lambda}_{i-n-1, j-n-1}^{(k)} \mathbf{1}_{\{n+1 \leq i, j \leq 2n+1\}}, \\ \bar{\lambda}_{i,j}^{(k+1),\text{sp}} &:= \rho \mathbf{1}_{\{(i,j)=(n,2n+1)\}} + \rho^k \mathbf{1}_{\{(i,j)=(2n+1,n)\}}, \\ \bar{\lambda}_{i,j}^{(k+1),\text{lr}} &:= \rho \pi_{j-n}^{(k)} \mathbf{1}_{\{i=n, n+1 \leq j \leq 2n\}} + \rho \pi_{j-n}^{(k)} \mathbf{1}_{\{i=2n+1, n+1 \leq j \leq 2n\}}.\end{aligned}$$

Next, we recursively define the multipliers for the co-coercivities Q_{ij}^h associated with h , which are new to our analysis for *proximal* GD. Similarly to $\lambda^{(k+1)}$, this construction of $\mu^{(k+1)}$ is based on recursive gluing: we use a recursion component that glues two copies from $\mu^{(k)}$, a sparse correction component which has three nonzero entries, and a low-rank correction component which is comprised of linear combinations of $\pi^{(k)}$ and $c^{(k)}$. As for λ , it is helpful for bookkeeping purposes to isolate the bottom row $\{\mu_{*,j}\}$ of μ into a separate vector $\underline{\mu}$. See Figure 1, right, for an illustration.

Definition 3.3 (Multipliers for h). *For $k \in \mathbb{N}$ and $n = 2^k - 1$, define $\mu_{i,j}^{(k)}, i \in \{1, \dots, n, *\}, j \in \{1, \dots, n\}$ as*

$$\mu_{i,j}^{(k)} := \bar{\mu}_{i,j}^{(k)} \mathbf{1}_{\{i \neq *\}} + \underline{\mu}_j^{(k)} \mathbf{1}_{\{i = *\}},$$

where $\underline{\mu}^{(k)} := c^{(k)} + e_1$ and

$$\begin{aligned}\bar{\mu}^{(1)} &:= \mathbf{1} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \\ \bar{\mu}^{(k+1)} &:= \underbrace{\bar{\mu}^{(k+1),\text{rec}}}_{\text{recursion}} + \underbrace{\bar{\mu}^{(k+1),\text{sp}}}_{\text{sparse correction}} + \underbrace{\bar{\mu}^{(k+1),\text{lr}}}_{\text{low-rank correction}}.\end{aligned}$$

The multipliers $\bar{\mu}^{(k+1),\text{rec}}, \bar{\mu}^{(k+1),\text{sp}}, \bar{\mu}^{(k+1),\text{lr}}$ are defined as

$$\begin{aligned}\bar{\mu}_{i,j}^{(k+1),\text{rec}} &:= \bar{\mu}_{i,j}^{(k)} \mathbf{1}_{\{1 \leq i, j \leq n\}} + \rho^2 \bar{\mu}_{i-n-1, j-n-1}^{(k)} \mathbf{1}_{\{n+2 \leq i, j \leq 2n+1\}}, \\ \bar{\mu}_{i,j}^{(k+1),\text{sp}} &:= \rho^k \mathbf{1}_{\{(i,j)=(n,n+1)\}} + \rho^2 \mathbf{1}_{\{(i,j)=(n+1,n+2)\}} + \left(\rho - \frac{1}{\rho^k}\right) (\rho^{k-1} + 1) \mathbf{1}_{\{(i,j)=(2n+1,n+1)\}}, \\ \bar{\mu}_{i,j}^{(k+1),\text{lr}} &:= \left(1 - \frac{\rho^k}{\rho^{k-1} + 1}\right) (c_j^{(k)} - \pi_j^{(k)}) \mathbf{1}_{\{i=n, 1 \leq j \leq n\}} + \frac{\rho^k}{\rho^{k-1} + 1} (c_j^{(k)} - \pi_j^{(k)}) \mathbf{1}_{\{i=n+1, 1 \leq j \leq n\}} \\ &\quad + \frac{\rho^k}{\rho^{k-1} + 1} \pi_{j-n-1}^{(k)} \mathbf{1}_{\{i=n, n+2 \leq j \leq 2n+1\}} + \frac{\rho}{\rho^{k-1} + 1} \pi_{j-n-1}^{(k)} \mathbf{1}_{\{i=n+1, n+2 \leq j \leq 2n+1\}} \\ &\quad + \left((\rho + 1)c_{j-n-1}^{(k)} - \left(1 + \frac{1}{\rho^k}\right)\pi_{j-n-1}^{(k)}\right) \mathbf{1}_{\{i=2n+1, n+2 \leq j \leq 2n+1\}}.\end{aligned}$$

The recursive construction of these multipliers ensures nonnegativity, proving item (i) of Theorem 3.1.

Lemma 3.4 (Nonnegativity of multipliers). *For all $k \in \mathbb{N}$ and $i \neq j$, it holds that $\lambda_{i,j}^{(k)} \geq 0$ and $\mu_{i,j}^{(k)} \geq 0$.*

Proof. This follows by induction on k . The base case $k = 1$ is trivial. For the inductive step, assuming $\lambda_{i,j}^{(k)}$ is nonnegative, $\lambda_{i,j}^{(k+1)}$ is clearly nonnegative as the entries are constructed by adding nonnegative numbers. This is similarly true for nearly all entries of $\mu^{(k+1)}$ from the inequality $c^{(k)} \geq \pi^{(k)}$ (Lemma 2.5). It only remains to check the $n - 1$ entries $\{\mu_{n,j}^{(k+1)}\}_{1 \leq j < n}$ where a nonpositive vector is added due to $\bar{\mu}^{(k+1),\text{lr}}$. A quick calculation shows that these entries are nonnegative after summation; details in Appendix A.2. \square

3.2 Laplacian sum-of-squares

Here we provide details on the slack matrix $S^{(k)}$ in the multi-step descent identity (3.1). In particular we show that it is positive semidefinite, proving item (ii) of Theorem 3.1. Key to this result is the identification of a Laplacian matrix from its recursive structure.³

³Recall that a Laplacian matrix is a symmetric matrix such that all nondiagonal entries are nonpositive, and all row sums (equivalently, all column sums) are 0. It is a classical fact that Laplacian matrices are positive semidefinite; this follows, e.g., from the Gershgorin circle theorem or from the observation $x^T L x = \sum_{i < j} (-L_{ij})(x_i - x_j)^2 \geq 0$. We refer to standard textbooks such as [14] for further background about Laplacian matrices.

We begin by defining $S^{(k)}$. Similarly to the construction of the multipliers in §3.1, this construction is recursive and involves $\pi^{(k)}$ and $c^{(k)}$.

Definition 3.5 (Construction of $L^{(k)}$ and $S^{(k)}$). *For $k \in \mathbb{N}$ and $n = 2^k - 1$, define*

$$L^{(k)} := \begin{bmatrix} \bar{L}^{(k)} & -(c^{(k)})^T \\ -c^{(k)} & 2(\rho^k - 1) \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n+1)},$$

where

$$\begin{aligned} \bar{L}^{(1)} &:= [2(\rho - 1)], \\ \bar{L}^{(k+1)} &:= \begin{bmatrix} \bar{L}^{(k)} + (c^{(k)} - \pi^{(k)})(c^{(k)} - \pi^{(k)})^T & \mathbf{0}_{n \times 1} & \mathbf{0}_{n \times n} \\ \mathbf{0}_{1 \times n} & 0 & \mathbf{0}_{1 \times n} \\ \mathbf{0}_{n \times n} & \mathbf{0}_{n \times 1} & \rho^2(\bar{L}^{(k)} + (c^{(k)} - \pi^{(k)})(c^{(k)} - \pi^{(k)})^T) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{0}_{n \times n} & -\rho^k(c^{(k)} - \pi^{(k)}) & \mathbf{0}_{n \times n} \\ -\rho^k(c^{(k)} - \pi^{(k)})^T & (\rho^{k-1} + 1)(\rho^{k+1} + 1) & -\rho(\pi^{(k)})^T \\ \mathbf{0}_{n \times n} & -\rho\pi^{(k)} & \mathbf{0}_{n \times n} \end{bmatrix} - (c^{(k+1)} - \pi^{(k+1)})(c^{(k+1)} - \pi^{(k+1)})^T. \end{aligned}$$

Also, define

$$S^{(k)} := \begin{bmatrix} \frac{1}{\sqrt{2}} & (-e_1 + e_{n+1})^T \\ -e_1 + e_{n+1} & L^{(k)} \end{bmatrix} \in \mathbb{R}^{(n+2) \times (n+2)}.$$

The following lemma proves several properties of these matrices, culminating in the fact that $S^{(k)}$ is positive semidefinite. The main idea is to observe that $L^{(k)}$ is Laplacian (by the recursive construction), and then use that to analyze $S^{(k)}$ by taking a Schur complement.

Lemma 3.6 (Properties of $L^{(k)}$ and $S^{(k)}$). *The following hold for all $k \in \mathbb{N}$.*

- (a) *All nondiagonal entries of $\bar{L}^{(k)} + (c^{(k)} - \pi^{(k)})(c^{(k)} - \pi^{(k)})^T$ are nonpositive (and thus similarly for $\bar{L}^{(k)}$ and $L^{(k)}$).*
- (b) *$L^{(k)}$ is Laplacian.*
- (c) *$S^{(k)}$ is positive semidefinite.*

Proof. We prove the items in order.

- (a) This follows by induction. The base case $k = 1$ is clear by inspection. The induction step follows from the recursive definition of $\bar{L}^{(k)}$ and the fact that $c^{(k)} \geq \pi^{(k)}$ by Lemma 2.5.
- (b) By item (a), it suffices to show that each row sum of $L^{(k)}$ is 0. Note that for any k , the sum of the final row is $-\sum c^{(k)} + 2(\rho^k - 1) = 0$ by Lemma 2.5. The other rows also sum to zero, as can be checked via a straightforward calculation from the recursive construction; details in Appendix A.3.
- (c) Since the top-left entry of $S^{(k)}$ is positive, it suffices to check that the corresponding Schur complement

$$L^{(k)} - \sqrt{2}(-e_1 + e_{n+1})(-e_1 + e_{n+1})^T$$

is positive semidefinite. In particular, we show that this matrix is Laplacian. Since both $L^{(k)}$ and $\sqrt{2}(-e_1 + e_{n+1})(-e_1 + e_{n+1})^T$ are Laplacian by item (b), each row sum of the Schur complement is 0. For the nondiagonal entries, it suffices to check that the $(1, n+1)$ entry of the Schur complement is nonpositive; this is straightforward since $c_1^{(k)} \geq \sqrt{2}$ for all k .

□

3.3 Verification of the multi-step descent identity

Here we provide a proof sketch of item (iii) of Theorem 3.1; full details are deferred to Appendix B. Item (iii) requires verifying the multi-step descent identity (3.1), which amounts to checking that the coefficients match on both sides of the identity. This identity has two components: a linear form (in the function evaluations $\{f_i\}$ and $\{h_i\}$), plus a quadratic form (in the gradient evaluations $\{g_i\}$, subgradient evaluations $\{s_i\}$, and initial distance $x_0 - x_*$). Thus, it suffices to separately check that the coefficients match for the linear and quadratic forms.

For the linear form, this is simple. The coefficients in $\{f_i\}$ match because they are linear combinations of only the multipliers $\{\lambda_{i,j}\}$ (but not $\{\mu_{i,j}\}$). As they are identical to those for the analysis of vanilla GD, we can appeal to existing results. The coefficients in $\{h_i\}$ can be seen to match by simply expanding and collecting terms from the definition of co-coercivities; see Appendix B.3 for details.

For the quadratic form, verifying the identity is more involved since it requires checking that $\Theta(n^2)$ coefficients match, for $n = 2^k - 1$. A key observation that makes this verification tractable is that if we combine terms in a certain order, then both sides of the identity become “succinct” quadratic polynomials in $O(1)$ variables; in particular, it suffices to check that the corresponding $O(1)$ coefficients match. This conciseness is crucially due to the construction of the multipliers based on the recursive gluing technique. The proof therefore amounts to (1) computing the *constant* number of coefficients of these variables on both sides of the identity, and (2) checking that they match. Both steps are conceptually simple (albeit tedious); details are deferred to Appendices B.4 and B.5, respectively.

References

- [1] Acceleration by stepsize hedging – verification of identities computer algebra script. <https://jasonaltschuler.github.io/AccelerationByStepsizeHedging>.
- [2] Jason M. Altschuler. Greed, hedging, and acceleration in convex optimization. Master’s thesis, Massachusetts Institute of Technology, 2018.
- [3] Jason M. Altschuler and Pablo A. Parrilo. Acceleration by stepsize hedging I: Multi-step descent and the silver stepsize schedule. *arXiv preprint arXiv:2309.07879*, to appear in *Journal of the ACM*, 2023.
- [4] Jason M. Altschuler and Pablo A. Parrilo. Acceleration by random stepsizes: hedging, equalization, and the arcsine stepsize schedule. *arXiv preprint*, 2024.
- [5] Jason M. Altschuler and Pablo A. Parrilo. Acceleration by stepsize hedging: Silver stepsize schedule for smooth convex optimization. *Mathematical Programming*, 2024.
- [6] Kyriakos Axiotis and Maxim Sviridenko. Gradient descent converges linearly for logistic regression on separable data. In *International Conference on Machine Learning*, volume 202 of *PMLR*, pages 1302–1319, 2023.
- [7] Nikhil Bansal and Anupam Gupta. Potential-function proofs for gradient methods. *Theory of Computing*, 15(4):1–32, 2019.
- [8] Jonathan Barzilai and Jonathan M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.
- [9] Amir Beck. *First-order methods in optimization*, volume 25 of *MOS-SIAM Series on Optimization*. Society for Industrial and Applied Mathematics, 2017.
- [10] 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.
- [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] Xuxing Chen, Krishna Balasubramanian, Promit Ghosal, and Bhavya Kumar Agrawalla. From stability to chaos: Analyzing gradient descent dynamics in quadratic regression. *Transactions on Machine Learning Research*, 2024.
- [14] Fan R. K. Chung. *Spectral graph theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. Conference Board of the Mathematical Sciences by the American Mathematical Society, 1997.
- [15] Jeremy Cohen, Simran Kaur, Yuanzhi Li, Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.
- [16] Antoine Daccache. Performance estimation of the gradient method with fixed arbitrary step sizes. Master’s thesis, UCL - Ecole polytechnique de Louvain, 2019.
- [17] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.
- [18] Damek Davis, Dmitriy Drusvyatskiy, and Liwei Jiang. Gradient descent with adaptive stepsize converges (nearly) linearly under fourth-order growth. *arXiv preprint arXiv:2409.19791*, 2024.
- [19] 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(7):1185–1199, 2017.
- [20] Yoel Drori. *Contributions to the complexity analysis of optimization algorithms*. PhD thesis, Tel Aviv University, 2014.
- [21] 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.
- [22] Yoel Drori and Marc Teboulle. An optimal variant of Kelley’s cutting-plane method. *Mathematical Programming*, 160(1-2):321–351, 2016.
- [23] Alexandre d’Aspremont, Damien Scieur, and Adrien Taylor. Acceleration methods. *Foundations and Trends® in Optimization*, 5(1-2):1–245, 2021.
- [24] Diego Eloi. Worst-case functions for the gradient method with fixed variable step sizes. Master’s thesis, UCL - Ecole polytechnique de Louvain, 2022.
- [25] Benjamin Grimmer. Provably faster gradient descent via long steps. *SIAM Journal on Optimization*, 34(3):2588–2608, 2024.
- [26] Benjamin Grimmer, Kevin Shu, and Alex L. Wang. Accelerated gradient descent via long steps. *arXiv preprint arXiv:2309.09961*, 2023.
- [27] Benjamin Grimmer, Kevin Shu, and Alex L. Wang. Accelerated objective gap and gradient norm convergence for gradient descent via long steps. *arXiv preprint arXiv:2403.14045*, 2024.
- [28] Benjamin Grimmer, Kevin Shu, and Alex L. Wang. Composing optimized stepsize schedules for gradient descent. *arXiv preprint arXiv:2410.16249*, 2024.
- [29] Shuvomoy Das Gupta, Bart P. G. Van Parys, and Ernest K. Ryu. Branch-and-bound performance estimation programming: a unified methodology for constructing optimal optimization methods. *Mathematical Programming*, 204(1–2):567–639, 2023.
- [30] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get M for free. In *International Conference on Learning Representations*, 2017.
- [31] Uijeong Jang, Shuvomoy Das Gupta, and Ernest K. Ryu. Computer-assisted design of accelerated composite optimization methods: OptISTA. *arXiv preprint arXiv:2305.15704*, 2024.

- [32] Jonathan Kelner, Annie Marsden, Vatsal Sharan, Aaron Sidford, Gregory Valiant, and Honglin Yuan. Big-step-little-step: Efficient gradient methods for objectives with multiple scales. In *Conference on Learning Theory*, volume 178 of *PMLR*, pages 2431–2540, 2022.
- [33] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.
- [34] Guy Kornowski and Ohad Shamir. Open problem: Anytime convergence rate of gradient descent. In *Conference on Learning Theory*, volume 247 of *PMLR*, pages 5335–5339, 2024.
- [35] Puya Latafat, Andreas Themelis, Lorenzo Stella, and Panagiotis Patrinos. Adaptive proximal algorithms for convex optimization under local Lipschitz continuity of the gradient. *Mathematical Programming*, 2024.
- [36] 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.
- [37] Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.
- [38] Yurii Nesterov. *Lectures on convex optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer, 2018.
- [39] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2006.
- [40] Samet Oymak. Provable super-convergence with a large cyclical learning rate. *IEEE Signal Processing Letters*, 28:1645–1649, 2021.
- [41] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.
- [42] Boris T. Polyak. *Introduction to optimization*. Translations Series in Mathematics and Engineering. Optimization Software, Inc., 1987.
- [43] Vinit Ranjan and Bartolomeo Stellato. Verification of first-order methods for parametric quadratic optimization. *arXiv preprint arXiv:2403.03331*, 2024.
- [44] Rajiv Sambharya. Personal communication, 2024.
- [45] Rajiv Sambharya and Bartolomeo Stellato. Learning algorithm hyperparameters for fast parametric convex optimization. *arXiv preprint arXiv:2411.15717*, 2024.
- [46] Leslie N. Smith. Cyclical learning rates for training neural networks. In *IEEE Winter Conference on Applications of Computer Vision*, pages 464–472, 2017.
- [47] Adrien B. Taylor, Julien M. Hendrickx, and François Glineur. Exact worst-case performance of first-order methods for composite convex optimization. *SIAM Journal on Optimization*, 27(3):1283–1313, 2017.
- [48] 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.
- [49] Adrien B. Taylor, Julien M. Hendrickx, and François Glineur. Exact worst-case convergence rates of the proximal gradient method for composite convex minimization. *Journal of Optimization Theory and Applications*, 178(2):455–476, 2018.
- [50] Marc Teboulle and Yakov Vaisbourd. An elementary approach to tight worst case complexity analysis of gradient based methods. *Mathematical Programming*, 201(1-2):63–96, 2023.

- [51] Kees van den Doel and Uri Ascher. The chaotic nature of faster gradient descent methods. *Journal of Scientific Computing*, 51(3):560–581, 2012.
- [52] Bofan Wang, Shiqian Ma, Junfeng Yang, and Danqing Zhou. Relaxed proximal point algorithm: Tight complexity bounds and acceleration without momentum. *arXiv preprint arXiv:2410.08890*, 2024.
- [53] Jingfeng Wu, Peter L. Bartlett, Matus Telgarsky, and Bin Yu. Large stepsize gradient descent for logistic loss: Non-monotonicity of the loss improves optimization efficiency. In *Conference on Learning Theory*, volume 247 of *PMLR*, pages 5019–5073, 2024.
- [54] 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.
- [55] Moslem Zamani and François Glineur. Exact convergence rate of the last iterate in subgradient methods. *arXiv preprint arXiv:2307.11134*, 2023.
- [56] Zehao Zhang and Rujun Jiang. Accelerated gradient descent by concatenation of stepsize schedules. *arXiv preprint arXiv:2410.12395*, 2024.
- [57] Zihan Zhang, Jason D. Lee, Simon S. Du, and Yuxin Chen. Anytime acceleration of gradient descent. *arXiv preprint arXiv:2411.17668*, 2024.

A Deferred details

A.1 Tightness of the silver convergence rate

Theorem 1.1 is asymptotically tight in that the exponent $\log_2 \rho$ in the rate $O(n^{-\log_2 \rho})$ is optimal for proximal GD with the silver stepsize schedule. Moreover, the hidden constant in the asymptotic rate is tight up to a modest constant. Indeed, there exists a problem instance such that

$$F(x_n) - F(x_*) = \frac{M}{4\rho^k - 4} \|x_0 - x_*\|^2, \quad (\text{A.1})$$

which matches the aforementioned exponent since $\rho^k = \rho^{\log_2(n+1)} \asymp \rho^{\log_2 n} = n^{\log_2 \rho}$. This bound (A.1) is achieved by the following 1-dimensional problem (normalized with $M = 1$ and $\|x_0 - x_*\|^2 = 1$ without loss of generality) from [20, Theorem 2.9] which is often used for lower bounds on proximal GD. Consider

$$\min_{x \in \mathbb{R}} F(x) = f(x) + h(x)$$

where $f(x) = ax$ is a linear function with slope $a = \frac{1}{2(\rho^k - 1)}$, and $h(x) = \iota\{x \geq 0\}$ is the indicator function for $x \geq 0$. Observe that $x_* = 0$, and that from initialization $x_0 = 1$, the final iterate of proximal GD is $x_n = 1 - a(\rho^k - 1)$, implying $F(x_n) - F(x_*) = a(1 - a(\rho^k - 1))$. Maximizing this rate over the slope $a \geq 0$ gives the aforementioned value of a and the rate (A.1).

We make three remarks about this “hard” problem instance. First, since h is the indicator of a convex set, this construction can also be viewed as a hard problem $\min_{x \geq 0} f(x)$ for projected GD. Second, by running semidefinite programs [21, 48] to numerically estimate the worst-case convergence rate of proximal GD with the silver stepsizes, it appears that (A.1) is the *exactly* optimal bound. Third, we compare the constants in this rate (A.1) for proximal GD with the silver stepsizes, to the exact rate $f(x_n) - f(x_*) \leq \frac{M}{4\rho^k - 2} \|x_0 - x_*\|^2$ of vanilla GD with the silver stepsizes [52]. The former is achieved by a linear function (equivalently, the ℓ_1 norm) constrained to the set $\{x \geq 0\}$, whereas the latter is achieved by a certain Huber loss function (i.e., the Moreau envelope of the ℓ_1 norm) over \mathbb{R} . This minor difference in problem instances results in the minor difference between the denominators in the rates: $4\rho^k - 4$ versus $4\rho^k - 2$. This discrepancy commonly arises between an algorithm and its proximal counterpart; see, for example, the discussion after [31, Theorem 1].

A.2 Nonnegativity of the multipliers

Here we complete the proof of Lemma 3.4 by verifying the nonnegativity of the $n - 1$ entries $\{\mu_{n,j}^{(k+1)}\}_{1 \leq j < n}$. To make the induction step clear, we isolate the claim into its own lemma.

Lemma A.1. *Let $\mu^{(k)}$ be as in Definition 3.3 and $n = 2^k - 1$. Then for all $k \in \mathbb{N}$ and $1 \leq j < n$,*

$$\bar{\mu}_{n,j}^{(k)} = (\rho^k - 1)(c_j^{(k)} - \pi_j^{(k)}).$$

As a corollary, for all $1 \leq j < n$,

$$\mu_{n,j}^{(k+1)} = \bar{\mu}_{n,j}^{(k+1),\text{rec}} + \bar{\mu}_{n,j}^{(k+1),\text{lr}} = \bar{\mu}_{n,j}^{(k)} + \left(1 - \frac{\rho^k}{\rho^{k-1} + 1}\right)(c_j^{(k)} - \pi_j^{(k)}) = \frac{\rho^{2k-1}}{\rho^{k-1} + 1}(c_j^{(k)} - \pi_j^{(k)}) \geq 0.$$

Proof. We prove by induction. For the base case, note that for $k = 1$ no such entry exists; and for $k = 2$, we have $\bar{\mu}_{3,1}^{(2)} = 0$ and $\bar{\mu}_{3,2}^{(2)} = (\rho - \frac{1}{\rho})(\rho^0 + 1) = (\rho^2 - 1)\frac{\rho^0 + 1}{\rho} = (\rho^2 - 1)(c_2^{(2)} - \pi_2^{(2)})$.

Assuming that the result holds for $k \geq 2$, by the recursive definition we have $\bar{\mu}_{2n+1,j}^{(k+1)} = 0$ for all $1 \leq j \leq n$ and $\bar{\mu}_{2n+1,n+1}^{(k+1)} = (\rho - \frac{1}{\rho^k})(\rho^{k-1} + 1) = (\rho^{k+1} - 1)\frac{\rho^{k-1} + 1}{\rho^k} = (\rho^{k+1} - 1)(c_{n+1}^{(k+1)} - \pi_{n+1}^{(k+1)})$. For $n+2 \leq j < 2n+1$,

$$\begin{aligned} \bar{\mu}_{2n+1,j}^{(k+1)} &= \rho^2 \bar{\mu}_{n,j-n-1}^{(k+1)} + (\rho + 1)c_{j-n-1}^{(k)} - \left(1 + \frac{1}{\rho^k}\right)\pi_{j-n-1}^{(k)} \\ &= \rho^2(\rho^k - 1)(c_{j-n-1}^{(k)} - \pi_{j-n-1}^{(k)}) + (\rho + 1)c_{j-n-1}^{(k)} - \left(1 + \frac{1}{\rho^k}\right)\pi_{j-n-1}^{(k)} \\ &= (\rho^{k+1} - 1)(\rho c_{j-n-1}^{(k)} - (\rho - \frac{1}{\rho^k})\pi_{j-n-1}^{(k)}) \\ &= (\rho^{k+1} - 1)(c_j^{(k+1)} - \pi_j^{(k+1)}). \end{aligned}$$

□

A.3 Laplacian structure

Here we complete the proof of Lemma 3.6 by showing that each row sum of $L^{(k)}$ is 0, for all $k \in \mathbb{N}$. We use induction. The base case $k = 1$ is trivial. Now assume that each row sum of $L^{(k)}$ is 0 (which implies that the j th row sum of $\bar{L}^{(k)}$ is $c_j^{(k)}$), and consider the row sums of $L^{(k+1)}$. For this, we make repeated use of the definitions of $\pi^{(k)}$ and $c^{(k)}$ and their respective helper lemmas (Lemmas 2.3 and 2.5) from §2.3.

For $1 \leq j \leq n$, the j th row sum of $L^{(k+1)}$ is given by

$$c_j^{(k)} + \left(\sum (c^{(k)} - \pi^{(k)})\right)(c_j^{(k)} - \pi_j^{(k)}) - \rho^k(c_j^{(k)} - \pi_j^{(k)}) - c_j^{(k+1)} = 0,$$

from $\sum (c^{(k)} - \pi^{(k)}) = \rho^k - 1$ and $c_j^{(k+1)} = \pi_j^{(k+1)} = \pi_j^{(k)}$.

The $(n+1)$ th row sum of $L^{(k+1)}$ is given by

$$-\rho^k(\rho^k - 1) + (\rho^{k-1} + 1)(\rho^{k+1} + 1) - \rho(\rho^k - 1) - \left(\sum (c^{(k+1)} - \pi^{(k+1)})\right)(c_{n+1}^{(k+1)} - \pi_{n+1}^{(k+1)}) - c_{n+1}^{(k+1)} = 0,$$

from $\sum (c^{(k+1)} - \pi^{(k+1)}) = \rho^{k+1} - 1$, $c_{n+1}^{(k+1)} = (1 + \frac{1}{\rho^k})(\rho^{k-1} + 1)$ and $\pi_{n+1}^{(k+1)} = \rho^{k-1} + 1$.

Finally, for $n+2 \leq j \leq 2n+1$, the j th row sum of $L^{(k+1)}$ is given by

$$\rho^2(c_{j-n-1}^{(k)} + (\rho^k - 1)(c_{j-n-1}^{(k)} - \pi_{j-n-1}^{(k)})) - \rho\pi_{j-n-1}^{(k)} - \left(\sum (c^{(k+1)} - \pi^{(k+1)})\right)(c_j^{(k+1)} - \pi_j^{(k+1)}) - c_j^{(k+1)} = 0,$$

from $c_j^{(k+1)} = \rho c_{j-n-1}^{(k)} - (\rho - 1 - \frac{1}{\rho^k})\pi_{j-n-1}^{(k)}$.

B Verification of the multi-step descent identity

Here we prove item (iii) of Theorem 3.1. The base case $k = 1$ is in Appendix B.1; we then prove the identity for larger k by induction. As described in the overview in §3.3, the desired identity has both a linear and a quadratic component, and it suffices to check these separately. We begin by introducing helpful bookkeeping notation in Appendix B.2, and then we check the linear component in Appendix B.3 and the quadratic components in Appendices B.4 and B.5. All proofs begin with a straightforward inspection over the coefficients of the relevant terms after expanding the definition of the co-coercivities; for the convenience of the reader, we explicitly write these expanded forms in Proposition C.1. The proofs then check that the constant number of coefficients match in a conceptually straightforward (albeit tedious) manner.

Throughout this section, for notational shorthand we denote $c = c^{(k)}$ and $S = S^{(k)}$. In the few cases when needed, we explicitly write $c^{(k+1)}$ and $S^{(k+1)}$ to distinguish them respectively from c and S .

B.1 Base case

Here we prove the base case $k = 1$ for the multi-step descent identity (3.1). For $k = 1$,

$$\lambda = \begin{matrix} & 0 & 1 \\ 0 & \begin{bmatrix} 0 & \rho \\ 1 & 0 \end{bmatrix} \\ * & \begin{bmatrix} \rho - 1 & \rho \end{bmatrix} \end{matrix}, \quad \mu = \begin{matrix} & 1 \\ 1 & \begin{bmatrix} 0 \\ 2\rho - 1 \end{bmatrix} \\ * & \end{matrix}, \quad S = \begin{bmatrix} \frac{1}{\sqrt{2}} & 1 & -1 \\ -1 & 2(\rho - 1) & -2(\rho - 1) \\ 1 & -2(\rho - 1) & 2(\rho - 1) \end{bmatrix},$$

$u = x_0 - x_* - (\rho - 1)g_0 - \rho g_1 - 2(\rho - 1)s_1 - s_*$, and $V = [x_0 - x_* \mid s_1 \mid s_*]$. Note that $x_1 = x_0 - (\rho - 1)(g_0 + s_1)$ and $g_* = -s_*$ by the definition of proximal GD. By plugging in these values and expanding the definition of

the co-coercivities (Definition 2.1), the multi-step descent identity (3.1) for $k = 1$ amounts to the identity

$$\begin{aligned}
& \rho(f_0 - f_1 - \langle g_1, x_0 - x_1 \rangle - \frac{1}{2}\|g_0 - g_1\|^2) + (f_1 - f_0 - \langle g_0, x_1 - x_0 \rangle - \frac{1}{2}\|g_1 - g_0\|^2) \\
& + (\rho - 1)(f_* - f_0 - \langle g_0, x_* - x_0 \rangle - \frac{1}{2}\|g_* - g_0\|^2) + \rho(f_* - f_1 - \langle g_1, x_* - x_1 \rangle - \frac{1}{2}\|g_* - g_1\|^2) \\
& + (2\rho - 1)(h_* - h_1 - \langle s_1, x_* - x_1 \rangle) \\
& = -(2\rho - 1)(f_1 + h_1 - f_* - h_*) + \frac{\rho}{2\sqrt{2}}\|x_0 - x_*\|^2 \\
& - \frac{1}{2}\|x_0 - x_* - (\rho - 1)g_0 - \rho g_1 - 2(\rho - 1)s_1 - s_*\|^2 - \frac{1}{4\sqrt{2}}\|x_0 - x_*\|^2 - \frac{1}{4\sqrt{2}}\|x_0 - x_* - 2(\rho - 1)(s_1 - s_*)\|^2.
\end{aligned}$$

It is straightforward to check this identity by matching coefficients. For example, on the left hand side, the coefficient of f_1 is $-\rho + 1 - \rho = -(2\rho - 1)$ and the coefficient of $\langle g_0, s_1 \rangle$ is $(\rho - 1) - (\rho - 1)(2\rho - 1) = -2(\rho - 1)^2$, respectively matching the corresponding coefficients on the right hand side.

B.2 Bookkeeping

We decompose terms in the multi-step descent identity based on the recursive definition of the multipliers. For the convenience of the reader, expanded expressions for these terms are provided in Proposition C.1.

Definition B.1 (Bookkeeping decomposition). *For $\lambda^{(k+1)}$ as in Definition 3.2 and $\mu^{(k+1)}$ as in Definition 3.3, let*

$$\begin{aligned}
R & := \sum_{i,j} \bar{\lambda}_{i,j}^{(k+1),\text{rec}} Q_{ij}^f + \sum_{i,j} \bar{\mu}_{i,j}^{(k+1),\text{rec}} Q_{ij}^h, \\
T^f & := \sum_{i,j} \lambda_{i,j}^{(k+1)} Q_{ij}^f, \\
T^h & := \sum_{i,j} \mu_{i,j}^{(k+1)} Q_{ij}^h, \\
C^f & := \sum_{i,j} (\bar{\lambda}_{i,j}^{(k+1),\text{sp}} + \bar{\lambda}_{i,j}^{(k+1),\text{lr}}) Q_{ij}^f, \\
C^h & := \sum_{i,j} (\bar{\mu}_{i,j}^{(k+1),\text{sp}} + \bar{\mu}_{i,j}^{(k+1),\text{lr}}) Q_{ij}^h.
\end{aligned} \tag{B.1}$$

The induction step from k to $k+1$ that we prove can be formally stated as follows. The rest of the section is devoted to proving this identity, by certifying that the coefficients on both sides are identical. Combined with the base case (Appendix B.1), this directly proves the multi-step identity (item (iii) of Theorem 3.1).

Theorem B.2. *Let R, T^f, T^h, C^f, C^h be as in (B.1), and assume that (3.1) holds for k . Then*

$$R + T^f + T^h + C^f + C^h = (2\rho^{k+1} - 1)(F_* - F_{2n+1}) + \frac{\rho}{2\sqrt{2}}\|x_0 - x_*\|^2 - \frac{1}{2}(\|\hat{u}\|^2 + \text{Tr}(\widehat{V}S^{(k+1)}\widehat{V}^T)) \tag{B.2}$$

where $\hat{u} := x_0 - x_* - \sum_{i=0}^{2n} \alpha_i g_i - \rho^{k+1} g_{2n+1} - \sum_{j=1}^{2n+1} c_j^{(k+1)} s_j - s_*$ and $\widehat{V} := [x_0 - x_* \mid s_1 \mid \dots \mid s_{2n+1} \mid s_*]$.

It is convenient to adopt the notation of [27] to write $[[\cdot]]_f$ to denote terms corresponding to $\{f_i\}$ in a given expression. For example, $[[Q_{01}^f]]_f = f_0 - f_1$; $[[\cdot]]_h$ is similarly defined.

These definitions readily extend to the quadratic form. Due to the composite nature of our problem, there are multiple types of vectors we need to consider. We categorize them into four types: $\{g, s, i, o\}$,

where each denotes⁴

$$\begin{aligned}
g &: g_0, g_1, \dots, g_{2n+1} \text{ (gradients } \underline{g}_i \text{ corresponding to } f), \\
s &: s_1, s_2, \dots, s_{2n+1} \text{ (gradients } \underline{s}_j \text{ corresponding to } h), \\
i &: x_0 - x_* \text{ (initial distance)}, \\
o &: s_* = -g_* \text{ (gradient at optimality)}.
\end{aligned}$$

For $y, z \in \{g, s, i, o\}$ we use $\llbracket \cdot \rrbracket_{y,z} = \llbracket \cdot \rrbracket_{z,y}$ to denote terms corresponding to the inner products between vectors in type y and z . For example, $\llbracket Q_{01}^f \rrbracket_{g,s} = -\sqrt{2} \langle g_1, s_1 \rangle$. It is clear that $\llbracket \cdot \rrbracket$ (with any subscript) is linear with respect to its input.

For each coefficient of interest, it is easy to see that only certain summands contribute. This observation is formalized as follows and is used throughout the rest of the proof when evaluating coefficients.

Observation B.3. *The following values are 0:*

$$\llbracket T^h + C^h \rrbracket_f, \llbracket T^f + C^f \rrbracket_h, \llbracket T^h + C^h \rrbracket_{g,g}, \llbracket T^f + C^f \rrbracket_{s,s}, \llbracket T^h \rrbracket_{i,g}, \llbracket T^f \rrbracket_{i,s}, \llbracket C^f + C^h \rrbracket_{i,y}, \llbracket T^h + C^f + C^h \rrbracket_{o,y},$$

where $y \in \{g, s, i, o\}$.

B.3 Linear form

Here we verify the linear form in the multi-step descent identity (B.2).

Proposition B.4 (Linear form). *Consider the setting of Theorem B.2. Then*

$$\begin{aligned}
\llbracket R + T^f + C^f \rrbracket_f &= (2\rho^{k+1} - 1)(f_* - f_{2n+1}), \\
\llbracket R + T^h + C^h \rrbracket_h &= (2\rho^{k+1} - 1)(h_* - h_{2n+1}).
\end{aligned}$$

Proof. The first equality was proved in [52, Theorem 5.2]. For the second equality, expanding each term following the definition yields (see Proposition C.1 for corresponding expressions)

$$\begin{aligned}
&\llbracket R + T^h + C^h \rrbracket_h \\
&= h_1 - h_n + \sum_{j=1}^n c_j (h_j - h_n) + \rho^2 (h_{n+2} - h_{2n+1}) + \sum_{j=n+2}^{2n+1} \rho^2 c_{j-n-1} (h_j - h_{2n+1}) + h_* - h_1 + \sum_{j=1}^n \alpha_{j-1} (h_* - h_j) \\
&\quad + (1 + \frac{1}{\rho^k})(\rho^{k-1} + 1)(h_* - h_{n+1}) + \sum_{j=n+2}^{2n+1} (\rho(c_{j-n-1} - \alpha_{j-1}) + (1 + \frac{1}{\rho^k})\alpha_{j-1})(h_* - h_j) \\
&\quad + \rho^k (h_n - h_{n+1}) + \rho^2 (h_{n+1} - h_{n+2}) + (\rho - \frac{1}{\rho^k})(\rho^{k-1} + 1)(h_{2n+1} - h_{n+1}) + \sum_{j=1}^n (c_j - \alpha_{j-1})(h_n - h_j) \\
&\quad + \sum_{j=1}^n \frac{\rho^k}{\rho^{k-1} + 1} (c_j - \alpha_{j-1})(h_{n+1} - h_n) + \sum_{j=n+2}^{2n+1} \rho \alpha_{j-1} (h_{n+1} - h_{2n+1}) + \sum_{j=n+2}^{2n+1} \frac{\rho^k}{\rho^{k-1} + 1} \alpha_{j-1} (h_n - h_{n+1}) \\
&\quad + \sum_{j=n+2}^{2n+1} ((\rho + 1)c_{j-n-1} + (\rho - 1 - \frac{1}{\rho^k})\alpha_{j-1})(h_{2n+1} - h_j).
\end{aligned}$$

The cancellation in the middle is due to $\sum_{j=1}^n c_j = 2(\rho^k - 1) = 2\sum_{j=1}^n \alpha_{j-1}$ (Lemmas 2.3 and 2.5). By

⁴While $x_0 - x_*$ (type i) or s_* (type o) are only single vectors and thus can be subsumed into other types, we treat them separately because the patterns they exhibit in the equations are different from (and simpler than) those for other types.

collecting the coefficients for each h_j , this is equal to

$$\begin{aligned}
& \sum_{j=1}^n (c_j - \alpha_{j-1} - (c_j - \alpha_{j-1}))h_j \\
& + \sum_{j=n+2}^{2n+1} (\rho^2 c_{j-n-1} - \rho(c_{j-n-1} - \alpha_{j-1}) - (1 + \frac{1}{\rho^k})\alpha_{j-1} - (\rho + 1)c_{j-n-1} - (\rho - 1 - \frac{1}{\rho^k})\alpha_{j-1})h_j \\
& + h_1 - h_1 + (-1 - 2(\rho^k - 1) + \rho^k + (\rho^k - 1))h_n \\
& + (-1 + \frac{1}{\rho^k})(\rho^{k-1} + 1) - \rho^k + \rho^2 - (\rho - \frac{1}{\rho^k})(\rho^{k-1} + 1) + \rho(\rho^k - 1))h_{n+1} + (\rho^2 - \rho^2)h_{n+2} \\
& + (-\rho^2 - 2\rho^2(\rho^k - 1) + (\rho - \frac{1}{\rho^k})(\rho^{k-1} + 1) - \rho(\rho^k - 1) + 2(\rho + 1)(\rho^k - 1) + (\rho - 1 - \frac{1}{\rho^k})(\rho^k - 1))h_{2n+1} \\
& + (1 + 2(\rho^{k+1} - 1))h_* \\
& = (2\rho^{k+1} - 1)(h_* - h_{2n+1}),
\end{aligned}$$

where the coefficient of h_* is calculated using $\sum_{j=1}^n \alpha_{j-1} + \sum_{j=n+2}^{2n+1} (\rho(c_{j-n-1} - \alpha_{j-1}) + (1 + \frac{1}{\rho^k})\alpha_{j-1}) = \sum c^{(k+1)} = 2(\rho^{k+1} - 1)$. \square

B.4 Quadratic form: setup

As briefly mentioned in the overview in §3.3, instead of checking the $\Theta(n^2)$ coefficients corresponding to all inner products between the $\Theta(n)$ variables $\{g_i\}$, $\{s_i\}$, and $x_0 - x_*$, we simplify the analysis by first observing that the quadratic form expressions in (B.1) can be expressed succinctly in a constant number of modified variables (obtained by taking linear combinations of the original variables). The upshot is that then we need to only compute and verify a constant number of coefficients. These modified variables are as follows.

Definition B.5 (Simple vectors for the quadratic form). *Define*

$$\begin{aligned}
\gamma &= [\gamma_1 \mid \gamma_2 \mid \gamma_3 \mid \gamma_4] \\
&:= \left[\sum_{i=0}^{n-1} \alpha_i g_i \mid g_n \mid \sum_{i=n+1}^{2n} \alpha_i g_i \mid g_{2n+1} \right], \\
\sigma &= [\sigma_1 \mid \sigma_2 \mid \sigma_3 \mid \sigma_4 \mid \sigma_5] \\
&:= \left[\sum_{j=1}^n \alpha_{j-1} s_j \mid \sum_{j=1}^n c_j s_j \mid s_{n+1} \mid \sum_{j=n+2}^{2n+1} \alpha_{j-1} s_j \mid \sum_{j=n+2}^{2n+1} c_{j-n-1} s_j \right].
\end{aligned}$$

From the recursive properties of $\pi^{(k)}$ and $c^{(k)}$ (Lemma 2.3 and Definition 2.4), one can observe that for

$$\begin{aligned}
w_g &:= [1, \rho^{k-1} + 1, 1, \rho^{k+1}], \\
w_s &:= [1, 0, (1 + \frac{1}{\rho^k})(\rho^{k-1} + 1), -(\rho - 1 - \frac{1}{\rho^k}), \rho],
\end{aligned}$$

we have

$$\begin{aligned}
\gamma w_g &= \sum_{i=0}^{2n} \alpha_i g_i + \rho^{k+1} g_{2n+1}, \\
\sigma w_s &= \sum_{j=1}^{2n+1} c_j^{(k+1)} s_j.
\end{aligned}$$

These vectors naturally arise from the quadratic forms in (3.1) and (B.2), which can be concisely represented by them; in particular,

$$\begin{aligned}
\frac{1}{2} \|u\|^2 &= \frac{1}{2} \|x_0 - x_* - \gamma_1 - \rho^k \gamma_2 - \sigma_2 - s_*\|^2, \\
\frac{1}{2} \|\hat{u}\|^2 &= \frac{1}{2} \|x_0 - x_* - \gamma w_g - \sigma w_s - s_*\|^2,
\end{aligned}$$

and the terms $\frac{1}{2} \text{Tr}(VSV^T)$ and $\frac{1}{2} \text{Tr}(\widehat{V}S^{(k+1)}\widehat{V}^T)$ together can be expressed similarly by σ modulo simple operations from recursion.

While some expressions in (B.1) can be represented with γ or σ in a straightforward way, in some cases the terms should be properly combined with others, for which we provide the details. First, we identify particular terms that are sums of inner products. These are categorized as A_1, A_2, A_3 in Proposition C.1, and with a slight abuse of notation we use A_i for the sum of the respective entries as well. The following equalities can be verified by inspection.

Observation B.6. For A_1, A_2, A_3 in Proposition C.1 and γ, σ as in Definition B.5,

$$\begin{aligned} \llbracket A_1 \rrbracket_{g,s} &= \langle \gamma_1, \sigma_2 - \sigma_1 \rangle, \\ \llbracket A_2 \rrbracket_{g,s} &= \rho \langle \gamma_3, 2\sigma_1 + (\rho^{k-1} + 1)\sigma_3 + \sigma_4 \rangle, \\ \llbracket A_3 \rrbracket_{g,s} &= \left\langle \gamma_1 + (\rho^{k-1} + 1)\gamma_2 + \gamma_3, (\rho - 1 - \frac{1}{\rho^k})\sigma_4 + (\rho + 1)\sigma_5 \right\rangle, \end{aligned}$$

and

$$\begin{aligned} \llbracket A_1 \rrbracket_{s,s} &= \langle \sigma_1, \sigma_2 - \sigma_1 \rangle, \\ \llbracket A_2 \rrbracket_{s,s} &= 0, \\ \llbracket A_3 \rrbracket_{s,s} &= \left\langle \sigma_1 + (\rho^{k-1} + 1)\sigma_3 + \sigma_4, (\rho - 1 - \frac{1}{\rho^k})\sigma_4 + (\rho + 1)\sigma_5 \right\rangle. \end{aligned}$$

Next, we identify other terms (B_1, B_2, B_3 in Proposition C.1, with similar abuse of notation) which are related to specific gradient vectors; namely, $g_n = \gamma_2, g_{2n+1} = \gamma_4$, and s_{n+2} . These can also be verified by inspection and by using the Pell recurrence $\rho^2 = 2\rho + 1$ for the silver ratio ρ .

Observation B.7. For B_1, B_2, B_3 in Proposition C.1 and γ, σ as in Definition B.5,

$$\begin{aligned} \llbracket B_1 \rrbracket_{g,s} &= \rho^k \langle \gamma_2, \sigma_1 + (\rho^{k-1} + 1)\sigma_3 + \sigma_4 \rangle, \\ \llbracket B_2 \rrbracket_{g,s} &= \rho^2 \langle \gamma_1 + (\rho^{k-1} + 1)\gamma_2, s_{n+2} \rangle, \\ \llbracket B_3 \rrbracket_{g,s} &= \rho^k (\rho + 1) \langle \gamma_4, \sigma_1 + (\rho^{k-1} + 1)\sigma_3 + \sigma_4 \rangle, \end{aligned}$$

and

$$\begin{aligned} \llbracket B_1 \rrbracket_{s,s} &= 0, \\ \llbracket B_2 \rrbracket_{s,s} &= \rho^2 \langle \sigma_1 + (\rho^{k-1} + 1)\sigma_3, s_{n+2} \rangle, \\ \llbracket B_3 \rrbracket_{s,s} &= 0. \end{aligned}$$

B.5 Quadratic form: verification

After the preprocessing steps in Appendix B.4, we are ready to prove that the coefficients of the quadratic form match in (B.2). The proof is divided into multiple parts based on the combination of $y, z \in \{g, s, i, o\}$.

First, we verify the terms involving $\langle g_i, g_j \rangle$ in the multi-step descent identity. This is identical to (and thus follows from) the corresponding analysis of vanilla GD.

Proposition B.8 (Quadratic form in (g, g) ; [52, Theorem 5.2]). Consider the setting of Theorem B.2 and let γ, w_g be as in Definition B.5. Then

$$\llbracket R + T^f + C^f \rrbracket_{g,g} = -\frac{1}{2} \|\gamma w_s\|^2.$$

The next two propositions respectively verify the terms involving $\langle g_i, s_j \rangle$ and $\langle s_i, s_j \rangle$.

Proposition B.9 (Quadratic form in (g, s)). Consider the setting of Theorem B.2 and let γ, w_g, σ, w_s be as in Definition B.5. Then

$$\llbracket R + T^f + T^h + C^f + C^h \rrbracket_{g,s} = -\langle \gamma w_g, \sigma w_s \rangle.$$

Proof. Using the representation with respect to γ and σ given from Observation B.6 and Observation B.7,

$$\begin{aligned}
\llbracket R + T^f + T^h + C^f + C^h \rrbracket_{g,s} &= \llbracket A_1 + A_2 + A_3 + B_1 + B_2 + B_3 \rrbracket_{g,s} + \llbracket (\text{remaining terms in (B.1)}) \rrbracket_{g,s} \\
&= \langle \gamma_1, \sigma_2 - \sigma_1 \rangle + \rho \langle \gamma_3, 2\sigma_1 + (\rho^{k-1} + 1)\sigma_3 + \sigma_4 \rangle \\
&\quad + \left\langle \gamma_1 + (\rho^{k-1} + 1)\gamma_2 + \gamma_3, (\rho - 1 - \frac{1}{\rho^k})\sigma_4 + (\rho + 1)\sigma_5 \right\rangle \\
&\quad + \rho^k \langle \gamma_2, \sigma_1 + (\rho^{k-1} + 1)\sigma_3 + \sigma_4 \rangle + \rho^2 \langle \gamma_1 + (\rho^{k-1} + 1)\gamma_2, s_{n+2} \rangle \\
&\quad + \rho^k(\rho + 1) \langle \gamma_4, \sigma_1 + (\rho^{k-1} + 1)\sigma_3 + \sigma_4 \rangle \\
&\quad - \langle \gamma_1 + \rho^k \gamma_2, \sigma_2 \rangle - \rho^2 \langle \gamma_1 + (\rho^{k-1} + 1)\gamma_2, \sigma_5 \rangle - \rho^2 \langle \gamma_3 + \rho^k \gamma_4, \sigma_1 + (\rho^{k-1} + 1)\sigma_3 \rangle \\
&\quad - \rho^2 \langle \gamma_3 + \rho^k \gamma_4, \gamma_5 \rangle - \rho^2 \langle \gamma_1 + (\rho^{k-1} + 1)\gamma_2, s_{n+2} \rangle - (\rho^{k-1} + 1) \langle \gamma_2, \sigma_1 \rangle \\
&\quad - (1 + \frac{1}{\rho^k})(\rho^{k-1} + 1) \langle \gamma_1 + (\rho^{k-1} + 1)\gamma_2, \sigma_3 \rangle - \rho \langle \gamma_4, (\rho^{k-1} + 1)\sigma_3 + \sigma_4 \rangle \\
&\quad - \rho^k(\rho^{k-1} + 1) \langle \gamma_2, \sigma_3 \rangle + (\rho - \frac{1}{\rho^k})(\rho^{k-1} + 1) \langle \gamma_3, \sigma_3 \rangle + \rho^k \langle \gamma_2, \sigma_2 - \sigma_1 \rangle \\
&\quad - \rho \langle \gamma_3, \sigma_4 \rangle - \rho^k \langle \gamma_2, \sigma_4 \rangle.
\end{aligned}$$

This is equal to $-\langle \gamma w_g, \sigma w_s \rangle$. This can be checked by hand; for brevity, we provide a simple Mathematica script that rigorously verifies these identities at the URL [1]. \square

Proposition B.10 (Quadratic form in (s, s)). *Consider the setting of Theorem B.2 and let σ, w_s be as in Definition B.5. Then*

$$\llbracket R + T^h + C^h \rrbracket_{s,s} = -\frac{1}{2} \|\sigma w_s\|^2 - \left\llbracket \frac{1}{2} \text{Tr}(\widehat{V} S^{(k+1)} \widehat{V}^T) \right\llbracket_{s,s},$$

where $S^{(k+1)}$ is as in Definition 3.5 and $\widehat{V} = [x_0 - x_* \mid s_1 \mid \dots \mid s_{2n+1} \mid s_*]$.

Proof of Proposition B.10. With the expressions from Observations B.6 and B.7,

$$\begin{aligned}
\left\llbracket R + T^h + C^h + \frac{1}{2} \text{Tr}(\widehat{V} S^{(k+1)} \widehat{V}^T) \right\llbracket_{s,s} &= \llbracket A_1 + A_2 + A_3 + B_1 + B_2 + B_3 \rrbracket_{s,s} \\
&\quad + \llbracket (\text{remaining terms in (B.1)}) \rrbracket_{s,s} + \left\llbracket \frac{1}{2} \text{Tr}(\widehat{V} S^{(k+1)} \widehat{V}^T) \right\llbracket_{s,s} \\
&= \langle \sigma_1, \sigma_2 - \sigma_1 \rangle + \left\langle \sigma_1 + (\rho^{k-1} + 1)\sigma_3 + \sigma_4, (\rho - 1 - \frac{1}{\rho^k})\sigma_4 + (\rho + 1)\sigma_5 \right\rangle \\
&\quad + \rho^2 \langle \sigma_1 + (\rho^{k-1} + 1)\sigma_3, s_{n+2} \rangle \\
&\quad - \frac{1}{2} \|\sigma_2\|^2 - \left\llbracket \frac{1}{2} \text{Tr}(V S V^T) \right\llbracket_{s,s} - \rho^2 \langle \sigma_1 + (\rho^{k-1} + 1)\sigma_3, \sigma_5 \rangle \\
&\quad - \frac{\rho^2}{2} \|\sigma_5\|^2 - \left\llbracket \frac{\rho^2}{2} \text{Tr}(\widetilde{V} S \widetilde{V}^T) \right\llbracket_{s,s} \\
&\quad - (1 + \frac{1}{\rho^k})(\rho^{k-1} + 1) \langle \sigma_3, \sigma_1 + (1 + \rho^{k-1})\sigma_3 \rangle - \rho^k(\rho^{k-1} + 1) \|\sigma_3\|^2 \\
&\quad + (\rho - \frac{1}{\rho^k})(\rho^{k-1} + 1) \langle \sigma_3, \sigma_4 \rangle + \rho^k \langle \sigma_3, \sigma_2 - \sigma_1 \rangle - \rho \|\sigma_4\|^2 - \rho^k \langle \sigma_3, \sigma_4 \rangle \\
&\quad + \left\llbracket \frac{1}{2} \text{Tr}(\widehat{V} S^{(k+1)} \widehat{V}^T) \right\llbracket_{s,s},
\end{aligned}$$

where $V = [x_0 - x_* \mid s_1 \mid \dots \mid s_n \mid s_*]$ and $\widetilde{V} = [x_{n+1} - x_* \mid s_{n+2} \mid \dots \mid s_{2n+1} \mid s_*]$. Collecting the terms

that involve vectors other than those from σ , by the recursive construction of $S^{(k+1)}$ (Definition 3.5) we have

$$\begin{aligned} & \rho^2 \langle \sigma_1 + (\rho^{k-1} + 1)\sigma_3, s_{n+2} \rangle - \left[\frac{1}{2} \text{Tr}(VSV^T) \right]_{s,s} - \left[\frac{\rho^2}{2} \text{Tr}(\tilde{V}S\tilde{V}^T) \right]_{s,s} + \left[\frac{1}{2} \text{Tr}(\widehat{V}S^{(k+1)}\widehat{V}^T) \right]_{s,s} \\ &= \rho^2 \langle \sigma_1 + (\rho^{k-1} + 1)\sigma_3, s_{n+2} \rangle + \frac{1}{2} \|\sigma_2 - \sigma_1\|^2 + \frac{\rho^2}{2} \|\sigma_5 - \sigma_4\|^2 - \rho^2 \langle \sigma_1 + (\rho^{k-1} + 1)\sigma_3, s_{n+2} \rangle \\ & \quad + \frac{1}{2} (\rho^{k-1} + 1)(\rho^{k+1} + 1) \|\sigma_3\|^2 - \rho^k \langle \sigma_3, \sigma_2 - \sigma_1 \rangle - \rho \langle \sigma_3, \sigma_4 \rangle - \frac{1}{2} \left\| \frac{\rho^{k-1} + 1}{\rho^k} \sigma_3 - \left(\rho - \frac{1}{\rho^k} \right) \sigma_4 + \rho \sigma_5 \right\|^2, \end{aligned}$$

where the last term is from $c^{(k+1)} - \pi^{(k+1)} = \underbrace{[0, \dots, 0]}_n, \frac{\rho^{k-1} + 1}{\rho^k}, \rho c^{(k)} - \left(\rho - \frac{1}{\rho^k} \right) \pi^{(k)}$. Thus, it suffices to show that

$$\begin{aligned} & \langle \sigma_1, \sigma_2 - \sigma_1 \rangle + \left\langle \sigma_1 + (\rho^{k-1} + 1)\sigma_3 + \sigma_4, \left(\rho - 1 - \frac{1}{\rho^k} \right) \sigma_4 + (\rho + 1)\sigma_5 \right\rangle - \frac{1}{2} \|\sigma_2\|^2 \\ & - \rho^2 \langle \sigma_1 + (\rho^{k-1} + 1)\sigma_3, \sigma_5 \rangle - \frac{\rho^2}{2} \|\sigma_5\|^2 - \left(1 + \frac{1}{\rho^k} \right) (\rho^{k-1} + 1) \langle \sigma_3, \sigma_1 + (\rho^{k-1} + 1)\sigma_3 \rangle - \rho^k (\rho^{k-1} + 1) \|\sigma_3\|^2 \\ & + \left(\rho - \frac{1}{\rho^k} \right) (\rho^{k-1} + 1) \langle \sigma_3, \sigma_4 \rangle + \rho^k \langle \sigma_3, \sigma_2 - \sigma_1 \rangle - \rho \|\sigma_4\|^2 - \rho^k \langle \sigma_3, \sigma_4 \rangle \\ & + \frac{1}{2} \|\sigma_2 - \sigma_1\|^2 + \frac{\rho^2}{2} \|\sigma_5 - \sigma_4\|^2 + \frac{1}{2} (\rho^{k-1} + 1)(\rho^{k+1} + 1) \|\sigma_3\|^2 \\ & - \rho^k \langle \sigma_3, \sigma_2 - \sigma_1 \rangle - \rho \langle \sigma_3, \sigma_4 \rangle - \frac{1}{2} \left\| \frac{\rho^{k-1} + 1}{\rho^k} \sigma_3 - \left(\rho - \frac{1}{\rho^k} \right) \sigma_4 + \rho \sigma_5 \right\|^2 \end{aligned}$$

is equal to $-\frac{1}{2} \|\sigma w_s\|^2$. This can be checked by hand; for brevity, we provide a simple Mathematica script that rigorously verifies these identities at the URL [1]. \square

Finally, we verify the terms involving $x_0 - x_*$ and s_* .

Proposition B.11 (Quadratic form involving $x_0 - x_*$). *Consider the setting of Theorem B.2 and let γ, w_g, σ, w_s be as in Definition B.5. Then*

$$\begin{aligned} \llbracket R + T^f \rrbracket_{i,g} &= \langle x_0 - x_*, \gamma w_g \rangle, \\ \llbracket R + T^h \rrbracket_{i,s} &= \langle x_0 - x_*, s_1 \rangle + \langle x_0 - x_*, \sigma w_s \rangle, \\ \llbracket R + T^f + T^h \rrbracket_{i,i} &= 0. \end{aligned}$$

Proof. The last equality $\llbracket R + T^f + T^h \rrbracket_{i,i} = 0$ is clear from inspection. For other cases,

$$\begin{aligned} \llbracket R + T^f \rrbracket_{i,g} &= \sum_{i=0}^{n-1} (-\alpha_i + \alpha_i + \alpha_i) \langle x_0 - x_*, g_i \rangle + \sum_{i=0}^{n-1} (-\rho^k + \rho^k + (\rho^{k-1} + 1)) \langle x_0 - x_*, g_n \rangle \\ & \quad + \sum_{i=n+1}^{2n} (-\rho^2 \alpha_i + \rho^2 \alpha_i + \alpha_i) \langle x_0 - x_*, g_i \rangle + (-\rho^{k+2} + \rho^{k+2} + \rho^{k+1}) \langle x_0 - x_*, g_{2n+1} \rangle \\ &= \langle x_0 - x_*, \gamma w_g \rangle, \\ \llbracket R + T^h \rrbracket_{i,s} &= \langle x_0 - x_*, s_1 \rangle + \sum_{j=1}^n (-c_j + c_j + \alpha_{j-1}) \langle x_0 - x_*, s_j \rangle + \left(1 + \frac{1}{\rho^k} \right) (\rho^{k-1} + 1) \langle x_0 - x_*, s_{n+1} \rangle \\ & \quad - \rho^2 \langle x_0 - x_*, s_{n+2} \rangle + \rho^2 \langle x_0 - x_*, s_{n+2} \rangle \\ & \quad + \sum_{j=n+2}^{2n+1} (-\rho^2 c_{j-n-1} + \rho^2 c_{j-n-1} + \rho(c_{j-n-1} - \alpha_{j-1})) + \left(1 + \frac{1}{\rho^k} \right) \alpha_{j-1} \langle x_0 - x_*, s_j \rangle \\ &= \langle x_0 - x_*, s_1 \rangle + \langle x_0 - x_*, \sigma w_s \rangle. \end{aligned}$$

\square

Proposition B.12 (Quadratic form involving s_*). *Consider the setting of Theorem B.2 and let γ, w_g , be as in Definition B.5. Then*

$$\begin{aligned} \llbracket R + T^f \rrbracket_{o,g} &= -\langle s_*, \gamma w_g \rangle, \\ \llbracket R + T^f \rrbracket_{o,s} &= 0, \\ \llbracket R + T^f \rrbracket_{o,i} &= 0, \\ \llbracket R + T^f \rrbracket_{o,o} &= -\frac{1}{2}(2\rho^{k+1} - 1)\|s_*\|^2. \end{aligned}$$

Proof. The second and third equalities $\llbracket R + T^f \rrbracket_{o,s} = \llbracket R + T^f \rrbracket_{o,i} = 0$ are clear from inspection. For other cases,

$$\begin{aligned} \llbracket R + T^f \rrbracket_{o,g} &= \sum_{i=0}^{n-1} (\alpha_i - \alpha_i - \alpha_i) \langle s_*, g_i \rangle + (\rho^k - \rho^k - (\rho^{k-1} + 1)) \langle s_*, g_n \rangle \\ &\quad + \sum_{i=n+1}^{2n} (\rho^2 \alpha_i - \rho^2 \alpha_i - \alpha_i) \langle s_*, g_i \rangle + (\rho^{k+2} - \rho^{k+2} - \rho^{k+1}) \langle s_*, g_{2n+1} \rangle \\ &= -\langle \gamma w_g, s_* \rangle, \\ \llbracket R + T^f \rrbracket_{o,o} &= \left(\sum_{i=0}^{n-1} \frac{\alpha_i}{2} + \frac{\rho^k}{2} - \frac{1}{2} + \sum_{i=n+1}^{2n} \frac{\rho^2 \alpha_i}{2} + \frac{\rho^{k+2}}{2} - \frac{\rho^2}{2} - \sum_{i=0}^{n-1} \frac{\alpha_i}{2} - \frac{\rho^{k-1} + 1}{2} - \sum_{i=0}^{n-1} \frac{\alpha_i}{2} - \frac{\rho^{k+1}}{2} \right) \|s_*\|^2 \\ &= -\frac{1}{2}(2\rho^{k+1} - 1)\|s_*\|^2. \end{aligned}$$

□

Combining these results proves Theorem B.2.

Proof of Theorem B.2. The propositions in Appendices B.3 and B.5 show that the left and right hand side of (B.2) match. □

C Expanded expression for the decomposition

For the reader's convenience, here we provide a fully expanded expression for the terms in (B.1). This follows by expanding the definition of co-coercivities (Definition 2.1) and using the induction hypothesis (3.1).

Proposition C.1 (Expanded expressions for (B.1)). *Let R, T^f, T^h, C^f, C^h be as in (B.1), and u, V, S be as in Theorem 3.1. Also, assume that (3.1) is true for k . Then with*

$$\begin{aligned} \tilde{u} &:= x_{n+1} - x_* - \sum_{i=n+1}^{2n} \alpha_i g_i - \rho^k g_{2n+1} - \sum_{j=n+2}^{2n+1} c_{j-n-1} s_j - s_*, \\ \tilde{V} &:= [x_{n+1} - x_* \mid s_{n+2} \mid \dots \mid s_{2n+1} \mid s_*], \end{aligned}$$

the explicit expressions for the terms in the decomposition are given as follows:

$$\begin{aligned}
R &= \underbrace{\sum_{i=0}^{n-1} \alpha_i (f_i - f_n + \langle g_i, x_* - x_i \rangle + \frac{1}{2} \|g_* - g_i\|^2)}_{A_1} + \underbrace{\rho^k \langle g_n, x_* - x_n \rangle}_{B_1} + \frac{\rho^k}{2} \|g_* - g_n\|^2 \\
&+ \underbrace{h_1 - h_n + \langle s_1, x_* - x_1 \rangle}_{B_2} + \underbrace{\sum_{j=1}^n c_j (h_j - h_n + \langle s_j, x_* - x_j \rangle)}_{A_1} + \frac{1}{2} (1 + \frac{1}{\sqrt{2}}) \|x_0 - x_*\|^2 - \frac{1}{2} (\|u\|^2 + \text{Tr}(VSV^T)) \\
&+ \underbrace{\sum_{i=n+1}^{2n} \rho^2 \alpha_i (f_i - f_{2n+1} + \langle g_i, x_* - x_i \rangle + \frac{1}{2} \|g_* - g_i\|^2)}_{A_2} + \underbrace{\rho^{k+2} \langle g_{2n+1}, x_* - x_{2n+1} \rangle}_{B_3} + \frac{\rho^{k+2}}{2} \|g_* - g_{2n+1}\|^2 \\
&+ \underbrace{\rho^2 (h_{n+2} - h_{2n+1} + \langle s_{n+2}, x_* - x_{n+2} \rangle)}_{B_2} + \underbrace{\sum_{j=n+2}^{2n+1} \rho^2 c_{j-n-1} (h_j - h_{2n+1} + \langle s_j, x_* - x_j \rangle)}_{A_3} \\
&+ \frac{\rho^2}{2} (1 + \frac{1}{\sqrt{2}}) \|x_{n+1} - x_*\|^2 - \frac{\rho^2}{2} (\|\tilde{u}\|^2 + \text{Tr}(\tilde{V}S\tilde{V}^T)), \\
T^f &= \underbrace{\sum_{i=0}^{n-1} \alpha_i (f_* - f_i - \langle g_i, x_* - x_i \rangle - \frac{1}{2} \|g_* - g_i\|^2)}_{A_1} + (\rho^{k-1} + 1) (f_* - f_n - \langle g_n, x_* - x_n \rangle - \frac{1}{2} \|g_* - g_n\|^2) \\
&+ \underbrace{\sum_{i=n+1}^{2n} \alpha_i (f_* - f_i - \langle g_i, x_* - x_i \rangle - \frac{1}{2} \|g_* - g_i\|^2)}_{A_2} + \underbrace{\rho^{k+1} (f_* - f_{2n+1} - \langle g_{2n+1}, x_* - x_{2n+1} \rangle - \frac{1}{2} \|g_* - g_{2n+1}\|^2)}_{B_3}, \\
T^h &= \underbrace{h_* - h_1 - \langle s_1, x_* - x_1 \rangle}_{B_1} + \sum_{j=1}^n \alpha_{j-1} (h_* - h_j - \langle s_j, x_* - x_j \rangle) + (1 + \frac{1}{\rho^k}) (\rho^{k-1} + 1) (h_* - h_{n+1} - \langle s_{n+1}, x_* - x_{n+1} \rangle) \\
&+ \underbrace{\sum_{j=n+2}^{2n+1} (\rho(c_{j-n-1} - \alpha_{j-1}) + (1 + \frac{1}{\rho^k}) \alpha_{j-1}) (h_* - h_j - \langle s_j, x_* - x_j \rangle)}_{A_3}, \\
C^f &= \underbrace{\sum_{i=n+1}^{2n} \rho \alpha_i (f_n - f_i - \langle g_i, x_n - x_i \rangle - \frac{1}{2} \|g_n - g_i\|^2)}_{A_2} + \rho (f_n - f_{2n+1} - \langle g_{2n+1}, x_n - x_{2n+1} \rangle - \frac{1}{2} \|g_n - g_{2n+1}\|^2) \\
&+ \underbrace{\sum_{i=n+1}^{2n} \rho \alpha_i (f_{2n+1} - f_i - \langle g_i, x_{2n+1} - x_i \rangle - \frac{1}{2} \|g_{2n+1} - g_i\|^2)}_{A_2} + \underbrace{\rho^k (f_{2n+1} - f_n - \langle g_n, x_{2n+1} - x_n \rangle - \frac{1}{2} \|g_n - g_{2n+1}\|^2)}_{B_1},
\end{aligned}$$

and

$$\begin{aligned}
C^h &= \rho^k(h_n - h_{n+1} - \langle s_{n+1}, x_n - x_{n+1} \rangle) + \underbrace{\rho^2(h_{n+1} - h_{n+2} - \langle s_{n+2}, x_{n+1} - x_{n+2} \rangle)}_{B_2} \\
&+ (\rho - \frac{1}{\rho^k})(\rho^{k-1} + 1)(h_{2n+1} - h_{n+1} - \langle s_{n+1}, x_{2n+1} - x_{n+1} \rangle) \\
&+ \underbrace{\sum_{j=1}^n (c_j - \alpha_{j-1})(h_n - h_j - \langle s_j, x_n - x_j \rangle)}_{A_1} + \sum_{j=1}^n \frac{\rho^k}{\rho^{k-1} + 1} (c_j - \alpha_{j-1})(h_{n+1} - h_n - \langle s_j, x_{n+1} - x_n \rangle) \\
&+ \sum_{j=n+2}^{2n+1} \rho \alpha_{j-1} (h_{n+1} - h_{2n+1} - \langle s_j, x_{n+1} - x_{2n+1} \rangle) + \sum_{j=n+2}^{2n+1} \frac{\rho^k}{\rho^{k-1} + 1} \alpha_{j-1} (h_n - h_{n+1} - \langle s_j, x_n - x_{n+1} \rangle) \\
&\underbrace{\sum_{j=n+2}^{2n+1} ((\rho + 1)c_{j-n-1} + (\rho - 1 - \frac{1}{\rho^k})\alpha_{j-1})(h_{2n+1} - h_j - \langle s_j, x_{2n+1} - x_j \rangle)}_{A_3}.
\end{aligned}$$

Proof. The equalities for T^f, T^h, C^f are obtained directly from expanding the co-coercivities. The expression for C^h can be obtained after combining terms sequentially; in particular,

$$\begin{aligned}
&\sum_{j=1}^n \bar{\mu}_{n,j}^{(k+1),\text{lr}} Q_{nj}^h + \sum_{j=1}^n \bar{\mu}_{n+1,j}^{(k+1),\text{lr}} Q_{(n+1)j}^h \\
&= \sum_{j=1}^n (1 - \frac{\rho^k}{\rho^{k-1} + 1})(c_j - \alpha_{j-1})(h_n - h_j - \langle s_j, x_n - x_j \rangle) + \sum_{j=1}^n \frac{\rho^k}{\rho^{k-1} + 1} (c_j - \alpha_{j-1})(h_{n+1} - h_j - \langle s_j, x_{n+1} - x_j \rangle) \\
&= \sum_{j=1}^n (c_j - \alpha_{j-1})(h_n - h_j - \langle s_j, x_n - x_j \rangle) + \sum_{j=1}^n \frac{\rho^k}{\rho^{k-1} + 1} (c_j - \alpha_{j-1})(h_{n+1} - h_n - \langle s_j, x_{n+1} - x_n \rangle), \\
&\sum_{j=n+2}^{2n+1} \bar{\mu}_{n,j}^{(k+1),\text{lr}} Q_{nj}^h + \sum_{j=n+2}^{2n+1} \bar{\mu}_{n+1,j}^{(k+1),\text{lr}} Q_{(n+1)j}^h \\
&= \sum_{j=n+2}^{2n+1} \frac{\rho^k}{\rho^{k-1} + 1} \alpha_{j-1} (h_n - h_j - \langle s_j, x_n - x_j \rangle) + \sum_{j=n+2}^{2n+1} \frac{\rho}{\rho^{k-1} + 1} \alpha_{j-1} (h_{n+1} - h_j - \langle s_j, x_{n+1} - x_j \rangle) \\
&= \underbrace{\sum_{j=n+2}^{2n+1} \rho \alpha_{j-1} (h_{n+1} - h_j - \langle s_j, x_{n+1} - x_j \rangle)}_{=: \Delta} + \sum_{j=n+2}^{2n+1} \frac{\rho^k}{\rho^{k-1} + 1} \alpha_{j-1} (h_n - h_{n+1} - \langle s_j, x_n - x_{n+1} \rangle),
\end{aligned}$$

and

$$\begin{aligned}
\Delta + \sum_{j=n+2}^{2n+1} \bar{\mu}_{2n+1,j}^{(k+1),\text{lr}} Q_{(2n+1)j}^h &= \Delta + \sum_{j=n+2}^{2n+1} ((\rho + 1)c_{j-n-1} - (1 + \frac{1}{\rho^k})\alpha_{j-1})(h_{2n+1} - h_j - \langle s_j, x_{2n+1} - x_j \rangle) \\
&= \sum_{j=n+2}^{2n+1} \rho \alpha_{j-1} (h_{n+1} - h_{2n+1} - \langle s_j, x_{n+1} - x_{2n+1} \rangle) \\
&\quad + \sum_{j=n+2}^{2n+1} ((\rho + 1)c_{j-n-1} + (\rho - 1 - \frac{1}{\rho^k})\alpha_{j-1})(h_{2n+1} - h_j - \langle s_j, x_{2n+1} - x_j \rangle).
\end{aligned}$$

The expression for R can be obtained directly from the definition of $\underline{\lambda}^{(k)}$ and (3.1), which is valid for the iterates $x_0, x_1, \dots, x_n, x_*$ of proximal GD with the silver stepsize schedule $\pi^{(k)}$ and for the iterates $x_{n+1}, x_{n+2}, \dots, x_{2n+1}, x_*$ of the same algorithm with initial point x_{n+1} , by the recursive construction of the silver stepsize schedule $\pi^{(k+1)} = [\pi^{(k)}, \rho^{k-1} + 1, \pi^{(k)}]$ (Lemma 2.3). \square