A Domain-Specific Harness for End-to-End Automation of Optimization Research

Heechang Kim
Seoul National University
`heechang.kim@snu.ac.kr`

Ernest Ryu
UCLA
`eryu@math.ucla.edu`

Shuvomoy Das Gupta
Rice University
`sd158@rice.edu`

Abstract

We present AutoOPT, a domain-specific harness for end-to-end automation of optimization research. AutoOPT organizes the discovery of optimal first-order methods into four stages: numerical design through the BnB-PEP methodology; symbolic discovery of the analytic description and a convergence proof through frontier large language models (LLMs); formal verification in the Lean 4 proof assistant; and human interpretation and write-up. We demonstrate the framework on two case studies, each of independent interest. The first, lemniscate acceleration, is a new accelerated gradient method for minimizing the gradient norm of a smooth convex function: after N gradient steps it reduces the squared gradient norm at the optimal $O(1/N^4)$ rate, with a constant governed by the lemniscate constant ϖ, a classical elliptic-integral constant. The second is the analytic description of ITEM-f, a method previously known only numerically: for L-smooth, μ-strongly convex minimization it contracts the function-value gap at an accelerated linear rate with a per-step factor $(1-\sqrt{\mu/L})^2$. The convergence theorems of both case studies are formalized and machine-checked in Lean 4.

Preface

In an era where frontier large language models (LLMs) provide remarkable mathematical intelligence, what role remains for human researchers? In this work, we (human experts) augment the general-purpose intelligence of LLMs with a technical domain-specific numerical methodology, and automate optimization research end-to-end, at a level not yet possible with LLMs alone. We hope this shows the community one way domain experts can productively contribute in an AI-driven research workflow: by augmenting the LLM's general-purpose intelligence with a domain-specific harness derived from their expertise.

1 Introduction

Since the pioneering work of Nesterov on accelerated gradient methods [43] and of Nemirovsky and Yudin on information-based complexity [42], the design of efficient and optimal first-order methods has been a central pursuit in the study of large-scale optimization. The performance estimation programming (PEP) framework [19, 65, 63] turned this pursuit into a computer-assisted discipline by formulating the worst-case performance computation of a first-order method as a convex semidefinite program (SDP). Since its inception, PEP has produced many notable algorithms including the optimized gradient method (OGM) [30, 17], its gradient-norm counterpart OGM-G [31], the information-theoretic exact method (ITEM) [62, 18]. Designing an optimal method through the PEP framework reduces to optimizing over the space of methods and branch-and-bound performance estimation programming (BnB-PEP) [14] formulates this design process as a nonconvex quadratically constrained quadratic program (QCQP) and solves it to global optimality using a spatial branch-and-bound algorithm.

Yet, this computer-assisted methodology can be heavily labor-intensive: every stage from numerically computing the method to coming up with the analytical form of the algorithm and its convergence analysis demands deep domain expertise, and the rate of discovery is set by human effort. While recent open-source packages such as PESTO [64] and PEPit [22] can help by solving the worst-case SDP numerically, they come at the cost of a rigid, language-specific syntax and a PEP derivation encoded in low-level library code that can be difficult for newcomers to parse or modify. Meanwhile, large language models (LLMs) and LLM-driven agents have begun to produce genuine algorithmic and mathematical discoveries, which we review in Section 1.2. The natural next step is to combine LLMs with these classical computer-assisted tools to further accelerate research in optimization.

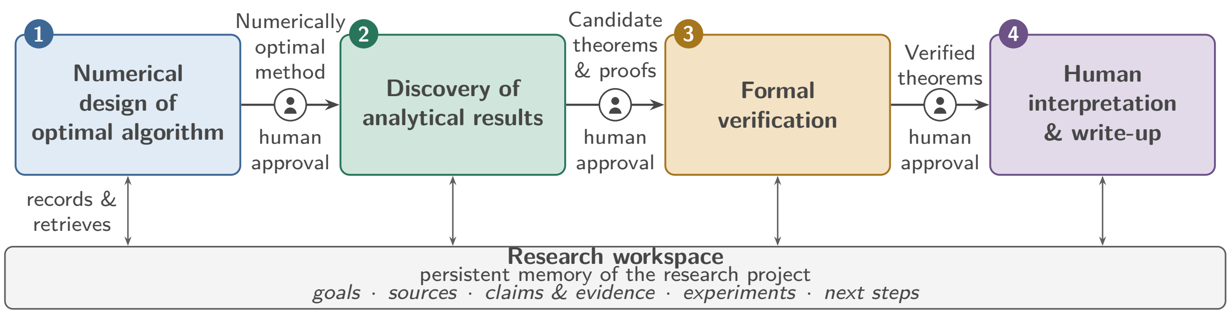

Figure 1: The researcher operates AutoOPT by conversing with an LLM agent in a research workspace in four stages. Stage 1 numerically designs an optimal first-order method for the problem class that the researcher specifies. Stage 2 consults a frontier LLM to propose an analytic form of the numerically designed method along with candidate convergence proofs. Stage 3 formally verifies the candidate theorems and proofs in the Lean proof assistant, ensuring every proof step is correct and explicit. In Stage 4, human authors interpret the verified results and write them up.

1.1 Contribution.

This work presents AutoOPT, a domain-specific harness that end-to-end automates optimization research by combining the computer-assisted PEP methodology with LLMs and Lean formalization; Figure 1 illustrates the pipeline.

We demonstrate AutoOPT on two case studies, each producing a result of independent interest. The first case study, *lemniscate acceleration* (Section 4.1), is a new accelerated gradient method for minimizing the gradient norm of an L-smooth convex function f with a minimizer $x_\star$. From any initial point x_0, its final iterate x_N satisfies

$$\|\nabla f(x_N)\|^2 \leq \frac{\varpi^4 L^2 \|x_0 - x_\star\|^2}{(N+1)^4},$$

where $\varpi \approx 2.622$ is the lemniscate constant. This rate is optimal in its N-dependence [40, 41], and its constant improves upon the previous best guarantee, obtained by concatenating OGM and OGM-G, by a factor of about 1.35 (Section 4.1). To the best of our knowledge, this is the first time the lemniscate constant and its associated elliptic functions have been connected to optimization theory.

The second case study, *analytic ITEM-f*, (Section 4.2) is the optimal method for minimizing an L-smooth, μ-strongly convex function f with minimizer $x_\star$ together with the accelerated linear rate

$$f(x_N) - f(x_\star) \le 4\left(1 - \sqrt{\frac{\mu}{L}}\right)^{2N} (f(x_0) - f(x_\star)),$$

whose per-step contraction factor $\left(1 - \sqrt{\mu/L}\right)^2$ matches the asymptotics of the complexity lower bound [18]. ITEM-f was previously available only through numerical values [62, Appendix E]. We formalize and machine-check the main convergence theorems of both lemniscate acceleration and ITEM-f in Lean 4 with mathlib [16, 66].

More broadly, this work serves as a case study on how domain experts can contribute productively to an AI-driven research workflow by designing and utilizing a domain-specific harness.

1.2 Prior work

AI-driven discovery and domain-specific harnesses. A rapidly growing body of work uses LLMs and related search methods as discovery engines. Evolutionary symbolic search produced the Lion optimizer [11]; AlphaTensor discovered faster matrix multiplication algorithms through reinforcement learning [20]; and FunSearch and AlphaEvolve couple LLM code generation with automatic evaluation to find new mathematical constructions and algorithmic improvements [55, 46]. At the level of whole research programs, the AI Scientist pursues end-to-end automation of machine-learning research [34, 35]; Aletheia reports research-level problem solving in mathematics [21]; an internal version of OpenAI's unreleased Astra model solved ten long-open problems in mathematics and theoretical computer science, with each argument formalized in Lean [52]; and, with the Lean 4 proof assistant and its mathematical library mathlib [16, 66] now mature enough to formalize research-level mathematics, AI-driven formal proof search has resolved open problems [67]. In the natural sciences, AlphaFold and RoseTTAFold transformed protein-structure prediction [28, 8], and the Virtual Lab coordinates a team of scientific agents that designed and experimentally validated new nanobodies [61].

In the modern parlance of agentic AI, a *harness* is the scaffolding built around an LLM, including the tools, prompts, workflows, and verification loops. We define a *domain-specific harness* to be a harness designed with expert domain knowledge for the purpose of augmenting the general-purpose intelligence of LLMs with the specialized methodology of a technical domain. Despite the large body of work using LLMs to conduct mathematical research, most of it operates the models through simple chat interfaces, and instances of domain-specific harnesses are not yet common. Although their authors do not use the term, AlphaEvolve [46] and Aletheia [21] can be regarded as domain-specific harnesses: AlphaEvolve targets scientific and algorithmic discovery across a range of computational problems, whereas Aletheia targets mathematical research. Both systems, however, remain broad in scope and were built in industry research labs with very large inference compute budgets. In contrast, AutoOPT specializes further, to the domain of first-order optimization research, and demonstrates this research principle at an academic scale of compute and token usage.

Computer-assisted design of optimization methods. The performance estimation methodology originates with Drori and Teboulle [19], who formulated the worst-case performance of a fixed-step first-order method as an optimization problem and bounded it through an SDP relaxation; Taylor, Hendrickx, and Glineur proved the relaxation exact through convex interpolation [65] and then extended the framework to the composite setup in [63]. The methodology has since driven a sequence of discoveries. Drori and Teboulle numerically constructed OGM [19], Kim and Fessler found its analytic description [30], and Drori proved its exact optimality through a matching lower bound [17]. Kim and Fessler constructed OGM-G, with the best known rate for reducing the gradient norm of smooth convex functions [31]. Taylor and Drori constructed ITEM for reducing the distance from the set of optimal solutions, whose exact optimality for smooth strongly convex minimization follows from a matching lower bound [62, 18]. BnB-PEP [14] unified the design side of this program by solving the underlying nonconvex QCQPs to certifiable global optimality, constructing optimal methods in setups where the prior convexity-based reformulations do not apply. Its methodology produced OptISTA, an optimal method for composite minimization [27]. Its numerical certificates also helped pave the way for the provably faster long-step gradient descent of Grimmer [23] and the silver stepsize schedule of Altschuler and Parrilo [2, 3], a line of work surveyed in [4]. In every one of these successes, the path from a numerically optimal solution to a theorem that includes analytic coefficients, a dual certificate, and a written proof was traversed by hand. AutoOPT automates precisely this path while keeping the human researcher in control of every decision boundary.

AI-assisted optimization theory. Very recently, frontier models have also begun proving research-level results in optimization theory: an AI agent writing directly in Lean discovered an improved last-iterate rate for anchored gradient descent ascent for minimax optimization [60], GPT 5.6 Sol Pro closed a query-complexity gap open since 1996 in derivative-free convex optimization, with the central lower bound subsequently formalized in Lean [29], and ChatGPT 5.5 produced the first convergence proof for Bregman Douglas–Rachford splitting as a matrix scaling algorithm [37]. In all three works the core workflow was that a human prompted a frontier model to solve a known open question, the model supplied the mathematics, and correctness was secured afterward, by the authors' own checking or by a Lean formalization provided for that one result.

Two recent works [68, 59] combine PEP with LLM consultation for optimization research. In [68], using ChatGPT and Codex for exploratory PEP computations, the authors discovered an optimal proximal gradient method for minimizing the sum of a smooth and a nonsmooth convex function, with the final Lyapunov proofs derived by the authors. Peppy [59] is an AI-assisted workflow for tight convergence analysis of known algorithms: given a problem class, a *known* algorithm, a performance measure, and an initial condition, it implements and solves the associated PEPs through the deterministic PEPFlow package, extracts fixed-horizon proof certificates, assembles Lyapunov partial sums, and simplifies them into closed-form analytic proofs, with symbolic checks along the way.

1.3 Organization

This paper is organized as follows. In Section 2, we review conversational chatbots, agentic research systems, coding and persistent agents, reusable agent skills, and the research practices on which AutoOPT builds. In Section 3, we describe AutoOPT methodology consisting of numerical design, symbolic discovery, formal verification, and human interpretation. In Section 4, we present lemniscate acceleration and analytic ITEM-f, respectively, together with their convergence analyses, continuous-time interpretations, AutoOPT design histories, and Lean formalizations. In Section 5, we discuss the applicability and limitations of AutoOPT and the broader role of domain-specific harnesses in AI-assisted research. Appendices A and B collect the deferred proofs, PEP formulations, and technical details for the two case studies.

1.4 Computational setup

We open-source the portable agent skills and Lean verification projects associated with AutoOPT at the link:

`https://github.com/Shuvomoy/AutoOPT`

Unless otherwise specified, we conducted the experiments from a laptop running macOS Tahoe with an Apple M5 Max processor and 128 GB of memory. The agentic experiments used GPT-5.5 [51] and GPT-5.6 Sol [50] through the USD 200-per-month ChatGPT Pro tier [47]. No internal or unreleased model was used for the agentic experiments in this paper. The Lean verification projects use Lean 4.32.0 [16] and mathlib 4.32.0 [66].

Within Stage 1 of AutoOPT, we formulated the numerical design problems in `JuMP` 1.31.1 [36], a domain-specific modeling language for mathematical optimization embedded in `Julia` 1.12.6 [9]. The reported BnB-PEP computations used `MOSEK` 11.2.2 [39] as the primary solver for the fixed-method SDPs and `KNITRO` 16.0.0 [10] and `Ipopt` 3.14.19 [69] as the primary solvers for the local QCQPs. `Gurobi` 13.0.2 [25] was used to perform spatial branch-and-bound. `MOSEK` and `Gurobi` provide no-cost academic licenses [38, 24]. `Ipopt` is open source under the Eclipse Public License 2.0 [12], while `KNITRO` is commercial and offers academic and teaching programs [7].

2 Background on agentic research

In this section, we briefly review recent agentic research practice, with an emphasis on what each class of tool can and cannot responsibly do in a mathematical research workflow.

From chatbots to agents. *Conversational* chatbots such as ChatGPT, Claude, and Gemini were the first widely used form of LLM assistance in research. A researcher can ask for explanations, brainstorming, proof debugging, literature orientation, or alternative formulations of a claim. This is extremely valuable, but it has a structural limitation: a chat agent does not own a repository, run the tests that would falsify its suggestions, or maintain a reproducible artifact trail. It can offer advice, but it cannot execute *research* over a long time horizon.

Agentic research systems. Recent work on using LLMs for research has moved in a more *agentic* direction, closing part of the propose-test-revise loop. Program-search systems such as FunSearch and AlphaEvolve use LLMs to generate candidate programs and evaluate them automatically, yielding new constructions and algorithmic improvements [55, 46]. Broader scientific-agent frameworks, such as the AI Scientist, attempt larger portions of the open-ended research cycle [34], and in mathematics, systems such as Aletheia point toward autonomous or semi-autonomous research-level problem solving [21].

Coding agents. For individual academic research and mathematical software development, the most practical setup beyond the conversational chatbot is the coding agent, such as Codex, Claude Code, and OpenCode [49, 5, 54]. A coding agent can operate on an active repository: it can read files, edit source code, run shell commands, inspect compiler or solver output, and iterate after errors. The important shift from the conversational chatbot is that the agent can act on artifacts the researcher will later inspect.

Persistent personal research agents extend this idea from a single coding session toward a more persistent research workflow. OpenClaw can be viewed as a personal AI assistant that runs on the user's devices and replies on messaging surfaces such as WhatsApp, Telegram, Slack, Mattermost, and Discord [53]. Hermes Agent is a self-improving agent with persistent memory, generated workflow packages, scheduled automations, and support for similar messaging surfaces [45]. These agents can support research progress by preserving state across repositories, files, commands, logs, memories, scheduled tasks, and other inspectable artifacts; their ability to maintain context across interactions and serve as standing research assistants is therefore highly valuable.

Agent skills. In the context of agentic LLMs, a *skill* is a reusable package of instructions and supporting material that an agent can load for a specialized task. In the open Agent Skills format, the minimal object is a folder containing a `SKILL.md` file with metadata and instructions; the folder may also contain scripts, references, templates, and other resources [1]. OpenAI Codex and Claude Code document skill mechanisms based on this principle [48, 6]. For an optimization workflow, the point of a skill is to move recurring guidance out of a transient prompt and into an inspectable artifact.

Agentic research practice. Across these tool classes, effective agentic research consistently depends on persistent instructions, inspectable reports and artifacts, staged evaluation, explicit verification, and intermittent human steering [74]. Mathematical research tightens each requirement. A claimed algorithm must have a precise statement; a computational run must record solver status, tolerances, and enough environment information to be reproduced; a proof attempt must be labeled theorem, conjecture, checked derivation, or failed approach; and a formal-verification claim requires actual checker output. AutoOPT instantiates these requirements for optimization research: each stage of the pipeline is packaged as a skill that any skill-loading agent can execute, every stage is approved or revised by a human, and the research state lives in a versioned repository. We describe the framework in Section 3.

3 Methodology

AutoOPT is an agentic framework for the end-to-end design, convergence analysis, and formal verification of optimal first-order methods. The framework has four stages, illustrated in Figure 1 and detailed in the following subsections. Throughout this section, we describe AutoOPT at a high level, focusing on how to use the methodology; readers interested in implementation details can refer to the appendix and the accompanying skills.

Operating AutoOPT. AutoOPT is model- and agent-agnostic: a user can run it from any skill-loading agent, e.g., Codex, Claude Code, or OpenCode, and can use any commercial or open-source LLM supported by the agent. The user operates AutoOPT by conversing with the agent.

At the start, AutoOPT invokes the agent skill research-repo-manager, which creates a research workspace in a directory of the user's choice. The workspace serves as persistent memory for the research project, keeping every source, claim, experiment, artifact, reproducibility note, deferred goal, and potential next step explicit and inspectable. The workspace overcomes a key limitation of single-session LLM workflows: as a conversation grows, the LLM's context window becomes crowded, compressed, or incomplete, making it harder to reliably recover earlier decisions, assumptions, and evidence. By storing the research state in versioned repository files rather than only in the chat transcript, research-repo-manager lets a later session reload the current goals, findings, artifacts, and next action directly from the workspace.

3.1 Stage 1: Numerical design of optimal first-order methods via bnb-pep-skill

The design problem. The research question AutoOPT addresses is finding the fastest gradient-based method for a given problem class: minimizing a function $f\colon \mathbb{R}^d \to \mathbb{R}$ from a known *function class* $\mathcal{F}$, such as smooth convex functions, smooth strongly convex functions, weakly convex functions, or a nonsmooth class. Here, "fastest" means the best worst-case guarantee with respect to a chosen performance measure $\mathcal{E}$. For example, the goal may be to design an N-step algorithm producing a final point x_N such that $\mathcal{E} = f(x_N) - \inf f$ or $\mathcal{E} = \|\nabla f(x_N)\|^2$ is guaranteed to be small, subject to an initial condition $\mathcal{C} \leq 0$ normalizing the starting point, such as $\mathcal{C} = \|x_0 - x_\star\|^2 - R^2$ for a given $R > 0$.

AutoOPT searches over the class of *fixed-step first-order methods* (FSFOMs): methods that form each iterate from the starting point $x_0 \in \mathbb{R}^d$ and the first-order information observed so far via

$$x_i = x_{i-1} - \sum_{j=0}^{i-1} h_{i,j}\, f'(x_j), \qquad i = 1, \ldots, N,$$

where $f'(x_j) \in \partial f(x_j)$ is a subgradient of f at x_j and $\{h_{i,j}\}_{0 \leq j < i \leq N}$ is a lower triangular array of stepsizes; lower triangularity expresses that each update depends only on the first-order information observed so far. The stepsizes $\{h_{i,j}\}_{0 \leq j < i \leq N}$ parameterize the FSFOM and are the main decision variables we optimize over; we write $\mathcal{M}_N$ for the class of all N-step FSFOMs. We write $M = \{h_{i,j}\}_{0 \leq j < i \leq N} \in \mathcal{M}_N$ to denote an individual FSFOM.

Formally, the design problem becomes a minimax problem of the form

$$\mathcal{R}^\star(\mathcal{M}_N) \triangleq \min_{M \in \mathcal{M}_N} \underbrace{\max \left\{\mathcal{E} \;:\; f \in \mathcal{F},\ \mathcal{C} \leq 0,\ x_1, \ldots, x_N \text{ generated by } M = \{h_{i,j}\}_{0 \leq j < i \leq N}\right\}}_{\mathcal{R}(M),\ \text{the worst-case performance of } M}. \quad (1)$$

From function classes to finitely many inequalities. At first sight, the minimax problem above may seem intractable as the inner maximization runs over all functions in $\mathcal{F}$, an infinite-dimensional class. The performance estimation methodology [19, 65] removes this obstacle in two steps. First, for a fixed method $M = \{h_{i,j}\}_{0\le j<i\le N}$, the inner problem depends on f only through its values and first-order information at the iterates $x_0, x_1, \dots, x_N$ and at a stationary point (minimizer for convex setups) $x_\star$, adjoined to encode stationarity (optimality for convex setups) through $0 \in \partial f(x_\star)$; collecting these indices in $I_N^\star \triangleq \{0, 1, \dots, N, \star\}$ and writing $g_i \in \partial f(x_i)$ and $f_i \triangleq f(x_i)$, the trajectory data $\{(x_i, g_i, f_i)\}_{i\in I_N^\star}$ carry everything the inner problem needs to know about f. Second, the function classes the methodology covers are *quadratically representable* [13, 14]: membership in $\mathcal{F}$ is described by inequalities of the form

$$\begin{aligned} c_0 f(y) \ge c_1 f(x) + c_2 \|x\|^2 + c_3 \|y\|^2 + c_4 \|u\|^2 + c_5 \|v\|^2 + c_6\langle x, y\rangle \\ + c_7\langle x, u\rangle + c_8\langle x, v\rangle + c_9\langle y, u\rangle + c_{10}\langle y, v\rangle + c_{11}\langle u, v\rangle \end{aligned} \tag{2}$$

for all $x, y \in \mathbb{R}^d$, $u \in \partial f(x)$, and $v \in \partial f(y)$, where ∂f denotes the subdifferential, reducing to $\partial f(x) = \{\nabla f(x)\}$ for the differentiable classes, and $c_0, \dots, c_{11}$ are fixed coefficients determined by the class parameters; smooth convex, smooth strongly convex, and weakly convex functions are all of this form.

Technically, replacing the constraint $f \in \mathcal{F}$ with the inequalities (2) imposed only on the finitely many pairs in $I_N^\star$ is, a priori, a relaxation. Establishing that this finite system is lossless requires a separate theorem for each function class, and such characterizations are known as *interpolation conditions*. We refer the reader to prior work for a detailed discussion of this issue; see, for example, [65, 56, 57].

Gram lifting: a convex SDP for fixed stepsizes. It is convenient to reparameterize the FSFOM as $x_i = x_0 - \sum_{j=0}^{i-1} \alpha_{i,j} g_j$, where $\alpha_{i,j} \triangleq \sum_{k=j+1}^{i} h_{k,j}$; this triangular map is invertible, so α is the design variable. Collect the relevant trajectory vectors as columns of a matrix P and the sampled function values in a row vector F, and form the Gram matrix $G \triangleq P^\top P \succeq 0$. Since the iterates are affine in α, the sampled inequalities (2), the performance measure, and the initial condition are affine in (F, G), with Gram-side coefficient matrices whose entries have degree at most two in α. Writing $\mathbf{tr}$ for the trace, the inner worst-case problem becomes

$$\mathcal{R}(M) = \left(\begin{array}{l} \text{maximize} \quad F\mathbf{u}_\mathcal{E} + \mathbf{tr}\, G Q_\mathcal{E}(\alpha) \\ \text{subject to} \\ F\mathbf{u}_{i,j} + \mathbf{tr}\, G Q_{i,j}(\alpha) \le 0, \quad i, j \in I_N^\star : i \ne j, \quad \triangleright \text{ dual var. } \lambda_{i,j} \ge 0 \\ -G \preceq 0, \quad \triangleright \text{ dual var. } Z \succeq 0 \\ F\mathbf{u}_\mathcal{C} + \mathbf{tr}\, G Q_\mathcal{C}(\alpha) \le r, \quad \triangleright \text{ dual var. } \nu \ge 0 \\ \mathbf{rank}\, G \le d, \end{array}\right), \tag{3}$$

where the vectors $\mathbf{u}_\mathcal{E}, \mathbf{u}_\mathcal{C}, \mathbf{u}_{i,j}$, the symmetric matrices $Q_\mathcal{E}(\alpha), Q_\mathcal{C}(\alpha), Q_{i,j}(\alpha)$, and the scalar r are assembled mechanically from (2), $\mathcal{E}$, and $\mathcal{C}$; for instance, the initial condition $\mathcal{C} = \|x_0 - x_\star\|^2 - R^2$ has $\mathbf{u}_\mathcal{C} = 0$ and $r = R^2$. Under the large-scale assumption that d is at least the number of columns of P, the constraint $\mathbf{rank}\, G \le d$ is vacuous. Dropping it makes (3) a convex semidefinite program (SDP) in (F, G), free of the problem dimension d. This mechanical assembly is exactly what bnb-pep-skill automates: it fixes the exact contents of P and F for the instance at hand, derives the corresponding vectors $\mathbf{u}$ and matrices $Q(\alpha)$ symbolically, and records the resulting Gram dimension together with the matching assumption on d. Appendix A.3 carries out this assembly explicitly for lemniscate acceleration.

Dualization: from worst-case values to proofs. SDP duality turns the inner maximization (3) into a minimization,

$$\overline{\mathcal{R}}(M) = \left(\begin{array}{l} \text{minimize} \quad \nu r \\ \text{subject to} \\ \mathbf{u}_{\mathcal{E}} - \nu \mathbf{u}_{\mathcal{C}} - \sum_{i,j \in I_N^\star : i \neq j} \lambda_{i,j} \mathbf{u}_{i,j} = 0, \\ \nu Q_{\mathcal{C}}(\alpha) - Q_{\mathcal{E}}(\alpha) + \sum_{i,j \in I_N^\star : i \neq j} \lambda_{i,j} Q_{i,j}(\alpha) = Z, \\ Z \succeq 0, \\ \nu \geq 0, \ \lambda_{i,j} \geq 0, \quad i,j \in I_N^\star : i \neq j \end{array} \right), \tag{4}$$

whose decision variables $\lambda = \{\lambda_{i,j}\}_{i,j \in I_N^\star : i \neq j}$, ν, and the positive-semidefinite slack matrix Z we call the *inner-dual variables*. By weak duality, $\mathcal{R}(M) \leq \overline{\mathcal{R}}(M)$, and this inequality carries the interpretive weight of the methodology: any feasible inner-dual point is a certificate, an explicit nonnegative aggregation of the sampled inequalities proving that $\mathcal{E} \leq \nu r$ for every function in the class satisfying the initial condition, that is, a machine-checkable convergence proof. Strong duality holds under mild conditions which implies $\mathcal{R}(M) = \overline{\mathcal{R}}(M)$ and is assumed in this paper. Each derivation that bnb-pep-skill writes states this dual explicitly, with the sign conventions of its multipliers fixed, and records whether the resulting bound invokes strong duality or only the weak direction. The certificate view is what the later stages of AutoOPT use internally: Stage 2 (Section 3.2) fits analytic expressions to the numerically designed $(\alpha^\star, \lambda^\star, \nu^\star, Z^\star)$ and then converts the fitted multipliers into the Lyapunov function based convergence proofs, and Stage 3 (Section 3.3) formalizes those Lyapunov proofs in Lean.

Cholesky elimination and the BnB-PEP QCQP. Substituting the dual (4) for the inner maximization in (1) collapses the minimax into a single minimization, jointly over the stepsizes and the inner-dual variables. In other words, the single minimization problem jointly optimizes over the algorithm and its convergence proof.

However, this joint optimization is non-convex. The BnB-PEP methodology [14] eliminates the semidefinite constraint $Z \succeq 0$ through the parameterization $Z = VV^\top$ and, by introducing auxiliary variables to represent bilinear or quadratic terms when necessary, formulates the numerical optimization problem as a nonconvex quadratically constrained quadratic program (QCQP)

$$\mathcal{R}^\star(\mathcal{M}_N) = \left(\begin{array}{l} \text{minimize} \quad \nu r \\ \text{subject to} \\ \mathbf{u}_{\mathcal{E}} - \nu \mathbf{u}_{\mathcal{C}} - \sum_{i,j \in I_N^\star : i \neq j} \lambda_{i,j} \mathbf{u}_{i,j} = 0, \\ \nu Q_{\mathcal{C}}(\alpha) - Q_{\mathcal{E}}(\alpha) + \sum_{i,j \in I_N^\star : i \neq j} \lambda_{i,j} Q_{i,j}(\alpha) = Z, \\ V \text{ is lower triangular with nonnegative diagonals,} \\ Z = VV^\top, \\ \nu \geq 0, \ \lambda_{i,j} \geq 0, \quad i,j \in I_N^\star : i \neq j \end{array} \right), \tag{5}$$

with decision variables $\{\alpha_{i,j}\}_{0 \leq j < i \leq N}$, λ, ν, Z, and V. Any feasible point to this QCQP corresponds to an algorithm and its convergence proof, while an optimal point gives the best algorithm along with its convergence proof.

The three-stage solution strategy. Despite its nonconvexity, (5) can be solved to certifiable global optimality, up to prescribed tolerances, by the three-stage BnB-PEP strategy of [14]: a fixed-method dual SDP supplies a feasible point, an interior-point solver refines it to a local optimum, and a customized spatial branch-and-bound solver performs the global certification. The final stage is made practical by the QCQP-specific bounds and cuts developed in that work. Accordingly, bnb-pep-skill runs the first two stages by default and invokes the global stage only at the user's explicit request. The BnB-PEP methodology can be laborious to implement and debug and the manual coding is cumbersome and error-prone and requires substantial PEP expertise. This implementation barrier has previously limited its use.

Automation with bnb-pep-skill. The methodology described above, including several technical details omitted from the exposition, can be tedious to formulate and implement by hand. Our bnb-pep-skill automates this process through an executable and auditable protocol.

A run of the bnb-pep-skill opens as a structured interview that assumes no PEP expertise on the user's part and produces a fully specified instance: the function class with its inequalities, the method update equations, the performance measure, the initial condition, the constants, and the horizon N. Whenever a required item is missing or ambiguous, the skill asks a targeted question rather than guessing, and it accepts the instance only when another researcher could write down every constraint and identify every decision variable without making further choices.

Code generation and guardrails. Once the user approves the mathematics, bnb-pep-skill writes code. The generated `Julia/JuMP` [36] program builds the primal and dual SDPs of the instance together with the design problem (5) and numerically solves the non-convex QCQP.

This protocol makes concrete the division of labor behind the domain-specific harness of Section 1: the LLM agent interviews the user, instantiating the derivation, and writing code, while the harness supplies the accumulated domain expertise, from exactness theorems for interpolation conditions to warm-start numerics, that a general-purpose model cannot be relied upon to reproduce unaided. The same division of labor recurs in Stages 2 and 3.

The agentic workflow. In the first stage, AutoOPT invokes bnb-pep-skill and interviews the user about the problem setup for which an optimal first-order method is sought. The interview assumes no PEP expertise on the user's part. From the agreed setup, the skill writes out the complete derivation, instantiating (2)–(5) for the user's instance, and stops for the user's approval of the mathematics before any code is written.

Once the user approves the derivation, the skill generates and runs `Julia/JuMP` code to obtain candidate stepsizes and associated inner-dual certificates. It stores the derivation, the code, the numerical solution, the solver status, and the logs in the research workspace. Figure 2 summarizes the workflow; the complete derivations produced this way for our two case studies appear, lightly edited for presentation, in Appendices A.3 and B.3.

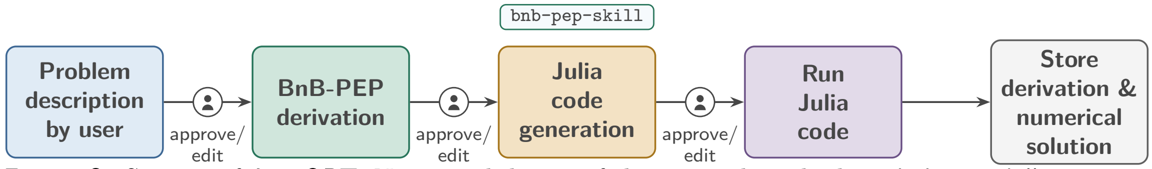

Figure 2: Stage 1 of AutoOPT. Numerical design of the optimal method via bnb-pep-skill.

3.2 Stage 2: Symbolic discovery of analytic descriptions and convergence proofs via LLMs

Numerical to symbolic discovery. Once the numerically optimal stepsizes and the BnB-PEP mathematical derivation are available from Stage 1, AutoOPT invokes the skill frontier-llm-consult to discover analytic descriptions of the numerically designed methods and candidate convergence proofs. Figure 3 summarizes the stage. The skill frontier-llm-consult supports two routes. The external consultation route, chatgpt-pro-session, runs a persistent ChatGPT Pro session in a web browser and is the default for ordinary Stage 2 work; the same route also covers bounded one-shot reviews, each conducted as a single approved turn in a fresh session. For the consultation route, the skill assembles a deliberately small context bundle containing a self-contained prompt, the numerical findings from the previous stage, and the relevant mathematical background and derivations. The skill then presents the bundle for user approval; the user may approve it as is or edit it first, and only after approval is the bundle shared with the frontier model. The persistent session enables symbolic discovery from numerical findings through several rounds of pattern search, algebraic refinement, and follow-up questions from the user. The native route, solve-with-highest-reasoning, runs only on explicit user invocation and is not an external consultation: the host agent itself conducts a long native campaign on the problem, using the strongest currently available Codex GPT model at its highest supported reasoning setting, with direct repository exploration, local tools, and adversarial verification, over at least eight wall-clock hours. The user approves the repository scope, no problem material is sent to any external model, and the full evidence trail is archived in the research workspace. For the consultation route, the user can edit the underlying `SKILL.md` file to substitute a different frontier model.

Whichever route is selected, discovery proceeds in three substeps. The frontier model first identifies the analytic form of the numerically designed method, that is, an analytic parametrization of the numerically optimal FSFOM stepsizes from Stage 1, together with an analytic feasible point of the inner dual (4). By the certificate view of Section 3.1, feasibility of this point already proves the worst-case bound, so the first substep yields a candidate convergence proof by dual feasibility. In the second substep, the model recasts the method from this raw stepsize parametrization into a simple, memory-efficient algorithmic form, recursive or momentum-based, which is the form in which this paper states each method. In the third substep, the frontier model derives a second argument from the same certificate: reading coefficients off the certificate's multipliers, it constructs a Lyapunov sequence of potentials that is nonincreasing along the iterates and whose endpoint values give the claimed bound, with the potentials stated directly on the recast form of the second substep. Both arguments are candidate proofs of the same theorem. This paper presents the Lyapunov proofs, the form that a classical optimization audience reads most naturally; the dual-feasibility certificates remain intermediate artifacts of the research workspace, and it is their multipliers that the printed proofs are built from.

Attached are the Stage 1 outputs of AutoOPT for the design of an N-step FSFOM that minimizes the worst-case squared gradient norm $\|\nabla f(x_N)\|^2$ over L-smooth convex functions with $\|x_0 - x_\star\| \leq R$: the BnB-PEP derivation (derivation.md), the Julia/JuMP code that solved it, and a CSV sweep (stepsize_certificate_sweep.csv) of the numerically optimal stepsizes h and inner-dual certificates (λ, ν, Z) for the horizons N = 1,...,25 with first 5 being globally optimal via spatial branch-and-bound and the last 20 being locally optimal.
From these data, propose a candidate analytic parametrization of the stepsizes h valid for every horizon N, together with a candidate analytic feasible point (λ, ν, Z) of the inner dual SDP stated in the derivation. Feasibility of that point already certifies the worst-case bound, so also state the candidate rate it implies in closed form. Every proposed formula is a candidate, not a proof; report which entries of the sweep you checked it against and to what precision. Please ask me clarifying questions if anything is not clear to you.

Your parametrization stores the full lower-triangular stepsize array h. Recast the method into an equivalent memory-efficient form, recursive or momentum-based, that reproduces the same iterates $x_0, \ldots, x_N$ while carrying only a constant number of vectors, and give closed-form expressions for the coefficients of the recast update. Check the recast against the h-form on the sweep horizons and report the agreement.

Reading the coefficients off the multipliers (λ, ν) of your candidate certificate, construct a candidate Lyapunov proof of the same bound: a sequence of potentials $\mathcal{V}_0, \ldots, \mathcal{V}_N$, stated directly on the recast form of your previous reply, that is nonincreasing along the iterates and whose endpoint values yield the claimed bound. Express each one-step decrement as a nonnegative combination of the interpolation inequalities from the derivation, and flag any step you cannot close.

Figure 4: An illustrative Stage-2 prompt sequence for the lemniscate case study: one initial prompt followed by two follow-up prompts corresponding to the three substeps of Section 3.2.

Figure 4 makes the workflow concrete for the lemniscate acceleration case study of Section 4.1. It shows an illustrative Stage-2 prompt sequence, consultation-shaped because both published runs used the external chatgpt-pro-session route: one initial consultation prompt requesting the first substep, followed by two short turns in the same session requesting the remaining two. Only the author-written task text is printed; the consultation package that wraps it and the approve-before-send gate are those described above.

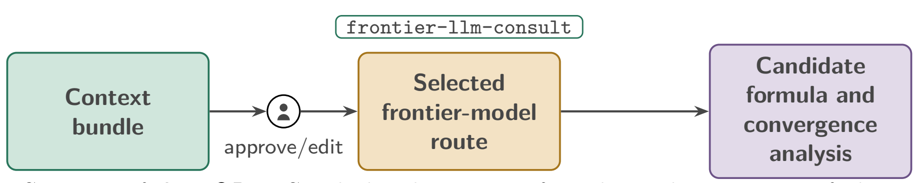

Figure 3: Stage 2 of AutoOPT. Symbolic discovery of analytic descriptions of the numerically designed methods and their candidate convergence proofs via frontier-llm-consult.

Preparation for machine verification. At the end of this stage, AutoOPT saves the outputs from the frontier LLMs in the research workspace so that the reasoning trace is inspectable by the user. AutoOPT marks the frontier LLM output as a "candidate" analytical solution: it may give the researcher something concrete to test, simplify, formalize, or prove, but it still needs independent validation, which is supplied, for the results advanced to the paper, by the Lean verification stage described next.

3.3 Stage 3: Formal verification of candidate theorems via lean-verify

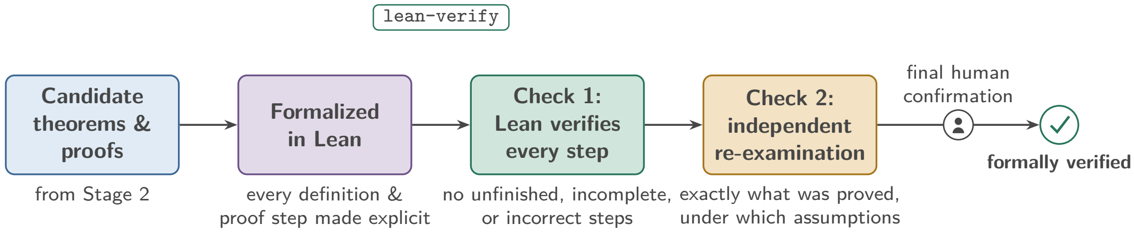

Figure 5: Formal verification via lean-verify: candidate theorems and proofs from Stage 2 are formalized in Lean, checked by Lean itself, and re-examined by an independent verifier; the result is recorded as formally verified only after the human authors confirm that the formal theorems faithfully express the paper's claims.

From candidate to formalization. Stage 3 of AutoOPT is the formal verification stage. Here, AutoOPT invokes the skill lean-verify, which uses machine verification in the open-source proof assistant Lean to determine whether the chain of logic in the candidate theorems and proposed proofs discovered in Stage 2 is correct. During machine verification, Lean records every definition, theorem statement, and proof precisely, examines the logic of every step of a proof, and accepts a proof only when each step is correct and explicit.

The lean-verify skill first formalizes in Lean the candidate theorems together with the proofs selected for presentation in the paper, which for our case studies are the Lyapunov proofs derived in Stage 2, and reconstructs the proof steps there in detail, so that any gap, hidden assumption, or algebraic slip in the candidate argument is exposed rather than granted on trust. A proof is recorded as *formally verified* only after two successive independent checks followed by a final human confirmation. The first check is Lean's own internal verification, which confirms that the accepted proof contains no unfinished, incomplete, or incorrect steps. The second independent check is through a separate verifier called the comparator that reexamines the finished formalization and confirms exactly which statements have been proved and which assumptions the proofs were permitted to use, so that what has been machine verified is pinned down without ambiguity. Once the two checks are complete, the human authors confirm that the formal theorem statements in Lean faithfully express the corresponding theorem statements in the paper, and only then record the result as formally verified. Figure 5 summarizes the workflow of the skill lean-verify.

An illustrative Stage-3 prompt, the file structure of each resulting Lean project, and per-project summary statistics are presented with the case studies, in Sections 4.1.5 and 4.2.4.

3.4 Stage 4: Human interpretation and write-up

The goal of this paper is to provide a new methodology for automating what used to be a highly time-consuming and technical component of optimization research. As researchers, we value accelerating progress and solving harder and more interesting problems.

However, we hope that this pursuit of automation does not cause us to lose sight of the fact that mathematical research is a human endeavor and that the value of research is realized when it is conveyed and used by other researchers. Therefore, the final part of the pipeline is for the human authors to interpret the results, frame why they are of interest to the community, and present insights that go beyond the mere statement of the result.

In what follows, we briefly offer our personal views on the best practices for humans using such methodologies.

Publication restraint. In this era of AI accelerating research, proofs of novel results are no longer scarce. We therefore believe authors should now ask whether a novel and correct result is sufficiently interesting to the research community, and exercise judgment and parsimony by forgoing publication of results that do not meet this new higher bar.

Distilling generalizable insights. While optimization is an applied field, worst-case complexity guarantees are theoretical analyses whose value lies in expanding our collective understanding of algorithm design and eventually informing the optimization practice. Merely presenting a new algorithm with a verifiable but opaque analysis delivers little of this value, especially when AI assistance means the human authors themselves lack strong intuition about the result.

It is therefore essential that authors extract compact, generalizable insights from the formal discovery and present them in a form that is maximally useful to human researchers. For example, a machine-discovered optimization algorithm may first arrive in a dense stepsize-matrix representation. In our pipeline, Stage 2 already recasts such a method into an equivalent memory-efficient, recursive, or momentum-based formulation (Section 3.2); what remains a matter of human judgment is which of the available forms serves readers best. The authors should make that choice deliberately, potentially with further AI assistance, and present the formulation that a human reader can most readily implement, interpret, and analyze.

A calibrated standard of evidence. In the final presentation, the authors should clearly assign each claim an evidentiary status (human-verified, formally verified, LLM-verified, numerical evidence, or conjecture), so that no intermediate object or computational observation is presented as stronger evidence than it actually provides.

We also emphasize that human authors must exercise their own due diligence in verification. Even a Lean formalization may fail to establish the intended result: the formal theorem statement may encode a claim weaker than the paper's natural-language statement, or the proof may remain incomplete because it contains `sorry` placeholders. Even with thorough harnesses, AI systems still make mistakes, and the human authors are ultimately accountable for the work they publish.

4 New algorithms discovered, analyzed, and verified using AutoOPT

In this section, we present two new first-order algorithms found using AutoOPT: *lemniscate acceleration* and *analytic ITEM-f*. We present these algorithms both as algorithmic advances in their own right and as demonstrations of the strength of AutoOPT.

Note that the order of discovery via AutoOPT differs from the order of presentation of those results in this paper. For each method, AutoOPT first solved the BnB-PEP design problem numerically, next identified the method's analytic form together with an analytic dual certificate of its worst-case guarantee, then recast the method from the raw stepsize parametrization of the design problem into the memory-efficient form in which this paper states it, and finally converted the certificate's multipliers into a Lyapunov convergence proof for the presented form. The paper presents each method in the order a reader expects: algorithm, convergence theorem, and proof. The printed proofs are the Lyapunov proofs from Stage 2, with Lean formalization discussed in Sections 4.1.5 and 4.2.4; the subsections titled "AutoOPT design" recount how each method was found, and Appendices A.3 and B.3 reproduce the Stage-1 PEP formulations.

We use the following notation and notions in the rest of the paper. Write $\mathbb{R}^d$ for the underlying Euclidean space and we write $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ to denote the standard inner product and norm on $\mathbb{R}^d$. Write $\mathbb{R}^{m \times n}$ for the set of $m \times n$ matrices, $\mathbb{S}^n$ for the set of $n \times n$ symmetric matrices, and $\mathbb{S}^n_+$ for the set of $n \times n$ positive-semidefinite matrices. We use the standard notation $e_i \in \mathbb{R}^d$ for the unit vector having a single 1 as its i-th component. Write $(\cdot \odot \cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d \times d}$ to denote the symmetric outer product, that is $x \odot y = (xy^\intercal + yx^\intercal)/2$ for every $x, y \in \mathbb{R}^d$, and $\delta_{i,j}$ denotes the Kronecker delta, equal to 1 if $i = j$ and 0 otherwise. We write $\mathbf{tr}$ for the trace of a square matrix. For $0 \leq \mu < L < \infty$, we write $\mathcal{F}_{\mu,L}$ for the class of L-smooth μ-strongly convex functions $f \colon \mathbb{R}^d \to \mathbb{R}$, that is, differentiable functions for which ∇f is L-Lipschitz and $f - (\mu/2)\|\cdot\|^2$ is convex; the case $\mu = 0$ gives the class $\mathcal{F}_{0,L}$ of L-smooth convex functions. Every function $f \in \mathcal{F}_{0,L}$ satisfies

$$f(x) - f(y) - \langle \nabla f(y), x - y \rangle - \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2 \geq 0 \tag{6}$$

for all $x, y \in \mathbb{R}^d$ [15, Theorem A.1].

Throughout, $x_\star$ denotes a minimizer of the function f under consideration, $f_\star \triangleq f(x_\star)$, and the subscript $\star$ refers to this optimal point. For a method that runs for N iterations and produces the iterates $x_0, x_1, \ldots, x_N$, we collect the iteration indices together with the optimal point in the index set $I_N^\star \triangleq \{0, 1, \ldots, N, \star\}$.

4.1 Lemniscate acceleration for efficient gradient-norm minimization

Consider the unconstrained minimization problem

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x), \tag{7}$$

where $f \colon \mathbb{R}^d \to \mathbb{R}$ is an L-smooth convex function. We seek an algorithm that uses N gradient evaluations to produce an output x_N with a small worst-case guarantee on $\|\nabla f(x_N)\|$.

Throughout this section, we use the single-gradient-step notation

$$z^+ \triangleq z - \frac{1}{L}\nabla f(z), \qquad \forall\, z \in \mathbb{R}^d.$$

Algorithm statement

The *lemniscate accelerated gradient method* (LemniAcc) is

$$\begin{aligned} z_{k+1} &= z_k - \Omega_N \left(\frac{1 + \rho_{k+1}^2}{2\rho_{k+1}} - \frac{1 + \rho_k^2}{2\rho_k} \right) \frac{1}{L} \nabla f(x_k) \\ x_{k+1} &= x_k^+ + \left(\frac{1 + \rho_{k+1}^2}{1 - \rho_{k+1}^2} - \frac{1 + \rho_{k+2}^2}{1 - \rho_{k+2}^2} \right) z_{k+1} \end{aligned} \tag{LemniAcc}$$

N	Ω_N	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7
1	2.4142	0.4142						
2	4.1628	0.6126	0.2402					
3	6.2479	0.7241	0.4142	0.1601				
4	8.6645	0.7931	0.5380	0.3004	0.1154			
5	11.4075	0.8388	0.6277	0.4142	0.2287	0.0877		
6	14.4730	0.8707	0.6942	0.5052	0.3287	0.1805	0.0691	
7	17.8578	0.8939	0.7446	0.5780	0.4142	0.2675	0.1464	0.0560

Table 1: Values of Ω_N and $\rho_1, \dots, \rho_N$ for $N = 1, \dots, 7$, computed by the bisection method.

for $k = 0, 1, 2, \dots, N-1$, where $N \geq 1$ is an integer-valued prespecified iteration budget, $x_0 \in \mathbb{R}^d$ is the starting point, $z_0 = 0$, and $\Omega_N > 0$ and $1 = \rho_0 > \rho_1 > \cdots > \rho_N > \rho_{N+1} = 0$ are constants that we define in Lemma 1.

We can also express (LemniAcc) equivalently in the so-called momentum form

$$\begin{aligned} x_1 &= x_0^+ + \alpha^{(\text{init})}(x_0^+ - x_0), \\ x_{k+1} &= x_k^+ + \alpha_k^{(\text{mom})}(x_k^+ - x_{k-1}^+) + \alpha_k^{(\text{corr})}(x_k^+ - x_k), \qquad k = 1, \dots, N-1, \end{aligned} \tag{8}$$

where

$$\begin{aligned} \alpha^{(\text{init})} &= \Omega_N \left(\frac{1+\rho_1^2}{2\rho_1} - \frac{1+\rho_0^2}{2\rho_0} \right) \left(\frac{1+\rho_N^2}{2\rho_N} - \frac{1+\rho_{N-1}^2}{2\rho_{N-1}} \right), \\ \alpha_k^{(\text{mom})} &= \frac{\frac{1+\rho_{N-k}^2}{2\rho_{N-k}} - \frac{1+\rho_{N-k-1}^2}{2\rho_{N-k-1}}}{\frac{1+\rho_{N-k+1}^2}{2\rho_{N-k+1}} - \frac{1+\rho_{N-k}^2}{2\rho_{N-k}}}, \\ \alpha_k^{(\text{corr})} &= \left(\frac{1+\rho_{N-k}^2}{2\rho_{N-k}} - \frac{1+\rho_{N-k-1}^2}{2\rho_{N-k-1}} \right) \left[\Omega_N \left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k} \right) - \frac{1}{\frac{1+\rho_{N-k+1}^2}{2\rho_{N-k+1}} - \frac{1+\rho_{N-k}^2}{2\rho_{N-k}}} \right]. \end{aligned}$$

The two forms are equivalent in the sense that they produce the same iterates $x_0, \dots, x_N$; a proof is given in Appendix A.2.

Construction of the coefficients

To fully define (LemniAcc), it remains to define the coefficients $\rho_0, \rho_1, \dots, \rho_N, \rho_{N+1}$ and Ω_N.

Lemma 1. *Let N be a nonnegative integer. There exists a unique $\Omega_N > 0$ for which there is a strictly decreasing nonnegative sequence $1 = \rho_0 > \rho_1 > \ldots > \rho_{N+1} = 0$ satisfying the recurrence equation*

$$\Omega_N \left(\rho_k - \rho_{k+1}\right)^2 = \rho_k \left(1 - {\rho_{k+1}}^2\right), \quad k = 0, \dots, N. \tag{9}$$

Moreover, for this value of Ω_N, the sequence $\rho_0, \rho_1, \dots, \rho_{N+1}$ is uniquely determined.

We can calculate the numerical values of $\{\rho_k\}_{k=0}^{N+1}$ and Ω_N using bisection based on $1 \le \Omega_N \le (N+1)^2$. Appendix A.1 gives the exact shooting decision rule, its termination case, and its correctness proof. Table 1 reports the values of Ω_N and of the nontrivial entries $\rho_1, \dots, \rho_N$ for $N = 1, \dots, 7$.

We note that the sequence $\{\rho_k\}_{k=0}^{N+1}$ has a symmetry: the map $T(\rho) \triangleq (1-\rho)/(1+\rho)$, which satisfies $T \circ T = \mathrm{id}$, relates mirrored entries through $\rho_{N+1-k} = T(\rho_k)$ for $0 \le k \le N+1$. This follows from the uniqueness in Lemma 1 together with the fact that $(\rho_k, \rho_{k+1}) = (a, b)$ satisfies the recurrence (9) if and only if $(T(b), T(a))$ does. In particular, for odd N, the middle index $k = (N+1)/2$ is its own mirror image, so $\rho_{(N+1)/2}$ must be the point such that $T(\rho) = \rho$, namely $\rho_{(N+1)/2} = \sqrt{2} - 1 \approx 0.4142$.

Convergence theorem

Theorem 1. *Let $N \ge 1$ be an integer, and let $f\colon \mathbb{R}^d \to \mathbb{R}$ be an L-smooth convex function with a minimizer $x_\star$. Then, the final iterate x_N of* (LemniAcc) *satisfies*

$$\|\nabla f(x_N)\|^2 \le \frac{L^2 \|x_0 - x_\star\|^2}{\Omega_N^2} \le \frac{\varpi^4 L^2 \|x_0 - x_\star\|^2}{(N+1)^4}.$$

Here, $\varpi = 2\int_0^1 1/\sqrt{1-x^4}\,dx$ is the lemniscate constant, and the latter relaxation comes from $(N+1)^2/\varpi^2 < \Omega_N$ (Corollary 2 in Appendix A.1). We present a convergence proof soon in Section 4.1.1.

(LemniAcc) is not the first method to attain an $O(L^2 \|x_0 - x_\star\|^2 / N^4)$ rate on the squared gradient norm $\|\nabla f(x_N)\|^2$ of L-smooth convex functions. Nesterov et al. [44, Remark 2.1] observed that running Nesterov's accelerated gradient method (AGM) [43] (or Kim–Fessler's optimized gradient method (OGM) [30]) for the first half of the iterations and OGM-G [31] for the second half achieves this rate, which matches the $\Omega(L^2 \|x_0 - x_\star\|^2 / N^4)$ lower bound of [40, 41] and is therefore optimal up to a constant factor.

Lemniscate acceleration improves upon this two-phase scheme in two ways. First, it attains the optimal rate as a single method, with one coefficient sequence and no switch between two algorithms. Second, it improves the constant of the guarantee. For even N, chaining the standard guarantee of OGM with that of OGM-G yields

$$\|\nabla f(x_N)\|^2 \le \frac{64\, L^2 \|x_0 - x_\star\|^2}{(N+2)^4}.$$

In comparison, Theorem 1 attains the constant $\varpi^4 \approx 47.27$, improving the leading asymptotic constant by a factor of approximately 1.35.

4.1.1 Convergence proof

In this section, we prove Theorem 1. Throughout, $f\colon \mathbb{R}^d \to \mathbb{R}$ is an L-smooth convex function with a minimizer $x_\star$, and x_k and z_k denote the iterates of (LemniAcc); in particular $z_0 = 0$. The argument has three steps. We first define the interpolation gaps and, from the multipliers of the analytic dual certificate found in Stage 2 of the pipeline, a Lyapunov sequence $\mathcal{V}_0, \mathcal{V}_1, \dots, \mathcal{V}_N$ along the iterates. We then evaluate the two endpoints: $\mathcal{V}_0$ is at most $(L/(2\Omega_N))\left\|x_0 - x_\star\right\|^2$, and the terminal value $\mathcal{V}_N$ dominates $(\Omega_N/(2L))\left\|\nabla f(x_N)\right\|^2$ (Lemma 2). Finally, Lemma 3 shows that the sequence is nonincreasing along the iterates, and chaining the three facts proves Theorem 1; the same machinery yields a function-value guarantee (Corollary 1).

Write $g_i \triangleq \nabla f(x_i)$ and $g_\star = 0$, so that $x_\star^+ = x_\star$, and denote the interpolation gap by

$$\mathcal{D}_{i,j} \triangleq f(x_i) - f(x_j) - \langle g_j, x_i - x_j\rangle - \frac{1}{2L}\left\|g_i - g_j\right\|^2 \geq 0, \quad i,j \in I_N^\star,$$

which is nonnegative by (6). The three-point identity

$$\mathcal{D}_{i,\star} - \left\langle x_i^+ - x_\star, g_j\right\rangle = \mathcal{D}_{i,j} - \mathcal{D}_{\star,j} \tag{10}$$

follows from direct calculation. For notational simplicity, we also set $\mathcal{D}_{-1,0} \triangleq 0$, $\mathcal{D}_{-1,\star} \triangleq 0$, and $x_{-1}^+ \triangleq x_0$. Evaluating the recurrence (9) at $k = N$, where $\rho_{N+1} = 0$, and at $k = 0$, where $\rho_0 = 1$, gives the two boundary identities $\rho_N = 1/\Omega_N$ and $\Omega_N = (1+\rho_1)/(1-\rho_1) > 1$, which we use repeatedly below.

For $0 \leq k \leq N$, define the Lyapunov sequence

$$\mathcal{V}_k \triangleq \frac{1-\rho_k^2}{2\rho_k}\mathcal{D}_{k-1,\star} - \frac{(1-\rho_k)^2}{2\rho_k}\mathcal{D}_{N,\star} + \frac{L}{2\Omega_N}\left\|x_k - x_\star + \frac{1+\rho_{k+1}^2}{1-\rho_{k+1}^2}z_k\right\|^2 - \frac{L}{2\Omega_N}\left\|x_N^+ - x_\star + z_k\right\|^2. \tag{11}$$

At $k = 0$, since $\mathcal{D}_{-1,\star} = 0$, $\rho_0 = 1$, and $z_0 = 0$, we get

$$\mathcal{V}_0 = \frac{L}{2\Omega_N}\left\|x_0 - x_\star\right\|^2 - \frac{L}{2\Omega_N}\left\|x_N^+ - x_\star\right\|^2 \leq \frac{L}{2\Omega_N}\left\|x_0 - x_\star\right\|^2.$$

Next, we evaluate the terminal value $\mathcal{V}_N$.

Lemma 2 (Terminal decomposition)**.** *The terminal value of the Lyapunov sequence* (11) *satisfies*

$$\mathcal{V}_N = \frac{\Omega_N}{2L}\left\|\nabla f(x_N)\right\|^2 + \mathcal{D}_{N,\star} + \frac{1}{\Omega_N}\mathcal{D}_{\star,N} + \frac{\Omega_N^2 - 1}{2\Omega_N}\mathcal{D}_{N-1,N} \geq \frac{\Omega_N}{2L}\left\|\nabla f(x_N)\right\|^2.$$

Proof of Lemma 2. Solving the x-update at $k = N-1$, whose coefficient evaluates to $(1+\rho_N^2)/(1-\rho_N^2) - (1+\rho_{N+1}^2)/(1-\rho_{N+1}^2) = 2/(\Omega_N^2 - 1)$, for z_N and adding $x_N - x_\star$ gives

$$x_N - x_\star + z_N = -\frac{\Omega_N^2 - 1}{2}(x_{N-1}^+ - x_\star) + \frac{\Omega_N^2 + 1}{2}(x_N - x_\star). \tag{12}$$

Substituting $\rho_N = 1/\Omega_N$, $\rho_{N+1} = 0$ into the Lyapunov sequence at $k = N$ and expanding,

$$
\begin{aligned}
\mathcal{V}_N &= \frac{\Omega_N^2 - 1}{2\Omega_N}\mathcal{D}_{N-1,\star} - \frac{(\Omega_N - 1)^2}{2\Omega_N}\mathcal{D}_{N,\star} + \frac{L}{2\Omega_N}\left\|x_N - x_\star + z_N\right\|^2 - \frac{L}{2\Omega_N}\left\|x_N^+ - x_\star + z_N\right\|^2 \\
&= \frac{\Omega_N^2 - 1}{2\Omega_N}\mathcal{D}_{N-1,\star} - \frac{(\Omega_N - 1)^2}{2\Omega_N}\mathcal{D}_{N,\star} + \frac{1}{\Omega_N}\left\langle x_N - x_\star + z_N, g_N\right\rangle - \frac{1}{2L\Omega_N}\left\|g_N\right\|^2 \\
&= \frac{\Omega_N^2 - 1}{2\Omega_N}\left(\mathcal{D}_{N-1,\star} - \left\langle x_{N-1}^+ - x_\star, g_N\right\rangle\right) - \frac{(\Omega_N - 1)^2}{2\Omega_N}\mathcal{D}_{N,\star} \\
&\quad + \frac{\Omega_N^2 + 1}{2\Omega_N}\left\langle x_N - x_\star, g_N\right\rangle - \frac{1}{2L\Omega_N}\left\|g_N\right\|^2 \\
&= \frac{\Omega_N^2 - 1}{2\Omega_N}\left(\mathcal{D}_{N-1,N} - \mathcal{D}_{\star,N}\right) - \frac{(\Omega_N - 1)^2}{2\Omega_N}\mathcal{D}_{N,\star} \\
&\quad + \frac{\Omega_N^2 + 1}{2\Omega_N}\left\langle x_N^+ - x_\star, g_N\right\rangle + \frac{\Omega_N}{2L}\left\|g_N\right\|^2 \\
&= \frac{\Omega_N}{2L}\left\|g_N\right\|^2 + \mathcal{D}_{N,\star} + \frac{1}{\Omega_N}\mathcal{D}_{\star,N} + \frac{\Omega_N^2 - 1}{2\Omega_N}\mathcal{D}_{N-1,N},
\end{aligned}
$$

where the second equality expands the difference of the two squared norms, the third substitutes (12) and groups the terms by their coefficients, the fourth applies (10) at $(i, j) = (N-1, N)$ and substitutes $x_N = x_N^+ + (1/L)\, g_N$, and the last applies (10) at $(i, j) = (N, N)$ in the form $\left\langle x_N^+ - x_\star, g_N\right\rangle = \mathcal{D}_{N,\star} + \mathcal{D}_{\star,N}$ using $\mathcal{D}_{N,N} = 0$, and collects terms. The three interpolation gaps are nonnegative and their multipliers are positive by $\Omega_N = (1+\rho_1)/(1-\rho_1) > 1$, which gives the second claim. $\square$

Lemma 3 (Monotonicity)**.** *The Lyapunov sequence* (11)*, evaluated on the iterates of* (LemniAcc)*, satisfies for* $0 \le k \le N-1$

$$
\mathcal{V}_k - \mathcal{V}_{k+1} = \frac{1-\rho_k^2}{2\rho_k}\mathcal{D}_{k-1,k} + (\rho_k - \rho_{k+1})\mathcal{D}_{\star,k} + \left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\right)\mathcal{D}_{N,k} \ge 0. \tag{13}
$$

Consequently, $\mathcal{V}_0 \ge \mathcal{V}_1 \ge \cdots \ge \mathcal{V}_N$*.*

Proof. We calculate the difference $\mathcal{V}_k - \mathcal{V}_{k+1}$ and group the resulting terms by the three-point identity (10); we first prepare the supporting identities.

We use the following two scalar identities. Rearranging the recurrence (9) gives for $0 \le k \le N-1$,

$$
\begin{aligned}
&\Omega_N\left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\right)\frac{1+\rho_{k+1}^2}{1-\rho_{k+1}^2} + 1 = \Omega_N\left(\frac{1-\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1-\rho_k^2}{2\rho_k}\right), \\
&\frac{\Omega_N}{2}\left[\left(\frac{1-\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1-\rho_k^2}{2\rho_k}\right)^2 - \left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\right)^2\right] = \frac{1-\rho_{k+1}^2}{2\rho_{k+1}}.
\end{aligned} \tag{14}
$$

The update equations of (LemniAcc) give the two one-step relations

$$\begin{aligned} x_{k+1} - x_\star + \frac{1+\rho_{k+2}^2}{1-\rho_{k+2}^2} z_{k+1} &= x_k - x_\star + \frac{1+\rho_{k+1}^2}{1-\rho_{k+1}^2} z_k - \frac{\Omega_N}{L}\left(\frac{1-\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1-\rho_k^2}{2\rho_k}\right) g_k, \\ x_N^+ - x_\star + z_{k+1} &= x_N^+ - x_\star + z_k - \frac{\Omega_N}{L}\left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\right) g_k, \end{aligned} \tag{15}$$

both for $0 \le k \le N-1$. For the first relation, the x-update gives

$$x_{k+1} - x_\star + \frac{1+\rho_{k+2}^2}{1-\rho_{k+2}^2} z_{k+1} = x_k - x_\star - \frac{1}{L} g_k + \frac{1+\rho_{k+1}^2}{1-\rho_{k+1}^2} z_{k+1},$$

and substituting the z-update for z_{k+1} on the right-hand side and evaluating the resulting coefficient of g_k by the first line of (14) proves it. The second relation of (15) follows by adding $x_N^+ - x_\star$ to both sides of the z-update.

We also state a useful vector identity: for $0 \le k \le N-1$,

$$\begin{aligned} &\left(\frac{1-\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1-\rho_k^2}{2\rho_k}\right)\left(x_k - x_\star + \frac{1+\rho_{k+1}^2}{1-\rho_{k+1}^2} z_k\right) - \left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\right)(x_N^+ - x_\star + z_k) \\ &= -\frac{1-\rho_k^2}{2\rho_k}(x_{k-1}^+ - x_\star) + \frac{1-\rho_{k+1}^2}{2\rho_{k+1}}(x_k - x_\star) - \left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\right)(x_N^+ - x_\star). \end{aligned} \tag{16}$$

At $k=0$, both sides equal $\left((1-\rho_1^2)/(2\rho_1)\right)(x_0 - x_\star) - \left(\left((1+\rho_1^2)/(2\rho_1)\right) - 1\right)(x_N^+ - x_\star)$, by $z_0 = 0$, $(1-\rho_0^2)/(2\rho_0) = 0$, and $x_{-1}^+ = x_0$. For $1 \le k \le N-1$, distribute the two products on the left-hand side: the coefficient of z_k collects to

$$\left(\frac{1-\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1-\rho_k^2}{2\rho_k}\right)\frac{1+\rho_{k+1}^2}{1-\rho_{k+1}^2} - \left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\right) = \frac{1-\rho_k^2}{2\rho_k}\left(\frac{1+\rho_k^2}{1-\rho_k^2} - \frac{1+\rho_{k+1}^2}{1-\rho_{k+1}^2}\right),$$

and the x-update at index $k-1$ gives

$$\frac{1-\rho_k^2}{2\rho_k}\left(\frac{1+\rho_k^2}{1-\rho_k^2} - \frac{1+\rho_{k+1}^2}{1-\rho_{k+1}^2}\right) z_k = \frac{1-\rho_k^2}{2\rho_k}\left(x_k - x_{k-1}^+\right).$$

Substituting this and splitting $x_k - x_{k-1}^+ = (x_k - x_\star) - (x_{k-1}^+ - x_\star)$ verifies (16).

Fix $0 \le k \le N-1$. Expanding the difference $\mathcal{V}_k - \mathcal{V}_{k+1}$ using the one-step relations (15),

$$
\begin{aligned}
\mathcal{V}_k - \mathcal{V}_{k+1} &= \frac{1-\rho_k^2}{2\rho_k}\mathcal{D}_{k-1,\star} - \frac{1-\rho_{k+1}^2}{2\rho_{k+1}}\mathcal{D}_{k,\star} + \left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\right)\mathcal{D}_{N,\star} \\
&\quad + \left(\frac{1-\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1-\rho_k^2}{2\rho_k}\right) \cdot \left\langle x_k - x_\star + \frac{1+\rho_{k+1}^2}{1-\rho_{k+1}^2} z_k, g_k \right\rangle \\
&\quad - \left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\right) \cdot \langle x_N^+ - x_\star + z_k, g_k \rangle \\
&\quad - \frac{\Omega_N}{2L}\left[\left(\frac{1-\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1-\rho_k^2}{2\rho_k}\right)^2 - \left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\right)^2\right]\|g_k\|^2 \\
&= \frac{1-\rho_k^2}{2\rho_k}\left(\mathcal{D}_{k-1,\star} - \langle x_{k-1}^+ - x_\star, g_k\rangle\right) \\
&\quad - \frac{1-\rho_{k+1}^2}{2\rho_{k+1}}\left(\mathcal{D}_{k,\star} - \langle x_k - x_\star, g_k\rangle + \frac{1}{L}\|g_k\|^2\right) \\
&\quad + \left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\right)\left(\mathcal{D}_{N,\star} - \langle x_N^+ - x_\star, g_k\rangle\right) \\
&= \frac{1-\rho_k^2}{2\rho_k}\left(\mathcal{D}_{k-1,k} - \mathcal{D}_{\star,k}\right) - \frac{1-\rho_{k+1}^2}{2\rho_{k+1}}\left(\mathcal{D}_{k,k} - \mathcal{D}_{\star,k}\right) + \left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\right)\left(\mathcal{D}_{N,k} - \mathcal{D}_{\star,k}\right) \\
&= \frac{1-\rho_k^2}{2\rho_k}\mathcal{D}_{k-1,k} + (\rho_k - \rho_{k+1})\mathcal{D}_{\star,k} + \left(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\right)\mathcal{D}_{N,k},
\end{aligned}
$$

where the second equality substitutes the vector identity (16) and the second line of (14), the third applies (10) at $j = k$ to each of the three groups, where at $k = 0$ the first group vanishes since its factor $(1-\rho_0^2)/(2\rho_0)$ is zero, and the fourth uses $\mathcal{D}_{k,k} = 0$. This proves the equality in (13).

Finally, the right-hand side of (13) is nonnegative: the interpolation gaps are nonnegative, and the three multipliers are nonnegative, since Lemma 1 gives $1 = \rho_0 > \rho_1 > \cdots > \rho_{N+1} = 0$ and since the map $\rho \mapsto (1+\rho^2)/(2\rho)$ is strictly decreasing for $0 < \rho \le 1$, so that $(1+\rho_{k+1}^2)/(2\rho_{k+1}) > (1+\rho_k^2)/(2\rho_k)$ for $0 \le k \le N-1$. □

Combining Lemma 2 and Lemma 3, the proof of Theorem 1 is now immediate.

Proof of Theorem 1. Combining Lemma 2, the monotonicity of Lemma 3, and the evaluation of $\mathcal{V}_0$ gives

$$\frac{\Omega_N}{2L}\|\nabla f(x_N)\|^2 \le \mathcal{V}_N \le \mathcal{V}_{N-1} \le \cdots \le \mathcal{V}_1 \le \mathcal{V}_0 \le \frac{L}{2\Omega_N}\|x_0 - x_\star\|^2,$$

and multiplying through by $2L/\Omega_N$ yields the first inequality; the second inequality then follows from the lower bound $\Omega_N > (N+1)^2/\varpi^2$ of Corollary 2. □

The Lyapunov sequence also yields a guarantee on the function value of the final iterate.

Corollary 1 (Function-value bound)**.** *Under the assumptions of Theorem 1, the final iterate x_N of* (LemniAcc) *satisfies*

$$f(x_N) - f(x_\star) \le \frac{L}{2\Omega_N} \|x_0 - x_\star\|^2 \le \frac{\varpi^2 L \|x_0 - x_\star\|^2}{2(N+1)^2}.$$

Proof. Dropping the two nonnegative terms $(1/\Omega_N)\mathcal{D}_{\star,N}$ and $\left((\Omega_N^2 - 1)/(2\Omega_N)\right)\mathcal{D}_{N-1,N}$ in Lemma 2 and expanding $\mathcal{D}_{N,\star}$,

$$\begin{aligned}\mathcal{V}_N &\ge \frac{\Omega_N}{2L} \|\nabla f(x_N)\|^2 + \mathcal{D}_{N,\star} \\ &= f(x_N) - f(x_\star) + \frac{\Omega_N - 1}{2L} \|\nabla f(x_N)\|^2 \ge f(x_N) - f(x_\star),\end{aligned}$$

where the last inequality uses $\Omega_N = (1+\rho_1)/(1-\rho_1) > 1$. Combining with $\mathcal{V}_N \le \mathcal{V}_0 \le (L/(2\Omega_N)) \|x_0 - x_\star\|^2$ proves the first inequality; the second follows from $(N+1)^2/\varpi^2 < \Omega_N$ (Corollary 2), as in the proof of Theorem 1. □

4.1.2 AutoOPT design: Optimal method for reducing the gradient magnitude from an initial-distance bound

The method (LemniAcc) was designed by AutoOPT as the optimal fixed-step first-order method such that for all L-smooth convex functions f with minimizer $x_\star$, the guarantee

$$\|\nabla f(x_N)\|^2 \le C_N \|x_0 - x_\star\|^2$$

is optimal, i.e., C_N is minimized. Specifically, among all N-step fixed-step first-order methods of the form

$$x_i = x_{i-1} - \frac{1}{L} \sum_{j=0}^{i-1} h_{i,j} \nabla f(x_j), \qquad i = 1, \dots, N,$$

AutoOPT jointly searches over the coefficients $\{h_{i,j}\}_{0 \le j < i \le N}$ and a convergence certificate to minimize C_N. In other words, AutoOPT jointly optimizes over the algorithm and its convergence proof to find the algorithm with the optimal (smallest) convergence guarantee.

In Stage 1 of the pipeline, the skill bnb-pep-skill wrote out the BnB-PEP formulation of this design problem. At a high level, performance estimation programming replaces the maximization over the infinite-dimensional function class $\mathcal{F}_{0,L}$ by a finite-dimensional semidefinite program involving only the function values, gradients, and pairwise inner products observed along the trajectory; dualizing this inner semidefinite program and jointly optimizing its dual variables and the method coefficients produces a nonconvex quadratically constrained quadratic program. Upon our approval of the formulation, reproduced with light editing in Appendix A.3, the skill solved this program for the horizons $N = 1, 2, 3, \dots, 25$, running spatial branch-and-bound at our request to certify global optimality for $N = 1, \dots, 5$ and local optimality for $N = 6, \dots, 25$, and returned the corresponding candidate stepsizes together with the candidate dual certificates.

These numerical solutions, together with the Stage-1 derivation, became the context for Stage 2, in which the skill frontier-llm-consult placed them before a frontier LLM. The consultation first identified an analytic description valid for arbitrary $N \in \mathbb{N}$, the parametrization in terms of Ω_N

and the sequence $\rho_0, \dots, \rho_{N+1}$, together with an analytic feasible point of the dual certifying the guarantee. At this point the method was still written as the dense stepsize array $\{h_{i,j}\}$, its entries now in closed form; the consultation next recast it into the memory-efficient two-sequence form (LemniAcc) in which Section 4.1 states the method. Continuing the same conversation, Stage 2 then converted the multipliers of this certificate into the Lyapunov proof of Theorem 1 presented in Section 4.1.1, a proof stated directly on this two-sequence form; its Lean formalization is discussed in Section 4.1.5.

4.1.3 Background: Lemniscate constant and lemniscate elliptic functions

The lemniscate constant $\varpi \in \mathbb{R}$ is a transcendental number [70] defined by

$$\varpi = 2\int_0^1 \frac{1}{\sqrt{1-x^4}}\,dx \approx 2.62205755.$$

It is also characterized as the ratio of perimeter to diameter of the unit lemniscate. See Figure 6.

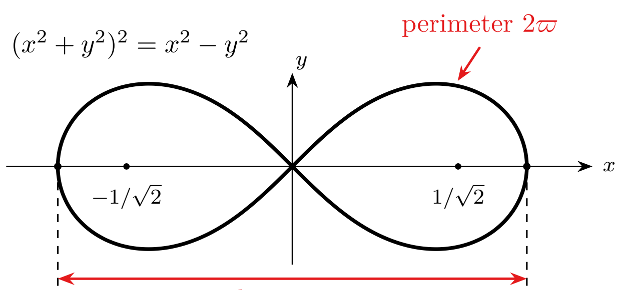

Figure 6: The unit lemniscate $(x^2+y^2)^2 = x^2 - y^2$ with foci at $(\pm 1/\sqrt{2}, 0)$ and tips at $(\pm 1, 0)$. Its perimeter is 2ϖ, and its diameter is 2. Thus $\varpi \approx 2.62206$ is its perimeter-to-diameter ratio.

The lemniscate elliptic functions are defined through the arc-length integral of the lemniscate. Specifically, let

$$\operatorname{arcsl}(u) = \int_0^u \frac{dx}{\sqrt{1-x^4}},$$

for $u \in [0,1]$ and define the lemniscate sine $\operatorname{sl}(x)$ on $[0, \varpi/2]$ as the inverse of arcsl. The lemniscate cosine $\operatorname{cl}(x)$ is defined on $[0, \varpi/2]$ by

$$\operatorname{cl}(x) = \operatorname{sl}\left(\frac{\varpi}{2} - x\right).$$

Finally, we briefly note that while Ω_N is an algebraic number, since a finite optimal value of an QCQP with rational coefficients, the limit $N^2/\Omega_N \to \varpi^2$ (see Lemma 8 in Appendix A.1) is transcendental.

4.1.4 Continuous-time analysis

In this section, we present the continuous time limit of lemniscate acceleration as an ODE, and state an analogous convergence theorem. The details for this section can be found in Appendix A.4.

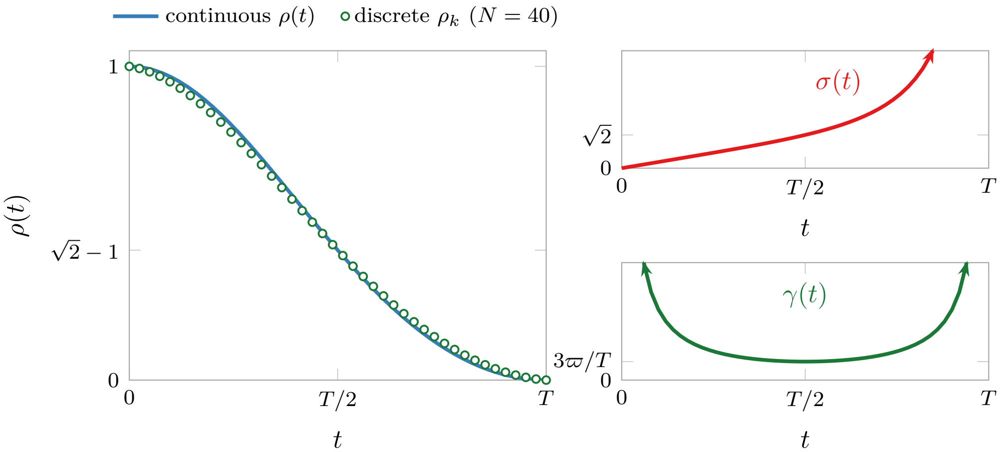

Figure 7: Continuous-time coefficient profiles. The left panel shows $\rho(t) = \mathrm{cl}^2(\varpi t/(2T))$ together with the discrete coefficients ρ_k for $N = 40$, placed at $t_k = kT/(N+1)$. We see that $\rho(t)$ is the continuous limit of the discrete coefficients. The right panels show $\sigma(t) = \sqrt{1/\rho(t) - \rho(t)}$ and $\gamma(t) = 3\dot{\sigma}(t)/\sigma(t)$. The arrows indicate endpoint divergences to $+\infty$.

Continuous-time ODE. For a finite time interval $[0, T]$, define ρ on $[0, T]$ and σ on $[0, T)$ by

$$\rho(t) \triangleq \mathrm{cl}^2\left(\frac{\varpi t}{2T}\right),$$
$$\sigma(t) \triangleq \sqrt{1/\rho(t) - \rho(t)} = \sqrt{\mathrm{cl}^{-2}\left(\varpi t/(2T)\right) - \mathrm{cl}^2\left(\varpi t/(2T)\right)}$$

where ϖ is the lemniscate constant. Figure 7 shows these profiles and the discrete coefficients ρ_k approaching the continuous curve $\rho(t)$.

Taking the small-stepsize limit of (LemniAcc) under the quantization $t = kT/N$ and $L = N^2/T^2$ with $N \to \infty$ suggests that the continuous-time counterparts X and Z of x_k and $\big(N/(\sqrt{\Omega_N}\,T)\big)z_k$, respectively, satisfy the ODE

$$\dot{Z}(t) = -\frac{\sigma(t)^3}{2}\nabla f(X(t)),$$
$$\dot{X}(t) = \frac{4}{\sigma(t)^3}Z(t)$$

for $t \in (0, T)$, with initial conditions $X(0) = x_0$ and $\dot{X}(0) = 0$. Eliminating $Z(t)$ and defining the friction coefficient $\gamma(t) \triangleq 3\dot{\sigma}(t)/\sigma(t)$ yields the second-order ODE

$$\ddot{X} + \gamma(t)\dot{X} + 2\nabla f(X) = 0,$$

which, in terms of the lemniscate functions cl and sl, has the form

$$\ddot{X}(t) + \underbrace{\frac{3\varpi}{2T}\left[\frac{\mathrm{cl}\left(\frac{\varpi t}{2T}\right)}{\mathrm{sl}\left(\frac{\varpi t}{2T}\right)} + \frac{\mathrm{sl}\left(\frac{\varpi t}{2T}\right)}{\mathrm{cl}\left(\frac{\varpi t}{2T}\right)}\right]}_{=\gamma(t)}\dot{X}(t) + 2\nabla f\left(X(t)\right) = 0$$

for $t \in (0,T)$. The endpoint expansions of γ give $\gamma(t) \sim 3/t$ as $t \downarrow 0$ and $\gamma(t) \sim 3/(T-t)$ as $t \uparrow T$ so near $t=0$ the ODE asymptotically matches the continuous-time limit of OGM and near $t=T$ that of OGM-G [58].

We present an analogous Lyapunov analysis for the continuous-time ODE.

Theorem 2. *Fix $T>0$, and let $f\colon \mathbb{R}^d \to \mathbb{R}$ be an L-smooth convex function with a minimizer $x_\star$. Any trajectory $X \in C^2([0,T];\mathbb{R}^d)$ satisfying $\ddot{X} + \gamma(t)\dot{X} + 2\nabla f(X) = 0$ on $(0,T)$, with $X(0) = x_0$ and $\dot{X}(0) = 0$, has the following bounds on the endpoint $X(T)$:*

$$\begin{aligned}
\|\nabla f(X(T))\|^2 &\leq \frac{\varpi^4}{T^4}\|x_0 - x_\star\|^2, \\
f(X(T)) - f(x_\star) &\leq \frac{\varpi^2}{2T^2}\|x_0 - x_\star\|^2.
\end{aligned}$$

Proof sketch. We denote $x_T \triangleq X(T)$, set $Z(t) \triangleq \sigma(t)^3\dot{X}(t)/4$ for $t \in (0,T)$, and define the continuous-time Lyapunov function $\mathcal{V}(t)$ on $(0,T)$ by

$$\begin{aligned}
\mathcal{V}(t) \triangleq{}& \frac{1-\rho^2}{2\rho}\left(f(X(t)) - f(x_\star)\right) - \frac{(1-\rho)^2}{2\rho}\left(f(x_T) - f(x_\star)\right) \\
&+ \frac{\varpi^2}{2T^2}\left(\left\|X(t) - x_\star + \frac{1+\rho^2}{1-\rho^2}\frac{T}{\varpi}Z(t)\right\|^2 - \left\|x_T - x_\star + \frac{T}{\varpi}Z(t)\right\|^2\right).
\end{aligned}$$

The derivative can be expressed as the following sum of Bregman divergences $D_f(x,y) \triangleq f(x) - f(y) - \langle x-y, \nabla f(y)\rangle$, thus is nonpositive:

$$\frac{d}{dt}\mathcal{V}(t) = -\frac{\varpi}{T}\left(\sigma\rho D_f\left(x_\star, X(t)\right) + \frac{1}{2}\sigma^3 D_f\left(x_T, X(t)\right)\right) \leq 0.$$

The limits of the endpoints $t \downarrow 0$, $t \uparrow T$ are

$$\begin{aligned}
\mathcal{V}(0+) &= \frac{\varpi^2}{2T^2}\left(\|x_0 - x_\star\|^2 - \|x_T - x_\star\|^2\right), \\
\mathcal{V}(T-) &= \frac{T^2}{2\varpi^2}\|\nabla f(x_T)\|^2 + f(x_T) - f(x_\star),
\end{aligned}$$

and the bounds follow by chaining $\left(\varpi^2/(2T^2)\right)\|x_0 - x_\star\|^2 \geq \mathcal{V}(0+) \geq \mathcal{V}(T-)$ and $\mathcal{V}(T-) \geq \left(T^2/(2\varpi^2)\right)\|\nabla f(x_T)\|^2$ for the gradient bound and $\mathcal{V}(T-) \geq f(x_T) - f(x_\star)$ for the function value bound. □

4.1.5 Lean formalization

The convergence theory of this section is machine-checked in Lean 4 with mathlib [16, 66] by Stage 3 of AutoOPT (Section 3.3); Figure 8 shows an illustrative Stage-3 prompt. The resulting Lean project proves ten public theorem declarations, stating the smooth convex inequality over the Euclidean space $\mathbb{R}^d$ and the algorithmic and continuous-time results over an arbitrary complete real inner-product space, which subsumes the manuscript's $\mathbb{R}^d$ setting. The declaration `LemniAcc.gradient_rate` is the Lean counterpart of Theorem 1, proving both displayed bounds;

```
Using the lean-verify skill, formalize and verify the lemniscate acceleration results that
are located in the ResearchLog folder. Targets for the verification are the coefficient
construction, the Lyapunov identities, the convergence theorem, the function-value bound,
the continuous-time theorem. State the results in the manuscript's Euclidean setting or a
generalization of it. Constraints are Lean 4.32.0 with mathlib 4.32.0, pinned by commit; no
sorry, axiom, admit, or unsafe anywhere in the development.
Deliverables are a Lake project passing lake build and the hygiene scan; wrappers Challenge.le
an, Solution.lean, and config.json naming the agreed public theorem surface for comparator
replay; build, axiom-audit, and replay logs under artifacts/.
Please ask me clarifying questions if anything is not clear to you.
```

Figure 8: An illustrative Stage-3 prompt for the formal verification of the lemniscate accelerated gradient method using the lean-verify skill.

Public theorem declarations	10
Dependency nodes in the closure	22
Lean source files	22
Lines of Lean	7,384
`lake build` jobs, native macOS	8,687
Comparator replay time, native macOS	109 s
Comparator replay time, WSL2 real Landrun	155 s
Toolchain	Lean 4.32.0, mathlib 4.32.0, commit-pinned
Axioms used	`propext`, `Quot.sound`, `Classical.choice`

Table 2: Summary statistics of the lemniscate acceleration Lean project: it formalizes a fixed 22-node dependency closure of manuscript claims and their supporting steps through ten public theorem declarations, builds cleanly under the commit-pinned toolchain, and is accepted by the comparator replay under exactly the three standard axioms listed.

`LemniAcc.functionValue_rate` proves both bounds of Corollary 1; and `LemniAcc.continuousTime_lyapunov` proves both endpoint bounds of Theorem 2, with the C^2 trajectory of the ODE supplied as a hypothesis exactly as stated there. The supporting declarations follow the paper's proofs: `recurrence_existsUnique` is Lemma 1; `omega_bounds` gives the strict bounds $(N+1)^2/\varpi^2 < \Omega_N < (N+1)^2$ of Corollary 2; `finite_interpolation` establishes (6) that the proofs use; `iterates` constructs the trajectory of (LemniAcc); `lyapunov_terminal` and `lyapunov_decrement` are Lemmas 2 and 3; and `Lemniscatic.sigma_identities` combines the lemniscatic calculus of Lemma 10 with the σ-identities of Lemma 11. As a representative statement, `LemniAcc.gradient_rate` reads as follows, where `E` is an arbitrary complete real inner-product space:

- **LemniAcc/** Lean/Lake project for lemniscate acceleration
  - **lakefile.toml, lean-toolchain** build configuration; commit-pinned Lean and mathlib
  - **config.json** pins the ten theorem names and permitted axioms for the comparator
  - **Challenge.lean** comparator challenge wrapper over the stated development
  - **Solution.lean** comparator solution wrapper, replayed against it
  - **AxiomAudit.lean** runs the axiom audit over the ten declarations
  - **LemniAcc.lean** umbrella module importing the development
  - **LemniAcc/** proof modules, organized in dependency layers:
    - **Model, FiniteInterpolation** problem class, oracle data, interpolation inequality
    - **Discrete/Recurrence/** coefficient existence and uniqueness by shooting
    - **Discrete/** iterates, Lyapunov identities, discrete rate theorems
    - **Lemniscatic/** the lemniscate integral and lemniscatic calculus
    - **Continuous/** the continuous-time Lyapunov development
  - **artifacts/** build, hygiene, axiom-audit, and comparator replay logs

Figure 9: File structure of the Lean project for the lemniscate accelerated gradient method (module files grouped by folder). The wrappers `Challenge.lean`, `Solution.lean`, and `config.json` pin the ten public theorems and the permitted axioms for the comparator, and `artifacts/` records the build, audit, and replay logs.

```
theorem gradient_rate
    (N : Nat) (hN : 1 ≤ N)
    (M : SmoothConvexModel E) (x0 xStar : E)
    (hxStar : M.IsMinimizer xStar) :
    ‖M.grad (Discrete.canonicalX N M x0 N)‖ ^ 2 ≤
        ((M.L : ℝ) ^ 2 / omega N ^ 2) * ‖x0 - xStar‖ ^ 2
      ∧
      ((M.L : ℝ) ^ 2 / omega N ^ 2) * ‖x0 - xStar‖ ^ 2 ≤
        (Lemniscatic.varpi ^ 4 * (M.L : ℝ) ^ 2 *
            ‖x0 - xStar‖ ^ 2) /
          (((N + 1 : Nat) : ℝ) ^ 4)
```

Figure 9 shows the layout of the project, and Table 2 reports its summary statistics. The project comprises 22 Lean source files and 7,384 lines of Lean, organized in dependency layers that parallel the paper: the problem class and the interpolation inequality, the recurrence construction, the discrete Lyapunov development, and the lemniscatic calculus supporting the continuous-time theorem. It builds on mathlib's theories of inner-product and Euclidean spaces, differentiability, and one-variable calculus; in particular, the shooting argument of Appendix A.1 rests on the intermediate value theorem, and the lemniscatic calculus of Appendix A.4 on interval integration.

The project passes `lake build` under the commit-pinned toolchain Lean 4.32.0 with mathlib 4.32.0 (8,687 build jobs on native macOS), a hygiene scan confirms that no `sorry`, `axiom`, `admit`, or `unsafe` appears anywhere in the development, and an axiom audit reports that every public declaration depends on exactly the three standard axioms `propext`, `Quot.sound`, and `Classical.choice`. The comparator then independently certified the ten public theorems: the approved wrappers `Challenge.lean`, `Solution.lean`, and `config.json` were replayed end to end, rebuilding both wrappers, exporting their environments, and re-checking the solution with the Lean default kernel, in 109 seconds; in this first project both wrappers import the proof development itself, and the ITEM-f project of Section 4.2.4 strengthens the wrapper design with a proof-free statement layer. Each project was certified twice, first replayed on native macOS, and then in a Linux environment with Landlock isolation and systemd restrictions. The project was accepted by the Lean-kernel of the sandboxed environment in 155 seconds, thus carries the evidence label `Lean+comparatorverified(WSL2realLandrun;systemdaddress-familyrestriction)`. The projects and their verification artifacts accompany the paper.

4.2 ITEM-f for efficient function-value ratio contraction

Consider the unconstrained minimization problem

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x), \tag{$\mathcal{P}'$}$$

where $f \colon \mathbb{R}^d \to \mathbb{R}$ is an L-smooth and μ-strongly convex function with $0 < \mu < L$. Strong convexity guarantees that f has a unique minimizer $x_\star$. Denote $f_\star \triangleq f(x_\star)$.

Algorithm statement

The method (ITEM-f) is

$$x_{k+1} = x_k^+ + \frac{a_k \phi_{N-k-1}}{(1-q) a_{k+1} \phi_{N-k}} (x_k^+ - x_{k-1}^+) + \frac{\phi_{N-k-1}}{\phi_{N-k}} (x_k^+ - x_k), \tag{ITEM-f}$$

for $k = 0, 1, 2, \ldots, N-1$, where N is a prespecified iteration count, $x_0 \in \mathbb{R}^d$ is the starting point, $x_{-1}^+ = x_0$ by convention, $x_k^+ = x_k - \frac{1}{L} \nabla f(x_k)$, $q = \mu/L$, $\phi_0 = \sqrt{1-q}$, $\phi_N = ((1-q)\Upsilon_N + a_N) / \sqrt{1-q}$, $\phi_k = \sqrt{2\Upsilon_N a_k - q}$ for $1 \leq k \leq N-1$ and $a_0, a_1, \ldots, a_N$ and Υ_N are positive constants that we define in Lemma 4.

Taylor and Drori [62, Appendix E] reported numerically optimized step sizes of ITEM-f for $N = 1, \ldots, 5$ in the setting $L = 1$ and $\mu = 0.1$, but did not assign a name to the resulting method. They also derived a closed-form method with a convergence analysis, called the Information-Theoretic Exact Method (ITEM), for a related performance criterion. We use the name *ITEM-f* to acknowledge their numerical construction while distinguishing it from ITEM. The optimization objectives underlying ITEM and ITEM-f are discussed further in Section 4.2.2.

Construction of the coefficients

To fully define ITEM-f, it remains to specify the coefficients $a_0, a_1, \ldots, a_N$ and Υ_N. Lemma 4 formally defines these coefficients, and Figure 10 illustrates the underlying geometric construction.

Lemma 4. *For any integer $N \geq 1$ and $0 < q < 1$, there exist a unique real $\Upsilon_N > 1$, unique reals*

$$1/\Upsilon_N = a_0 < a_1 < \cdots < a_N < a_{N+1} = \Upsilon_N,$$

and unique positive reals $b_0, b_1, \ldots, b_N, b_{N+1}$ satisfying the following conditions. Write $C = (\Upsilon_N, 0)$

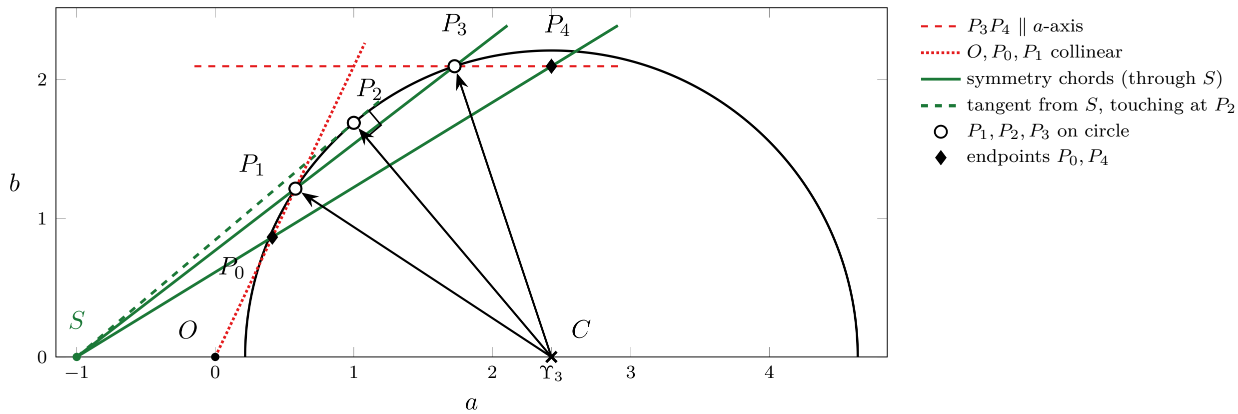

Figure 10: The planar construction underlying Lemma 4, shown for $N = 3$, $q = 0.1$. The intermediate points $P_1, \ldots, P_N$ ($\circ$) lie on the circle of radius $R_N = \sqrt{\Upsilon_N^2 - 1}$ centered at $C = (\Upsilon_N, 0)$; the endpoints P_0 and P_{N+1} ($\diamond$) lie strictly inside; successive points satisfy $(P_k - C) \cdot (P_{k+1} - C) = R_N^2(1 - q\, a_{k+1}/\Upsilon_N)$. *Shooting (red).* The construction is run backward from the horizontal final edge $b_N = b_{N+1} = \sqrt{1-q}\, R_N$ (dashed); Υ_N is the value for which O, P_0, P_1 are collinear in this order (dotted). *Symmetry (green).* The map $T(a, b) = (1/a, b/a)$ satisfies $T(P_k) = P_{N+1-k}$; hence every chord $P_k P_{N+1-k}$ passes through $S = (-1, 0)$. For odd N, the tangent from S touches the circle at the fixed point $P_{(N+1)/2}$.

and $R_N = \sqrt{\Upsilon_N^2 - 1}$ *and write* $P_k = (a_k, b_k) \in \mathbb{R}^2$ *for* $0 \le k \le N + 1$. *The requirements are:*

$$\begin{aligned}
&\|P_k - C\|^2 = R_N^2, \quad 1 \le k \le N,\\
&(P_k - C) \cdot (P_{k+1} - C) = R_N^2\Big(1 - \tfrac{q\, a_{k+1}}{\Upsilon_N}\Big), \quad 0 \le k \le N,\\
&P_0 = \Big(\tfrac{1}{\Upsilon_N}, \tfrac{\sqrt{1-q}\, R_N}{\Upsilon_N}\Big),\\
&P_{N+1} = \big(\Upsilon_N, \sqrt{1-q}\, R_N\big).
\end{aligned} \tag{17}$$

The proof of Lemma 4 is provided in Appendix B.1. To clarify, a_{N+1} is not used in the definition of ITEM-f but we defined it for convenience. We can compute the numerical values of $a_0, \ldots, a_N$ and Υ_N again by a one-dimensional bisection, as discussed in Appendix B.1.

The geometric construction carries a symmetry analogous to the lemniscate recurrence in Section 4.1: the map $T(a, b) \triangleq (1/a,\, b/a)$, defined for a>0, maps the circle $a^2 + b^2 - 2\Upsilon_N a + 1 = 0$ to itself and reverses the construction, $T(P_k) = P_{N+1-k}$ for $0 \le k \le N + 1$, so that $a_k a_{N+1-k} = 1$ and $a_k b_{N+1-k} = b_k$ (Lemma 17 in Appendix B.1); in Figure 10 this symmetry is visible as the chords $P_k P_{N+1-k}$ passing through the point $S \triangleq (-1, 0)$. This symmetry supplies the identities $a_{k+1} a_{N-k} = 1$ and $a_1 a_N = 1$ used in the convergence proof of Section 4.2.1.

Convergence theorem

Theorem 3. *Let* $N \ge 1$ *and* $d \ge 1$ *be integers, and let* $f\colon \mathbb{R}^d \to \mathbb{R}$ *be* L*-smooth and* μ*-strongly convex with* $q = \mu/L \in (0, 1)$*, let* Υ_N *and* $a_0, \ldots, a_N$ *be as in Lemma 4, and let* $x_\star = \operatorname{argmin} f$*. Then, for any starting point* $x_0 \in \mathbb{R}^d$*, the final iterate* x_N *of* (ITEM-f) *satisfies*

$$f(x_N) - f_\star \;\le\; \frac{1}{\Upsilon_N^2}\big(f(x_0) - f_\star\big) \;\le\; 4(1 - \sqrt{q})^{2N}\big(f(x_0) - f_\star\big). \tag{18}$$

We prove Theorem 3 in the following section.

4.2.1 Convergence proof

In this section, we prove Theorem 3. Throughout, $f\colon \mathbb{R}^d \to \mathbb{R}$ is an L-smooth and μ-strongly convex function with the minimizer $x_\star$, $q = \mu/L$, and x_k denotes the iterates of (ITEM-f) with the convention $x_{-1}^+ = x_0$. The argument has three steps. We first define a Lyapunov sequence $\mathcal{V}_1, \ldots, \mathcal{V}_N$ along the iterates, whose coefficients are read off from the multipliers of the analytic dual certificate found in Stage 2 of the pipeline, and compress it into a compact form built from transformed interpolation gaps and vectors $W_k \in \mathbb{R}^{2d}$. We then compare the two endpoints: $\mathcal{V}_1$ is at most $\left(f(x_0) - f_\star\right)/\Upsilon_N$, and $\mathcal{V}_N$ dominates $\Upsilon_N\left(f(x_N) - f_\star\right)$. Finally, Lemma 6 shows that the sequence is nonincreasing, and chaining the three facts proves Theorem 3. The norm identities that drive these computations (Lemma 5) are stated below and proved in Appendix B.2.

For $1 \le k \le N-1$, we define the Lyapunov sequence

$$\begin{aligned}\mathcal{V}_k \triangleq{}& a_k\left(f(x_{k-1}) - f_\star - \frac{\mu}{2}\left\|x_{k-1} - x_\star\right\|^2 - \frac{1}{2(L-\mu)}\left\|\nabla f(x_{k-1}) - \mu(x_{k-1} - x_\star)\right\|^2\right)\\ &+ \frac{\mu(1-q)a_{k+1}^2}{2\Upsilon_N}\left\|x_k - x_\star\right\|^2\\ &+ \frac{\mu}{2\Upsilon_N(1-q)(1-c_k^2)}\left\|(1-q)c_k a_{k+1}(x_k - x_\star) - a_k(x_{k-1}^+ - x_\star)\right\|^2,\end{aligned}$$

and for the final iteration N, define

$$\begin{aligned}\mathcal{V}_N \triangleq{}& a_N\left(f(x_{N-1}) - f_\star - \frac{\mu}{2}\left\|x_{N-1} - x_\star\right\|^2 - \frac{1}{2(L-\mu)}\left\|\nabla f(x_{N-1}) - \mu(x_{N-1} - x_\star)\right\|^2\right)\\ &+ \frac{\mu\Upsilon_N}{2}\left\|x_N - x_\star\right\|^2\\ &+ \frac{L}{2\Upsilon_N(1-q)}\left\|(1-q)\Upsilon_N(x_N - x_\star) - a_N(x_{N-1}^+ - x_\star)\right\|^2,\end{aligned}$$

where we define $c_k \triangleq 1 - (q a_{k+1})/\Upsilon_N$ for $1 \le k \le N$.

Following the approach of [62], we introduce the convex $(L-\mu)$-smooth function and its gradient:

$$\begin{aligned}\widetilde{f}(x) &\triangleq f(x) - f_\star - \frac{\mu}{2}\left\|x - x_\star\right\|^2,\\ \widetilde{g}_k &\triangleq \nabla\widetilde{f}(x_k) = \nabla f(x_k) - \mu(x_k - x_\star).\end{aligned}$$

This makes $\widetilde{g}_\star = 0$ and $\widetilde{f}(x_\star) = 0$. For $i, j \in I_N^\star$, denote its interpolation gap by

$$\mathcal{G}_{i,j} \triangleq \widetilde{f}(x_i) - \widetilde{f}(x_j) - \langle \widetilde{g}_j, x_i - x_j\rangle - \frac{1}{2(L-\mu)}\left\|\widetilde{g}_i - \widetilde{g}_j\right\|^2 \ge 0,$$

which is nonnegative by (6) applied to $\widetilde{f}$. From the relation $x_i^+ - x_\star = (1-q)(x_i - x_\star) - L^{-1}\widetilde{g}_i$ (which follows from $x_i^+ = x_i - L^{-1}\nabla f(x_i)$, $\nabla f(x_i) = \widetilde{g}_i + \mu(x_i - x_\star)$, and $\mu = qL$), the three-point identity follows:

$$\mathcal{G}_{i,\star} - \frac{1}{1-q}\left\langle x_i^+ - x_\star, \widetilde{g}_j\right\rangle = \mathcal{G}_{i,j} - \mathcal{G}_{\star,j}. \tag{19}$$

To simplify the analysis of (ITEM-f), for $1 \leq k \leq N$, we define

$$W_k \triangleq \begin{bmatrix} a_k(x_{k-1}^+ - x_\star) \\ \dfrac{(1-q)a_{k+1}(x_k - x_\star) - c_k a_k(x_{k-1}^+ - x_\star)}{s_k} \end{bmatrix} \in \mathbb{R}^{2d}, \tag{20}$$

where we define $s_k \triangleq \sqrt{q}\, a_{k+1}\phi_{N-k}/\Upsilon_N$. At $k = N$, these definitions give the boundary values $c_N = 1 - q$ and $s_N = \sqrt{q(1-q)}$, using $a_{N+1} = \Upsilon_N$ and $\phi_0 = \sqrt{1-q}$.

For $1 \leq k \leq N-1$, the relation $s_k^2 + c_k^2 = 1$ can be proved using the symmetry identity $a_{k+1}a_{N-k} = 1$ of the construction:

$$s_k^2 + c_k^2 = \frac{q a_{k+1}^2}{\Upsilon_N^2}(2\Upsilon_N a_{N-k} - q) + \left(1 - \frac{q a_{k+1}}{\Upsilon_N}\right)^2 = \frac{q a_{k+1}}{\Upsilon_N^2}(2\Upsilon_N - q a_{k+1}) + \left(1 - \frac{q a_{k+1}}{\Upsilon_N}\right)^2 = 1.$$

For $1 \leq k \leq N$, define

$$\mathcal{O}_k \triangleq \begin{bmatrix} c_k I_d & s_k I_d \\ -s_k I_d & c_k I_d \end{bmatrix} \in \mathbb{R}^{2d \times 2d}.$$

For $1 \leq k \leq N-1$, the identity $s_k^2 + c_k^2 = 1$ makes $\mathcal{O}_k$ orthogonal, and multiplying it by W_k gives

$$\mathcal{O}_k W_k = \begin{bmatrix} (1-q)a_{k+1}(x_k - x_\star) \\ \dfrac{(1-q)c_k a_{k+1}(x_k - x_\star) - a_k(x_{k-1}^+ - x_\star)}{s_k} \end{bmatrix}, \quad 1 \leq k \leq N-1. \tag{21}$$

Since $\mathcal{O}_k$ is orthogonal for $k < N$, this gives the two quadratic terms in $\mathcal{V}_k$. At $k = N$, where $s_N^2 + c_N^2 = 1 - q$ and the terminal block $\mathcal{O}_N$ is not orthogonal, the quadratic terms in the final $\mathcal{V}_N$ follow instead from the direct expansion in the third identity of Lemma 5, using $c_N = 1 - q$, $a_{N+1} = \Upsilon_N$, and $s_N = \sqrt{q(1-q)}$. Thus the Lyapunov sequence has the compact form

$$\mathcal{V}_k = a_k \mathcal{G}_{k-1,\star} + \frac{\mu}{2(1-q)\Upsilon_N}\|W_k\|^2, \quad 1 \leq k \leq N, \tag{22}$$

where the norm is the Euclidean norm on $\mathbb{R}^{2d}$.

Lemma 5 (Norm identities)**.** *The squared norms $\|W_k\|^2$ for the iterates of* (ITEM-f) *satisfy the following identities:*

$$\begin{aligned}
\|W_1\|^2 &= (1-q)\|x_0 - x_\star\|^2 - \frac{2\Upsilon_N}{\mu}(a_1 - a_0)\left\langle x_0^+ - x_\star, \widetilde{g}_0\right\rangle + \frac{\Upsilon_N a_0}{\mu L}\|\widetilde{g}_0\|^2, \\
\|W_{k+1}\|^2 &= \left\| W_k - \frac{\Upsilon_N}{\mu}\begin{bmatrix} (c_k - 1)\widetilde{g}_k \\ s_k \widetilde{g}_k \end{bmatrix} \right\|^2, \quad 1 \leq k \leq N-1, \\
\|W_N\|^2 &= (1-q)\Upsilon_N^2\|x_N - x_\star\|^2 + \frac{1}{q}\left\|(1-q)\Upsilon_N(x_N - x_\star) - a_N(x_{N-1}^+ - x_\star)\right\|^2.
\end{aligned}$$

The proof is tedious but straightforward algebra on the coordinate relations of the construction; it is deferred to Appendix B.2.

Lemma 6 (Monotonicity)**.** *For $1 \le k \le N-1$,*

$$\mathcal{V}_k - \mathcal{V}_{k+1} = a_k \mathcal{G}_{k-1,k} + (a_{k+1} - a_k)\mathcal{G}_{\star,k} \ge 0.$$

Consequently, $\mathcal{V}_1 \ge \mathcal{V}_2 \ge \cdots \ge \mathcal{V}_N$.

Proof. From the definition of W_k and $c_k = 1 - qa_{k+1}/\Upsilon_N$, we have

$$\begin{aligned}
\left\langle \begin{bmatrix} (c_k - 1)\widetilde{g}_k \\ s_k \widetilde{g}_k \end{bmatrix}, W_k \right\rangle_{\mathbb{R}^{2d}} &= \left\langle (1-q)a_{k+1}(x_k - x_\star) - a_k(x_{k-1}^+ - x_\star), \widetilde{g}_k \right\rangle, \\
\left\| \begin{bmatrix} (c_k - 1)\widetilde{g}_k \\ s_k \widetilde{g}_k \end{bmatrix} \right\|^2 &= \left((1-c_k)^2 + s_k^2 \right) \|\widetilde{g}_k\|^2 = 2(1-c_k)\|\widetilde{g}_k\|^2 = \frac{2qa_{k+1}}{\Upsilon_N} \|\widetilde{g}_k\|^2 .
\end{aligned}$$

Using the compact form (22) and expanding the second identity of Lemma 5, we obtain

$$\begin{aligned}
\mathcal{V}_k - \mathcal{V}_{k+1} &= a_k \mathcal{G}_{k-1,\star} - a_{k+1}\mathcal{G}_{k,\star} + \frac{\mu}{2(1-q)\Upsilon_N} \left(\|W_k\|^2 - \|W_{k+1}\|^2 \right) \\
&= a_k \mathcal{G}_{k-1,\star} - a_{k+1}\mathcal{G}_{k,\star} \\
&\qquad + \frac{1}{1-q} \left\langle (1-q)a_{k+1}(x_k - x_\star) - a_k(x_{k-1}^+ - x_\star), \widetilde{g}_k \right\rangle - \frac{a_{k+1}}{L(1-q)} \|\widetilde{g}_k\|^2 \\
&= a_k \left(\mathcal{G}_{k-1,\star} - \frac{1}{1-q} \left\langle x_{k-1}^+ - x_\star, \widetilde{g}_k \right\rangle \right) + a_{k+1}\mathcal{G}_{\star,k} \\
&= a_k \left(\mathcal{G}_{k-1,k} - \mathcal{G}_{\star,k} \right) + a_{k+1}\mathcal{G}_{\star,k} \\
&= a_k \mathcal{G}_{k-1,k} + (a_{k+1} - a_k)\mathcal{G}_{\star,k},
\end{aligned}$$

where the second equality uses the above two identities and the third and fourth use the three-point identity (19) at $(i,j) = (k,k)$ and $(k-1,k)$, respectively. Since $a_{k+1} \ge a_k > 0$ and $\mathcal{G}_{i,j} \ge 0$, the last line is nonnegative. □

We now provide the proof of Theorem 3.

Proof of Theorem 3. The relation $a_0 = 1/\Upsilon_N$, the first identity in Lemma 5, and the compact form (22) of $\mathcal{V}_1$ give

$$\begin{aligned}
\frac{f(x_0) - f_\star}{\Upsilon_N} - \mathcal{V}_1 &= (a_1 - a_0) \left(\frac{1}{1-q} \left\langle x_0^+ - x_\star, \widetilde{g}_0 \right\rangle - \mathcal{G}_{0,\star} \right) \\
&= (a_1 - a_0)\mathcal{G}_{\star,0} \ge 0,
\end{aligned}$$

where the last equality applies (19) at $(i,j) = (0,0)$.

The relation $a_{N+1} = \Upsilon_N$, the third identity in Lemma 5, and the compact form (22) of $\mathcal{V}_N$ give

$$\begin{aligned}
&\mathcal{V}_N - \Upsilon_N\big(f(x_N) - f_\star\big) \\
&\qquad - \frac{L}{2(1-q)\Upsilon_N} \left\| (1-q)\Upsilon_N(x_N - x_\star) - a_N(x_{N-1}^+ - x_\star) - \frac{\Upsilon_N}{L}\widetilde{g}_N \right\|^2
\end{aligned}$$

$$
\begin{aligned}
&= a_N \left(\mathcal{G}_{N-1,\star} - \frac{1}{1-q} \left\langle x_{N-1}^{+} - x_\star, \widetilde{g}_N \right\rangle \right) + \Upsilon_N \mathcal{G}_{\star,N} \\
&= a_N \left(\mathcal{G}_{N-1,N} - \mathcal{G}_{\star,N} \right) + \Upsilon_N \mathcal{G}_{\star,N} \\
&= a_N \mathcal{G}_{N-1,N} + (\Upsilon_N - a_N) \mathcal{G}_{\star,N} \geq 0,
\end{aligned}
$$

where the second equality applies (19) at $(i,j) = (N-1,N)$. The completed square is nonnegative, so $\mathcal{V}_N \geq \Upsilon_N \left(f(x_N) - f_\star \right)$; combining the two endpoint bounds with Lemma 6 gives

$$
\frac{f(x_0) - f_\star}{\Upsilon_N} \geq \mathcal{V}_1 \geq \cdots \geq \mathcal{V}_N \geq \Upsilon_N \big(f(x_N) - f_\star \big).
$$

The explicit relaxation in the second inequality then follows from Lemma 18. $\square$

4.2.2 AutoOPT design: Optimal method for contracting function-value suboptimality

The method (ITEM-f) was designed by AutoOPT as the optimal fixed-step first-order method for contracting the function-value suboptimality. Specifically, AutoOPT minimizes the contraction factor C_N in the guarantee

$$
f(x_N) - f_\star \leq C_N \left(f(x_0) - f_\star \right)
$$

and attains the optimal value $C_N = 1/\Upsilon_N^2$. The pipeline ran as for lemniscate acceleration (Section 4.1.2): bnb-pep-skill wrote the BnB-PEP formulation of this design problem, reproduced with light editing in Appendix B.3, and, upon our approval, solved the resulting QCQP numerically over a range of horizons.

Taylor and Drori [62, Appendix E] previously considered the same optimization problem and numerically obtained ITEM-f for $N = 1, \ldots, 5$ in the setting $L = 1$ and $\mu = 0.1$. From the Stage-1 formulation and its numerical solutions, frontier-llm-consult identified a closed-form expression for the algorithm valid for arbitrary N, μ, and L, together with an analytic feasible point of the dual certifying the contraction factor $1/\Upsilon_N^2$, and then rewrote the method, first obtained in the stepsize variables of the design problem, as the momentum form (ITEM-f) in which this paper states it; continued consultation converted the certificate's multipliers into the Lyapunov proof of Section 4.2.1, stated on that momentum form, and Section 4.2.4 discusses its Lean formalization.

4.2.3 Continuous-time analysis

For a fixed finite horizon $T > 0$ and strong convexity parameter $\mu > 0$, let $\Upsilon > 1$ be the unique real number such that, with $s \triangleq \sqrt{1 - \Upsilon^{-2}}$,

$$
\sqrt{2\mu}\, T = \int_0^{\arcsin s} \frac{dx}{\sqrt{1 - s \sin x}}.
$$

We note the continuous-time limit concerns $L = N^2/T^2$, and with $q_N = \mu T^2/N^2$, we have $\Upsilon_N(q_N) \to \Upsilon$ as $N \to \infty$. Define $a, b \colon [0,T] \to \mathbb{R}$ by

$$
\begin{aligned}
\dot{a}(t) &= b(t) \sqrt{\frac{2\mu a(t)}{\Upsilon}}, \\
\dot{b}(t) &= (\Upsilon - a(t)) \sqrt{\frac{2\mu a(t)}{\Upsilon}},
\end{aligned}
$$

Public theorem declarations	14
Dependency nodes in the closure	20
Lean source files	38
Lines of Lean	7,985
`lake build` jobs	8,694
Comparator replay time, native macOS	105 s
Comparator replay time, WSL2 real Landrun	115 s
Toolchain	Lean 4.32.0, mathlib 4.32.0, commit-pinned
Axioms used	`propext`, `Quot.sound`, `Classical.choice`

Table 3: Summary statistics of the ITEM-f Lean project: it formalizes a fixed 20-node dependency closure of manuscript claims and their supporting steps through fourteen public theorem declarations, builds cleanly under the commit-pinned toolchain, and is accepted by the comparator replay under exactly the three standard axioms listed.

$$a(0) = \frac{1}{\Upsilon},$$
$$b(0) = \frac{\sqrt{\Upsilon^2 - 1}}{\Upsilon}.$$

Then the continuous-time ODE of ITEM-f is

$$\ddot{X}(t) + \frac{3\dot{a}(t)}{2a(t)}\dot{X}(t) + 2\nabla f(X(t)) = 0, \quad 0 \le t \le T.$$

Remark 1 (Continuous-time convergence of ITEM-f)**.** *For an L-smooth and μ-strongly convex function $f\colon \mathbb{R}^d \to \mathbb{R}$, any solution $X \in C^2([0,T];\mathbb{R}^d)$ of the ODE above with $X(0) = x_0$ and $\dot{X}(0) = 0$ satisfies*

$$f(X(T)) - f_\star \le \frac{1}{\Upsilon^2}\left(f(X(0)) - f_\star\right).$$

This can be shown with a continuous-time Lyapunov function

$$\mathcal{V}(t) = a(t)\left(f(X(t)) - f_\star - \frac{\mu}{2}\left\|X(t) - x_\star\right\|^2\right) + \frac{\mu a(t)^2}{2\Upsilon}\left\|X(t) - x_\star\right\|^2 + \frac{1}{4a(t)}\left\|\dot{a}(t)(X(t) - x_\star) + a(t)\dot{X}(t)\right\|^2,$$

by the monotonicity $\dot{\mathcal{V}}(t) \le 0$ and the endpoints $\mathcal{V}(0) = \Upsilon^{-1}\left(f(X(0)) - f_\star\right)$ and $\mathcal{V}(T) \ge \Upsilon\left(f(X(T)) - f_\star\right)$. We do not pursue a detailed proof or Lean verification of this remark; both can be generated by running the AutoOPT pipeline on this continuous-time claim.

4.2.4 Lean formalization

The convergence theory of ITEM-f is machine-checked in the same manner as that of lemniscate acceleration (Section 4.1.5), by a second Lean project whose fourteen public declarations correspond to the results of this section and its appendix. The root declaration `ITEMf.convergence` is the Lean counterpart of Theorem 3, stated over the Euclidean space $\mathbb{R}^d$ and proving both inequalities of (18); it reads:

```
ITEM-f/  Lean/Lake project for analytic ITEM-f
├── lakefile.toml, lean-toolchain  build configuration; commit-pinned Lean and mathlib
├── config.json  pins the fourteen theorem names and permitted axioms for the comparator
├── Challenge.lean  states the fourteen declarations, one approved placeholder each
├── Solution.lean  the same statements, closed by the proof development
├── AxiomAudit.lean  runs the axiom audit over the fourteen declarations
├── ITEMf.lean  umbrella module importing the development
├── ITEMf/  proof modules, organized in dependency layers:
│   ├── Spec/  proof-free statement interfaces; all that Challenge.lean imports
│   ├── Model/  problem class, interpolation inequality, transformed objective
│   ├── Construction/  the shooting argument: one-step map, orbit, target angle, symmetry, rate
│   ├── Algorithm/  iterates, coefficient identities, coordinate relations
│   ├── Lyapunov/  norm identities, rotations, monotonicity
│   └── Convergence.lean  the convergence theorem
└── artifacts/  build, hygiene, axiom-audit, and comparator replay logs
```

Figure 11: File structure of the ITEM-f Lean project (module files grouped by folder). The proof-free `Spec/` layer is all that `Challenge.lean` imports, so the proofs reach the comparator only through `Solution.lean`; `config.json` pins the fourteen public theorems and the permitted axioms, and `artifacts/` records the build, audit, and replay logs.

```
theorem convergence
    {d N : Nat} (hd : 1 ≤ d) (hN : 1 ≤ N)
    (M : StronglyConvexSmoothModel (Euclidean d)) {xStar : Euclidean d}
    (hxStar : M.IsMinimizer xStar)
    (C : CoeffData N) (hC : ValidCoefficients M.q C)
    (x0 : Euclidean d) :
    M.f (itemfIterate M C x0 N) - M.f xStar ≤
          (1 / C.Upsilon ^ 2) * (M.f x0 - M.f xStar) ∧
      (1 / C.Upsilon ^ 2) * (M.f x0 - M.f xStar) ≤
          4 * (1 - Real.sqrt M.q) ^ (2 * N) *
            (M.f x0 - M.f xStar)
```

The supporting declarations follow the paper's proofs. `coefficientsExistUnique` is the construction Lemma 4, obtained through the shooting argument of Appendix B.1 via `oneStepMap` (Lemma 13), `orbitComparison` (Lemma 14), `targetAngle` (Lemma 15), and `shootingIffAdmissible` (Lemma 16); `involutionSymmetry` is Lemma 17; `explicitRate` is Lemma 18; and `finiteInterpolation` is (6). The Lyapunov argument of Section 4.2.1 is formalized by `transformedMemF0`, `threePointIdentity`, `coordinateRelations` (the identities (46)), `normIdentities` (Lemma 5), and `lyapunovMonotone` (Lemma 6).

The project comprises 38 Lean source files and 7,985 lines of Lean, organized as in Figure 11 in a layout parallel to the LemniAcc project, with one addition: a proof-free specification layer (`ITEMf/Spec`) contains the definitions and statements of the fourteen declarations, and the comparator-facing `Challenge.lean`, which restates each declaration with exactly one approved placeholder, imports only this layer, so the proofs reach the comparator only through `Solution.lean`, which closes every placeholder. The project passes `lake build` under the same commit-pinned toolchain (8,694 build jobs); the hygiene scan confirms that no `sorry`, `axiom`, `admit`, or `unsafe` appears anywhere in the proof development or in `Solution.lean`; and the axiom audit again reports exactly `propext`, `Quot.sound`, and `Classical.choice` for every public declaration. The comparator accepted the fourteen public theorems twice, under the two-platform protocol, evidence label, and disclosures of Section 4.1.5: in 105 seconds on native macOS and in 115 seconds under the isolated WSL2 real-Landrun replay; Table 3 summarizes the project.

4.3 Self-H-duality of LemniAcc and ITEM-f

The notion of *H-duality* was introduced in [32] and further developed in [33, 72, 73, 71]. Recall that a fixed-step first-order method (FSFOM) with stepsizes $\{h_{i,j}\}_{0\le j<i\le N}$ is a method of the form

$$x_i = x_{i-1} - \frac{1}{L}\sum_{j=0}^{i-1} h_{i,j}\,\nabla f(x_j), \qquad i = 1,\dots,N.$$

The *H-dual* of this method is the FSFOM with stepsizes $h^{\mathrm{A}}_{i,j} \triangleq h_{N-j,N-i}$ for $0 \le j < i \le N$. Viewing the lower-triangular array $\{h_{i,j}\}_{0\le j<i\le N}$ as an $(N+1)\times(N+1)$ matrix, taking the H-dual corresponds to taking the anti-diagonal transpose, i.e., the entries are reflected across the anti-diagonal rather than the usual diagonal. These prior works observe that H-duality exchanges the type of guarantee while preserving its worst-case value: for example, OGM, which reduces the function value under an initial-distance condition, maps to OGM-G, which reduces the squared gradient norm under an initial function-gap condition, with exactly the same constant [32].

Remark 2 (Self-H-duality)**.** *In this work, we point out that* (LemniAcc) *and* (ITEM-f) *are* self-H-dual *in the sense that each method is its own H-dual. Moreover, the corresponding continuous-time ODEs are also self-H-dual in the sense of [32]: the ODEs have the form* $\ddot{X}(t) + \gamma(t)\dot{X}(t) + 2\nabla f(X(t)) = 0$*, with the symmetry* $\gamma(t) = \gamma(T-t)$ *for* $0 < t < T$*. We record this observation without a detailed proof or Lean verification; a reader who wants both can obtain them by running the* AutoOPT *pipeline on this claim.*

5 Conclusion

In this work, we present AutoOPT, a framework that automates much of the labor-intensive parts in the design and analysis of optimal first-order methods. AutoOPT combines the domain-specific methodology of performance estimation programming with the general-purpose intelligence of frontier LLMs. We demonstrate the strength of this harness with two case studies, producing two optimal algorithms of independent interest: lemniscate acceleration (Section 4.1) and analytic ITEM-f (Section 4.2). More broadly, this work serves as a case study of how a domain-specific harness can augment the general-purpose intelligence of LLMs.

AutoOPT has significant limitations. For example, it does not handle stochastic or second-order optimization. We expect AutoOPT itself to be extensible to such settings, and we believe the general principle of the domain-specific harness extends further still.

Finally, we offer this paper itself as one data point on what scholarship can look like in the era of LLM intelligence. As stated in the Preface, we believe domain experts can contribute productively in an AI-driven research workflow by augmenting the general-purpose intelligence of LLMs with harnesses derived from their expertise. Automation lets us take on harder and more interesting problems and elevate our scholarship, and it frees the researcher to concentrate on the parts of research that are irreducibly human: choosing the problems, interpreting the answers, and conveying the insight to others.

Acknowledgments

S. Das Gupta and E. K. Ryu acknowledge support by AFOSR Grant Number FA9550-25-1-0183.

References

[1] Agent Skills. Agent Skills overview, 2026. URL: `https://agentskills.io/home` [cited 2026-08-03].

[2] Jason M. Altschuler and Pablo A. Parrilo. Acceleration by stepsize hedging: Multi-Step Descent and the Silver Stepsize Schedule. *Journal of the ACM*, 72(2):1–38, 2025. `doi:10.1145/3708502`.

[3] Jason M. Altschuler and Pablo A. Parrilo. Acceleration by stepsize hedging: Silver Stepsize Schedule for smooth convex optimization. *Mathematical Programming*, 213(1–2):1105–1118, 2025. `doi:10.1007/s10107-024-02164-2`.

[4] Jason M. Altschuler and Pablo A. Parrilo. Stepsize Hedging: An alternative mechanism for accelerating gradient descent, 2026. To appear as an invited Research Highlight in the 2026 INFORMS Computing Society Newsletter. `arXiv:2605.31386`.

[5] Anthropic. Claude Code overview, 2026. URL: `https://code.claude.com/docs/en/overview` [cited 2026-08-03].

[6] Anthropic. Extend Claude with skills, 2026. URL: `https://code.claude.com/docs/en/skills` [cited 2026-08-03].

[7] Artelys. Artelys Knitro, 2026. URL: `https://www.artelys.com/solvers/knitro/` [cited 2026-08-03].

[8] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021. `doi:10.1126/science.abj8754`.

[9] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017. `doi:10.1137/141000671`.

[10] Richard H. Byrd, Jorge Nocedal, and Richard A. Waltz. KNITRO: An integrated package for nonlinear optimization. In Gianni di Pillo and Massimo Roma, editors, *Large-Scale Nonlinear Optimization*, volume 83 of *Nonconvex Optimization and Its Applications*, pages 35–59. Springer, Boston, MA, 2006. `doi:10.1007/0-387-30065-1_4`.

[11] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of

optimization algorithms. In *Advances in Neural Information Processing Systems*, volume 36, pages 49205–49233, 2023. doi:10.52202/075280-2140.

[12] COIN-OR. Ipopt license, 2026. URL: https://coin-or.github.io/Ipopt/LICENSE.html [cited 2026-08-03].

[13] Shuvomoy Das Gupta. Research highlight: A tutorial on branch-and-bound performance estimation programming. *INFORMS Computing Society Newsletter*, pages 15–28, May 2025. URL: https://higherlogicdownload.s3.amazonaws.com/INFORMS/a4f852dd-2bdc-46bc-98e7-47e7d7c12752/UploadedImages/ICSNews_May2025.pdf.

[14] 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, 2024. doi:10.1007/s10107-023-01973-1.

[15] Alexandre d'Aspremont, Damien Scieur, and Adrien B. Taylor. Acceleration methods. *Foundations and Trends® in Optimization*, 5(1–2):1–245, 2021. doi:10.1561/2400000036.

[16] Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, volume 12699 of *Lecture Notes in Computer Science*, pages 625–635. Springer, 2021. doi:10.1007/978-3-030-79876-5_37.

[17] Yoel Drori. The exact information-based complexity of smooth convex minimization. *Journal of Complexity*, 39:1–16, 2017. doi:10.1016/j.jco.2016.11.001.

[18] Yoel Drori and Adrien B. Taylor. On the oracle complexity of smooth strongly convex minimization. *Journal of Complexity*, 68:101590, 2022. doi:10.1016/j.jco.2021.101590.

[19] 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. doi:10.1007/s10107-013-0653-0.

[20] Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, David Silver, Demis Hassabis, and Pushmeet Kohli. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022. doi:10.1038/s41586-022-05172-4.

[21] Tony Feng, Trieu H. Trinh, Garrett Bingham, Dawsen Hwang, Yuri Chervonyi, Junehyuk Jung, Joonkyung Lee, Carlo Pagano, Sang-hyun Kim, Federico Pasqualotto, Sergei Gukov, Jonathan N. Lee, Junsu Kim, Kaiying Hou, Golnaz Ghiasi, Yi Tay, YaGuang Li, Chenkai Kuang, Yuan Liu, Hanzhao Lin, Evan Zheran Liu, Nigamaa Nayakanti, Xiaomeng Yang, Heng-Tze Cheng, Demis Hassabis, Koray Kavukcuoglu, Quoc V. Le, and Thang Luong. Towards autonomous mathematics research, 2026. arXiv:2602.10177.

[22] Baptiste Goujaud, Céline Moucer, François Glineur, Julien M. Hendrickx, Adrien B. Taylor, and Aymeric Dieuleveut. PEPit: computer-assisted worst-case analyses of first-order optimization methods in Python. *Mathematical Programming Computation*, 16(3):337–367, 2024. doi:10.1007/s12532-024-00259-7.

[23] Benjamin Grimmer. Provably faster gradient descent via long steps. *SIAM Journal on Optimization*, 34(3):2588–2608, 2024. doi:10.1137/23M1588408.

[24] Gurobi Optimization, LLC. Gurobi academic licenses, 2026. URL: https://www.gurobi.com/academics [cited 2026-08-03].

[25] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, version 13.0, 2026. URL: https://docs.gurobi.com/projects/optimizer/en/current/ [cited 2026-08-03].

[26] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2012. doi:10.1017/CBO9781139020411.

[27] Uijeong Jang, Shuvomoy Das Gupta, and Ernest K. Ryu. Computer-assisted design of accelerated composite optimization methods: OptISTA. *Mathematical Programming*, 2025. doi:10.1007/s10107-025-02258-5.

[28] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi:10.1038/s41586-021-03819-2.

[29] Phillip Kerger. Closing the oracle-complexity gap in derivative-free convex optimization: A near-quadratic lower bound from exact function values, 2026. arXiv:2607.13335.

[30] Donghwan Kim and Jeffrey A. Fessler. Optimized first-order methods for smooth convex minimization. *Mathematical Programming*, 159(1-2):81–107, 2016. doi:10.1007/s10107-015-0949-3.

[31] Donghwan Kim and Jeffrey A. Fessler. Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions. *Journal of Optimization Theory and Applications*, 188(1):192–219, 2021. doi:10.1007/s10957-020-01770-2.

[32] Jaeyeon Kim, Asuman Ozdaglar, Chanwoo Park, and Ernest K. Ryu. Time-reversed dissipation induces duality between minimizing gradient norm and function value. *Advances in Neural Information Processing Systems*, 36:23389–23440, 2023. doi:10.52202/075280-1014.

[33] Jaeyeon Kim, Chanwoo Park, Asuman Ozdaglar, Jelena Diakonikolas, and Ernest K. Ryu. Mirror duality in convex optimization, 2023. arXiv:2311.17296.

[34] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery, 2024. arXiv:2408.06292.

[35] Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of AI research. *Nature*, 651(8107):914–919, 2026. doi:10.1038/s41586-026-10265-5.

[36] Miles Lubin, Oscar Dowson, Joaquim Dias Garcia, Joey Huchette, Benoît Legat, and Juan Pablo Vielma. JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*, 15:581–589, 2023. doi:10.1007/s12532-023-00239-3.

[37] Shiqian Ma. Convergence of BDRS as a matrix scaling algorithm. Optimization Online preprint, 2026. URL: `https://optimization-online.org/2026/05/convergence-of-bdrs-as-a-matrix-scaling-algorithm/`.

[38] MOSEK ApS. MOSEK academic licenses, 2026. URL: `https://www.mosek.com/products/academic-licenses/` [cited 2026-08-03].

[39] MOSEK ApS. The MOSEK optimizer API for Julia 11.2.2, 2026. URL: `https://docs.mosek.com/latest/juliaapi/index.html` [cited 2026-08-03].

[40] Arkadi S. Nemirovsky. On optimality of Krylov's information when solving linear operator equations. *Journal of Complexity*, 7(2):121–130, 1991. `doi:10.1016/0885-064X(91)90001-E`.

[41] Arkadi S. Nemirovsky. Information-based complexity of linear operator equations. *Journal of Complexity*, 8(2):153–175, 1992. `doi:10.1016/0885-064X(92)90013-2`.

[42] Arkadi S. Nemirovsky and David B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, 1983.

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

[44] Yurii Nesterov, Alexander Gasnikov, Sergey Guminov, and Pavel Dvurechensky. Primal-dual accelerated gradient methods with small-dimensional relaxation oracle. *Optimization Methods and Software*, 36(4):773–810, 2021. `doi:10.1080/10556788.2020.1731747`.

[45] Nous Research. Hermes Agent, 2026. URL: `https://hermes-agent.nousresearch.com/` [cited 2026-08-03].

[46] Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery, 2025. `arXiv:2506.13131`.

[47] OpenAI. Introducing ChatGPT Pro, 2024. URL: `https://openai.com/index/introducing-chatgpt-pro/` [cited 2026-08-03].

[48] OpenAI. Build skills, 2026. URL: `https://learn.chatgpt.com/docs/build-skills` [cited 2026-08-03].

[49] OpenAI. Codex CLI, 2026. URL: `https://developers.openai.com/codex/cli` [cited 2026-08-03].

[50] OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition, 2026. URL: `https://openai.com/index/gpt-5-6/` [cited 2026-08-03].

[51] OpenAI. Introducing GPT-5.5, 2026. URL: `https://openai.com/index/introducing-gpt-5-5/` [cited 2026-08-03].

[52] OpenAI. Ten advances in mathematics and theoretical computer science, 2026. URL: `https://cdn.openai.com/pdf/ten-proofs-oai.pdf` [cited 2026-08-03].

[53] OpenClaw Foundation. OpenClaw docs, 2026. URL: `https://docs.openclaw.ai/` [cited 2026-08-03].

[54] OpenCode. OpenCode: The open source AI coding agent, 2026. URL: `https://opencode.ai/` [cited 2026-08-03].

[55] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024. `doi:10.1038/s41586-023-06924-6`.

[56] Anne Rubbens and Julien M. Hendrickx. A constraint-based approach to function interpolation, with application to performance estimation for weakly convex optimization. *Mathematical Programming*, 2026. `doi:10.1007/s10107-026-02353-1`.

[57] Anne Rubbens, Julien M. Hendrickx, and Adrien B. Taylor. A constructive approach to strengthen algebraic descriptions of function and operator classes, 2025. `arXiv:2504.14377`.

[58] Jaewook J. Suh, Gyumin Roh, and Ernest K. Ryu. Continuous-time analysis of accelerated gradient methods via conservation laws in dilated coordinate systems. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20640–20667. PMLR, 2022. URL: `https://proceedings.mlr.press/v162/suh22a.html`.

[59] Jaewook J. Suh, TaeHo Yoon, Edward D. H. Nguyen, Bicheng Ying, and Shiqian Ma. Peppy: An AI-assisted workflow for tight convergence analysis of optimization algorithms. In *ICML 2026 Workshop AI4Research*, 2026. Seoul, South Korea. URL: `https://openreview.net/forum?id=q7TfzOgGnb`.

[60] Anja Surina, Arun Suggala, George Tsoukalas, Anton Kovsharov, Sergey Shirobokov, Francisco J. R. Ruiz, Pushmeet Kohli, and Swarat Chaudhuri. An improved last-iterate convergence rate for anchored gradient descent ascent, 2026. `arXiv:2604.03782`.

[61] Kyle Swanson, Wesley Wu, Nash L. Bulaong, John E. Pak, and James Zou. The Virtual Lab of AI agents designs new SARS-CoV-2 nanobodies. *Nature*, 646(8085):716–723, 2025. `doi:10.1038/s41586-025-09442-9`.

[62] Adrien B. Taylor and Yoel Drori. An optimal gradient method for smooth strongly convex minimization. *Mathematical Programming*, 199(1–2):557–594, 2023. `doi:10.1007/s10107-022-01839-y`.

[63] 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. `doi:10.1137/16M108104X`.

[64] Adrien B. Taylor, Julien M. Hendrickx, and François Glineur. Performance estimation toolbox (PESTO): automated worst-case analysis of first-order optimization methods. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1278–1283. IEEE, 2017. `doi:10.1109/CDC.2017.8263832`.

[65] 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. `doi:10.1007/s10107-016-1009-3`.

[66] The mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 367–381. Association for Computing Machinery, 2020. `doi:10.1145/3372885.3373824`.

[67] George Tsoukalas, Anton Kovsharov, Sergey Shirobokov, Anja Surina, Moritz Firsching, Gergely Bérczi, et al. Advancing mathematics research with AI-driven formal proof search, 2026. `arXiv:2605.22763`.

[68] Manu Upadhyaya, Daniel Berg Thomsen, Aymeric Dieuleveut, and Adrien B. Taylor. An optimal first-order method for smooth and strongly convex composite optimization and its stationary limit, 2026. `arXiv:2605.22929`.

[69] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006. `doi:10.1007/s10107-004-0559-y`.

[70] Michel Waldschmidt. Elliptic functions and transcendence. In Krishnaswami Alladi, editor, *Surveys in Number Theory*, volume 17 of *Developments in Mathematics*, pages 143–188. Springer, New York, 2008. `doi:10.1007/978-0-387-78510-3_7`.

[71] TaeHo Yoon and Benjamin Grimmer. A theory of composition and duality of extremal optimal fixed-point algorithms, 2026. `arXiv:2605.02231`.

[72] TaeHo Yoon, Jaeyeon Kim, Jaewook J. Suh, and Ernest K. Ryu. Optimal acceleration for minimax and fixed-point problems is not unique. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 57244–57314. PMLR, 2024. URL: `https://proceedings.mlr.press/v235/yoon24b.html`.

[73] TaeHo Yoon, Ernest K. Ryu, and Benjamin Grimmer. H-invariance theory: A complete characterization of minimax optimal fixed-point algorithms, 2025. `arXiv:2511.14915`.

[74] Max Zimmer, Nico Pelleriti, Christophe Roux, and Sebastian Pokutta. The agentic researcher: A practical guide to AI-assisted research in mathematics and machine learning, 2026. `arXiv:2603.15914`.

A Deferred proofs and details for lemniscate acceleration

A.1 Existence and uniqueness of the lemniscate recurrence

In this section we prove Lemma 1, establish the bounds on Ω_N used in Theorem 1, and give a complete bisection rule for computing Ω_N.

The one-step map. For $\Omega > 0$ and $\rho \in (0, 1]$, a pair of consecutive terms of (9) solves

$$\Omega\left(\rho - t\right)^2 = \rho\left(1 - t^2\right) \tag{23}$$

in the unknown t, which rearranges to the quadratic equation $(\Omega+\rho)\,t^2 - 2\Omega\rho\,t + \Omega\rho^2 - \rho = 0$. We work on the admissible set $\mathcal{A} \triangleq \{(\Omega,\rho) : \Omega > 0,\ 0 < \rho \le 1,\ \Omega\rho \ge 1\}$.

Lemma 7 (One-step map)**.** *For $(\Omega,\rho) \in \mathcal{A}$, equation (23) has exactly one solution t with $0 \le t < \rho$, namely*

$$F_\Omega(\rho) \triangleq \frac{\Omega\rho - \sqrt{\rho^2 + \Omega\rho(1-\rho^2)}}{\Omega+\rho}, \tag{24}$$

and the other solution is greater than or equal to ρ, with equality only when $\rho = 1$. Moreover:

1. *$F_\Omega(\rho) = 0$ if and only if $\Omega\rho = 1$;*
2. *F is continuous on $\mathcal{A}$ and strictly increasing in each of Ω and ρ on the fibers of $\mathcal{A}$ (the other variable being fixed).*

Proof. The quarter-discriminant of the quadratic equals $\Omega^2\rho^2 - (\Omega+\rho)(\Omega\rho^2-\rho) = \rho^2 + \Omega\rho(1-\rho^2) > 0$, so (23) has the two real solutions

$$t_\pm = \frac{\Omega\rho \pm \sqrt{\rho^2 + \Omega\rho(1-\rho^2)}}{\Omega+\rho}.$$

Since $\Omega\rho(1-\rho^2) \ge 0$, the square root is at least ρ, hence

$$t_+ \ge \frac{\Omega\rho+\rho}{\Omega+\rho} \ge \frac{\Omega\rho+\rho^2}{\Omega+\rho} = \rho,$$

and both inequalities hold with equality only when $\rho = 1$. On the other hand, $t_- < \Omega\rho/(\Omega+\rho) < \rho$. The product of the two solutions is $t_+t_- = \rho(\Omega\rho-1)/(\Omega+\rho) \ge 0$ on $\mathcal{A}$; since $t_+ \ge \rho > 0$, this gives $t_- \ge 0$, with $t_- = 0$ exactly when $\Omega\rho = 1$. Therefore $F_\Omega(\rho) = t_-$ is the unique solution in $[0,\rho)$, which proves the root dichotomy and claim 1.

For claim 2, continuity follows from (24) because the denominator is positive. For the strict monotonicity, define

$$\psi(\Omega,\rho,t) \triangleq \Omega(\rho-t)^2 - \rho(1-t^2),$$

and fix a point in the interior of $\mathcal{A}$ with $t = F_\Omega(\rho) \in [0,\rho)$. Substituting the root identity $1-t^2 = \Omega(\rho-t)^2/\rho$ into the partial derivatives of ψ and using $0 \le t < \rho \le 1$, tedious but straightforward algebra gives $\psi_\Omega > 0$, $\psi_\rho > 0$, and $\psi_t < 0$ at this point. By the implicit function theorem, F is continuously differentiable in the interior of $\mathcal{A}$ with $\partial F/\partial\Omega = -\psi_\Omega/\psi_t > 0$ and $\partial F/\partial\rho = -\psi_\rho/\psi_t > 0$. Consequently, for fixed $\rho \in (0,1)$ the map $\Omega \mapsto F_\Omega(\rho)$ is continuous on $[1/\rho,\infty)$ and strictly increasing on its interior, hence strictly increasing on $[1/\rho,\infty)$; the same argument applies to $\rho \mapsto F_\Omega(\rho)$ on $[1/\Omega,1]$ for fixed $\Omega > 1$; and on the remaining boundary segment $\rho = 1$ the explicit formula $F_\Omega(1) = (\Omega-1)/(\Omega+1)$ is strictly increasing in Ω. □

Shooting orbits. For $\Omega \ge 1$ define the shooting iterates $\Phi_0(\Omega) \triangleq 1$ and $\Phi_{k+1}(\Omega) \triangleq F_\Omega\left(\Phi_k(\Omega)\right)$ defined for as long as $(\Omega,\Phi_k(\Omega)) \in \mathcal{A}$. Taking the positive square root in (23) gives the step identity

$$\Phi_k(\Omega) - \Phi_{k+1}(\Omega) = \sqrt{\frac{\Phi_k(\Omega)\,(1-\Phi_{k+1}(\Omega)^2)}{\Omega}} \le \frac{1}{\sqrt{\Omega}} \tag{25}$$

whenever the left-hand side is defined; the final bound holds because the iterates satisfy $0 \le \Phi_{k+1}(\Omega) < \Phi_k(\Omega) \le \Phi_0(\Omega) = 1$ (Lemma 7), so $\Phi_k(\Omega)\left(1-\Phi_{k+1}(\Omega)^2\right) \le 1$.

Proof of Lemma 1. Call a pair $(\Omega, \{\rho_k\}_{k=0}^{N+1})$ *admissible for horizon* N if it satisfies all requirements of the lemma: $1 = \rho_0 > \rho_1 > \cdots > \rho_{N+1} = 0$ and (9) for $0 \le k \le N$.

Step 0: every admissible pair is a shooting orbit. Let $(\Omega, \{\rho_k\})$ be admissible for horizon N. Strict decrease gives $\rho_k > \rho_{N+1} = 0$ for $k \le N$. The recurrence at $k = N$ reads $\Omega\rho_N^2 = \rho_N$, so $\rho_N = 1/\Omega$, and the decrease of the sequence gives $\Omega\rho_k \ge \Omega\rho_N = 1$ for $k \le N$; hence $(\Omega, \rho_k) \in \mathcal{A}$ for $0 \le k \le N$. In particular $1/\Omega = \rho_N \le \rho_0 = 1$ forces $\Omega \ge 1$, and for $N \ge 1$ the strict inequality $\rho_N < \rho_0$ forces $\Omega > 1$. Finally, for $0 \le k \le N$ the term ρ_{k+1} solves (23) with $0 \le \rho_{k+1} < \rho_k$, so Lemma 7 gives $\rho_{k+1} = F_\Omega(\rho_k)$. Therefore $\rho_k = \Phi_k(\Omega)$ for all $0 \le k \le N+1$: the sequence is uniquely determined by Ω, which is the final claim of the lemma.

Step 1: existence, by induction on the horizon. We construct numbers $1 = \Omega_0 < \Omega_1 < \Omega_2 < \cdots$ such that $\left(\Omega_N, \{\Phi_k(\Omega_N)\}_{k=0}^{N+1}\right)$ is admissible for horizon N. For $N = 0$ the requirements reduce to $\Omega(\rho_0 - \rho_1)^2 = \rho_0(1 - \rho_1^2)$ with $\rho_0 = 1$ and $\rho_1 = 0$, that is, $\Omega_0 = 1$; and indeed $\Phi_1(1) = F_1(1) = 0$.

For the induction step, assume $(\Omega_N, \{\Phi_k(\Omega_N)\})$ is admissible for horizon N. We first record, by induction on $k \in \{1, \ldots, N+1\}$, that the maps $\Omega \mapsto \Phi_k(\Omega)$ are well defined, continuous, and strictly increasing on the ray $[\Omega_N, \infty)$. For $k = 1$, $\Phi_1(\Omega) = F_\Omega(1) = (\Omega - 1)/(\Omega + 1)$ has all three properties. For the step from k to $k+1$ (with $k \le N$), the monotonicity of F and admissibility at horizon N give the domain bound $\Omega\,\Phi_k(\Omega) \ge \Omega_N\,\Phi_k(\Omega_N) \ge \Omega_N\,\Phi_N(\Omega_N) = \Omega_N/\Omega_N = 1$, so $(\Omega, \Phi_k(\Omega)) \in \mathcal{A}$ and $\Phi_{k+1}(\Omega) = F_\Omega(\Phi_k(\Omega))$ is well defined; it is continuous by Lemma 7 and the induction hypothesis, and strictly increasing because, for $\Omega_N \le \Omega < \Omega'$, we have $\Phi_{k+1}(\Omega) = F_\Omega\left(\Phi_k(\Omega)\right) < F_{\Omega'}\left(\Phi_k(\Omega)\right) < F_{\Omega'}\left(\Phi_k(\Omega')\right) = \Phi_{k+1}(\Omega')$, with the middle application being legitimate since $\Omega'\,\Phi_k(\Omega) > \Omega\,\Phi_k(\Omega) \ge 1$, and the second inequality being strict by the induction hypothesis $\Phi_k(\Omega) < \Phi_k(\Omega')$.

Now consider $h(\Omega) \triangleq \Omega\,\Phi_{N+1}(\Omega)$ on $[\Omega_N, \infty)$. It is continuous, and it is strictly increasing because Φ_{N+1} is nonnegative and strictly increasing there: for $\Omega_N \le \Omega < \Omega'$, $\Omega\,\Phi_{N+1}(\Omega) \le \Omega\,\Phi_{N+1}(\Omega') < \Omega'\,\Phi_{N+1}(\Omega')$. Moreover $h(\Omega_N) = \Omega_N \cdot \Phi_{N+1}(\Omega_N) = 0$, and telescoping (25) over $0 \le k \le N$, all iterates being defined on this ray, gives $\Phi_{N+1}(\Omega) \ge 1 - (N+1)/\sqrt{\Omega}$, so $h(\Omega) \ge \Omega - (N+1)\sqrt{\Omega} \to \infty$ as $\Omega \to \infty$. By the intermediate value theorem there is a unique $\Omega_{N+1} \in (\Omega_N, \infty)$ with $h(\Omega_{N+1}) = 1$, that is, $\Phi_{N+1}(\Omega_{N+1}) = 1/\Omega_{N+1}$. The sequence $\rho_k \triangleq \Phi_k(\Omega_{N+1})$ for $0 \le k \le N+1$, extended by $\rho_{N+2} \triangleq 0$, is then admissible for horizon $N+1$: the recurrence holds for $0 \le k \le N$ by construction, and at $k = N+1$ it reads $\Omega_{N+1}(\rho_{N+1} - 0)^2 = 1/\Omega_{N+1} = \rho_{N+1}(1 - 0^2)$; the sequence is strictly decreasing because $F_\Omega(\rho) < \rho$ at every step and $\rho_{N+1} = 1/\Omega_{N+1} > 0 = \rho_{N+2}$; and it is nonnegative because $\rho_k \ge \rho_{N+1} > 0$ for $k \le N+1$. This completes the induction, so an admissible Ω exists for every horizon.

Step 2: uniqueness of Ω. Suppose $(\Omega, \{\rho_k\})$ and $(\Omega', \{\rho_k'\})$ are both admissible for horizon N with $\Omega < \Omega'$. By Step 0, both sequences are shooting orbits, every pair (Ω, ρ_k) and (Ω', ρ_k') with $k \le N$ lies in $\mathcal{A}$, and $\Omega, \Omega' \ge 1$. For $N = 0$, Step 0 gives $\rho_1 = (\Omega - 1)/(\Omega + 1) = 0$ and $\rho_1' = (\Omega' - 1)/(\Omega' + 1) = 0$, forcing $\Omega = \Omega' = 1$ and contradicting $\Omega < \Omega'$; so assume $N \ge 1$. We prove $\rho_k < \rho_k'$ for $1 \le k \le N+1$ by induction. For $k = 1$,

$$\rho_1 = \frac{\Omega - 1}{\Omega + 1} < \frac{\Omega' - 1}{\Omega' + 1} = \rho_1'.$$

For the induction step, the strict monotonicity of F in both arguments (Lemma 7) gives $\rho_{k+1} = F_\Omega(\rho_k) < F_{\Omega'}(\rho_k) < F_{\Omega'}(\rho_k') = \rho_{k+1}'$, where the middle steps are valid because $\Omega'\rho_k > \Omega\rho_k \ge 1$, and the second inequality is strict by the induction hypothesis $\rho_k < \rho'_k$. Iterating the induction through the final step $k = N$ yields $\rho_{N+1} < \rho'_{N+1}$, contradicting $\rho_{N+1} = \rho'_{N+1} = 0$. Hence the admissible Ω is unique, and Step 0 shows the sequence is uniquely determined by it. $\square$

The same machinery yields the quantitative bounds on Ω_N quoted in Theorem 1 and a complete bisection rule for computing Ω_N.

Corollary 2 (Bounds on Ω_N and bisection correctness)**.** *Let $N \geq 1$, and let Ω_N and $\{\rho_k\}_{k=0}^{N+1}$ be as in Lemma 1. Then:*

1. $\dfrac{(N+1)^2}{\varpi^2} < \Omega_N < (N+1)^2$*, where ϖ is the lemniscate constant defined in Theorem 1; moreover* $1 = \Omega_0 < \Omega_1 < \Omega_2 < \cdots$.
2. *Fix $\Omega \geq 1$ and generate the shooting iterates $\Phi_0(\Omega), \Phi_1(\Omega), \ldots,$ computing $\Phi_{k+1}(\Omega) = F_\Omega(\Phi_k(\Omega))$ whenever $\Phi_k(\Omega) \geq 1/\Omega$. If $\Omega > \Omega_N$, then $\Phi_k(\Omega) > 1/\Omega$ for all $0 \leq k \leq N$ and $\Phi_{N+1}(\Omega) > 0$. If $\Omega < \Omega_N$, then there is an index $k \leq N$ with $\Phi_k(\Omega) < 1/\Omega$. If $\Omega = \Omega_N$, the iterates realize the sequence of Lemma 1.*
   Consequently, the following exact bisection is correct. Initialize $[\Omega_l, \Omega_r] = [1, (N+1)^2]$. Given the midpoint $\Omega_m = (\Omega_l + \Omega_r)/2$, generate the shooting iterates until either some $k \leq N$ satisfies $\Phi_k(\Omega_m) < 1/\Omega_m$, in which case set $\Omega_l \leftarrow \Omega_m$, or the iterates have been generated through $\Phi_{N+1}(\Omega_m)$. In the latter case, set $\Omega_r \leftarrow \Omega_m$ if $\Phi_{N+1}(\Omega_m) > 0$, and return $\Omega_m = \Omega_N$ and the generated sequence if $\Phi_{N+1}(\Omega_m) = 0$. At every nonterminal step the invariant $\Omega_l < \Omega_N \leq \Omega_r$ is maintained, the bracket length is halved, and the midpoints converge to Ω_N.

Proof. Part 1. At $\Omega = \Omega_N$ the step identity (25) holds for $0 \leq k \leq N$, and telescoping gives

$$1 = \rho_0 - \rho_{N+1} = \sum_{k=0}^{N} \sqrt{\frac{\rho_k\left(1 - \rho_{k+1}^2\right)}{\Omega_N}} \leq \frac{N+1}{\sqrt{\Omega_N}},$$

and the term $k = 1$, present since $N \geq 1$, is strictly smaller than $1/\sqrt{\Omega_N}$ because $\rho_1 < 1$; hence $\sqrt{\Omega_N} < N + 1$. For the lower bound, dividing the step identity by $\sqrt{\rho_k(1 - \rho_{k+1}^2)}$ and summing gives

$$\frac{N+1}{\sqrt{\Omega_N}} = \sum_{k=0}^{N} \frac{\rho_k - \rho_{k+1}}{\sqrt{\rho_k\left(1 - \rho_{k+1}^2\right)}} = \sum_{k=0}^{N} \int_{\rho_{k+1}}^{\rho_k} \frac{dt}{\sqrt{\rho_k\left(1 - \rho_{k+1}^2\right)}} < \sum_{k=0}^{N} \int_{\rho_{k+1}}^{\rho_k} \frac{dt}{\sqrt{t\left(1 - t^2\right)}} = \int_0^1 \frac{dt}{\sqrt{t\left(1 - t^2\right)}} = \varpi,$$

where the strict inequality holds because on the interior of each interval the integrand on the right strictly dominates the constant integrand on the left, and the substitution $t = s^2$ identifies the comparison integral with the form $\varpi = 2\int_0^1 ds/\sqrt{1 - s^4}$ of Section 4.1.3. Hence $\Omega_N > (N+1)^2/\varpi^2$. Strict monotonicity of $N \mapsto \Omega_N$ was established in Step 1 of the proof of Lemma 1 and is inherited by the unique admissible values.

Part 2. If $\Omega > \Omega_N$, the sub-induction in Step 1 (anchored at Ω_N) shows that $\Phi_k(\Omega)$ is defined and strictly increasing in Ω for $1 \leq k \leq N+1$; therefore, for $0 \leq k \leq N$, using that the iterates decrease in the index ($F_\Omega(\rho) < \rho$),

$$\Phi_k(\Omega) \geq \Phi_N(\Omega) > \Phi_N(\Omega_N) = \frac{1}{\Omega_N} > \frac{1}{\Omega},$$

and $\Phi_{N+1}(\Omega) > \Phi_{N+1}(\Omega_N) = 0$. If $1 \le \Omega < \Omega_N$, suppose toward a contradiction that $\Phi_k(\Omega) \ge 1/\Omega$ for all $0 \le k \le N$; then all iterates up to $\Phi_N(\Omega)$ are defined. The comparison induction of Step 2 does not use admissibility of the lower orbit at the terminal index: it only needs both shooting orbits to be defined up to the index being compared, the domain of each step being guaranteed here by the contradiction hypothesis and by $\Omega_N\, \Phi_k(\Omega) > \Omega\, \Phi_k(\Omega) \ge 1$. The same induction therefore gives $\Phi_N(\Omega) < \Phi_N(\Omega_N) = 1/\Omega_N < 1/\Omega$, a contradiction. The case $\Omega = \Omega_N$ is Lemma 1 itself.

For the bisection, Part 1 gives $\Omega_l = 1 < \Omega_N < (N+1)^2 = \Omega_r$ at initialization. The sign characterization just proved says that the left update occurs exactly when $\Omega_m < \Omega_N$, the right update occurs exactly when $\Omega_m > \Omega_N$, and the remaining exact case is $\Omega_m = \Omega_N$, where Lemma 1 gives $\Phi_{N+1}(\Omega_m) = 0$. Hence each nonterminal update preserves $\Omega_l < \Omega_N \le \Omega_r$ while halving the bracket, and the midpoints converge to Ω_N.

□

The comparison with the integral in Part 1 in fact identifies the limit of $\Omega_N/(N+1)^2$: the lower bound of Corollary 2 is asymptotically tight.

Lemma 8. *Let Ω_N be as in Lemma 1. Then*

$$\lim_{N\to\infty} \frac{\Omega_N}{(N+1)^2} = \frac{1}{\varpi^2}.$$

Proof. Write $\{\rho_k\}_{k=0}^{N+1}$ for the sequence of Lemma 1 at Ω_N, put $\ell_k \triangleq \rho_k - \rho_{k+1} > 0$, and set

$$g(x) \triangleq \frac{1}{\sqrt{x\,(1-x^2)}}, \quad 0 < x < 1,$$

so that the improper integral $\int_0^1 g(x)\,dx = \varpi$, as recorded in the proof of Corollary 2. As in Part 1 of that proof, dividing the step identity (25) at $\Omega = \Omega_N$ by $\sqrt{\rho_k(1-\rho_{k+1}^2)}$ and summing over $0 \le k \le N$ gives

$$\frac{N+1}{\sqrt{\Omega_N}} = \sum_{k=0}^{N} \frac{\ell_k}{\sqrt{\rho_k\,\left(1-\rho_{k+1}^2\right)}}. \tag{26}$$

We claim that the right-hand side converges to ϖ; the lemma follows, since then $\sqrt{\Omega_N}/(N+1) \to 1/\varpi$.

The intervals $[\rho_{k+1}, \rho_k]$, $0 \le k \le N$, partition $[0,1]$, and by the step identity and the lower bound of Corollary 2 their lengths satisfy $\delta_N \triangleq \max_{0\le k\le N} \ell_k \le 1/\sqrt{\Omega_N} < \varpi/(N+1)$, so the mesh δ_N tends to zero. For $x \in [\rho_{k+1}, \rho_k] \cap (0,1)$ we have $x \le \rho_k$ and $1-x^2 \le 1-\rho_{k+1}^2$, hence $g(x) \ge 1/\sqrt{\rho_k(1-\rho_{k+1}^2)}$, so the right-hand side of (26) is a lower Riemann-type sum for $\int_0^1 g(x)\,dx = \varpi$.

In particular, each summand is at most $\int_{\rho_{k+1}}^{\rho_k} g(x)\,dx$, so the sum is at most ϖ for every N. For the matching lower bound, fix $\varepsilon \in (0, 1/4]$. On $(0,\varepsilon]$ we have $1-x^2 \ge 3/4$, and on $[1-\varepsilon, 1)$ we have $x \ge 1/2$ and $1-x^2 \ge 1-x$, so the endpoint tails satisfy

$$\int_0^{\varepsilon} g(x)\,dx \le \frac{4}{\sqrt{3}}\sqrt{\varepsilon} \qquad \text{and} \qquad \int_{1-\varepsilon}^{1} g(x)\,dx \le 2\sqrt{2}\,\sqrt{\varepsilon}.$$

Once $\delta_N < \varepsilon/2$, every interval $[\rho_{k+1}, \rho_k]$ that meets $[\varepsilon, 1-\varepsilon]$ lies in $[\varepsilon/2, 1-\varepsilon/2]$, where the two-variable kernel $(x, y) \mapsto 1/\sqrt{x(1-y^2)}$ is uniformly continuous, with modulus of continuity ω_ε; since $g(x)$ is this kernel at (x, x), we get $1/\sqrt{\rho_k(1-\rho_{k+1}^2)} \geq g(x) - \omega_\varepsilon(\delta_N)$ for every $x \in [\rho_{k+1}, \rho_k]$. Discarding the remaining (positive) summands and summing over the intervals that meet $[\varepsilon, 1-\varepsilon]$, whose union covers $[\varepsilon, 1-\varepsilon]$ and has total length at most 1,

$$\sum_{k=0}^{N} \frac{\ell_k}{\sqrt{\rho_k \left(1-\rho_{k+1}^2\right)}} \geq \int_{\varepsilon}^{1-\varepsilon} g(x)\, dx - \omega_\varepsilon(\delta_N) \geq \varpi - \left(\frac{4}{\sqrt{3}} + 2\sqrt{2}\right)\sqrt{\varepsilon} - \omega_\varepsilon(\delta_N).$$

Letting $N \to \infty$ and then $\varepsilon \downarrow 0$ shows that the sum converges to ϖ.

Hence $(N+1)/\sqrt{\Omega_N} \to \varpi$. $\square$

A.2 Equivalence of the two forms of (LemniAcc)

Throughout this subsection, set $\phi_k \triangleq (1+\rho_k^2)/(2\rho_k)$ for $0 \leq k \leq N$, $\Delta_k \triangleq \phi_{k+1} - \phi_k$ for $0 \leq k \leq N-1$, and $\psi_k \triangleq (1+\rho_k^2)/(1-\rho_k^2)$ for $1 \leq k \leq N+1$ so that the update of x_{k+1} in (LemniAcc) reads $x_{k+1} = x_k^+ + (\psi_{k+1} - \psi_{k+2})\, z_{k+1}$ and the coefficients of (8) read $\alpha^{(\mathrm{init})} = \Omega_N \Delta_0 \Delta_{N-1}$, $\alpha_k^{(\mathrm{mom})} = \Delta_{N-k-1}/\Delta_{N-k}$, and $\alpha_k^{(\mathrm{corr})} = \Delta_{N-k-1}\left(\Omega_N \Delta_k - \frac{1}{\Delta_{N-k}}\right)$. Since $1 = \rho_0 > \rho_1 > \cdots > \rho_N > 0$ and the map $\rho \mapsto (1+\rho^2)/(2\rho)$ is strictly decreasing on $(0, 1]$, we have $\Delta_k > 0$ for $0 \leq k \leq N-1$.

Lemma 9. *Fix $N \geq 1$, and let Ω_N and $\rho_0, \ldots, \rho_{N+1}$ be as in Lemma 1. From any starting point $x_0 \in \mathbb{R}^d$, the iterates $x_0, x_1, \ldots, x_N$ of (LemniAcc) coincide with those of the momentum form (8).*

Proof. First, we claim that

$$\psi_k = \phi_{N+1-k}, \quad 1 \leq k \leq N+1. \tag{27}$$

Indeed, for $0 < \rho < 1$, the map $T(\rho) = (1-\rho)/(1+\rho)$ satisfies

$$\frac{1 + T(\rho)^2}{2T(\rho)} = \frac{(1+\rho)^2 + (1-\rho)^2}{2(1+\rho)(1-\rho)} = \frac{1+\rho^2}{1-\rho^2},$$

and combining this identity at $\rho = \rho_k$, which lies in $(0, 1)$ for $1 \leq k \leq N$, with the relation $\rho_{N+1-k} = T(\rho_k)$ recorded after Lemma 1 in Section 4.1 gives $\phi_{N+1-k} = \psi_k$ for $1 \leq k \leq N$. At the endpoint $k = N+1$, where $\rho_{N+1} = 0$ and $\rho_0 = 1$, direct evaluation gives $\psi_{N+1} = 1 = \phi_0$, which completes (27). By (27), $\psi_{k+1} - \psi_{k+2} = \phi_{N-k} - \phi_{N-k-1} = \Delta_{N-k-1}$ for $0 \leq k \leq N-1$, so the x-update of (LemniAcc) is

$$x_{k+1} = x_k^+ + \Delta_{N-k-1}\, z_{k+1}, \quad 0 \leq k \leq N-1. \tag{28}$$

We now argue by induction that the two forms generate the same iterates. They start from the same x_0, and whenever the iterates agree up to index k, both forms evaluate the gradient at the same points, so the vectors x_j^+ for $0 \leq j \leq k$ agree as well.

At $k = 0$: since $z_0 = 0$ and $-\nabla f(x_0)/L = x_0^+ - x_0$, the z-update of (LemniAcc) gives $z_1 = \Omega_N \Delta_0 \left(x_0^+ - x_0\right)$, and (28) yields $x_1 = x_0^+ + \Omega_N \Delta_0 \Delta_{N-1}\left(x_0^+ - x_0\right) = x_0^+ + \alpha^{(\mathrm{init})}\left(x_0^+ - x_0\right)$, the first update of (8).

For $1 \leq k \leq N-1$: solving (28) at index $k-1$ for the auxiliary variable gives $z_k = \left(x_k - x_{k-1}^+\right)/\Delta_{N-k}$, and the z-update gives $z_{k+1} = z_k + \Omega_N \Delta_k \left(x_k^+ - x_k\right)$. Substituting both into (28) and writing $x_k - x_{k-1}^+ = \left(x_k^+ - x_{k-1}^+\right) - \left(x_k^+ - x_k\right)$, we get

$$x_{k+1} = x_k^+ + \frac{\Delta_{N-k-1}}{\Delta_{N-k}}\left(x_k^+ - x_{k-1}^+\right) + \Delta_{N-k-1}\left(\Omega_N \Delta_k - \frac{1}{\Delta_{N-k}}\right)\left(x_k^+ - x_k\right),$$

which is the k-th update of (8) with the coefficients $\alpha_k^{(\mathrm{mom})}$ and $\alpha_k^{(\mathrm{corr})}$ above. By induction, the two forms produce the same iterates $x_0, x_1, \ldots, x_N$. □

A.3 Computer-assisted algorithm design by PEP

PEP formulation of the problem. This subsection presents, lightly edited for presentation, the Stage-1 derivation that bnb-pep-skill wrote for the design problem behind (LemniAcc). The skill instantiates the Stage 1 design problem of Section 3.1 for problem (7), in the notation of Section 4: the function class is $\mathcal{F} = \mathcal{F}_{0,L}$, the performance measure is $\mathcal{E} = \|\nabla f(x_N)\|^2$, the initial condition is $\mathcal{C} = \|x_0 - x_\star\|^2 - R^2 \leq 0$ for a given $R > 0$, and the method class is the class $\mathcal{M}_N$ of fixed-step first-order methods (FSFOMs) with N steps. Applied to $f \in \mathcal{F}_{0,L}$ from a starting point $x_0 \in \mathbb{R}^d$, an FSFOM produces its iterates with

$$x_i = x_{i-1} - \frac{1}{L}\sum_{j=0}^{i-1} h_{i,j} \nabla f(x_j)$$

for $1 \leq i \leq N$, where, relative to the generic form displayed in Section 3.1, the stepsizes $h_{i,j}$ are normalized by L. We can equivalently express the FSFOM as

$$x_i = x_0 - \frac{1}{L}\sum_{j=0}^{i-1} \alpha_{i,j} \nabla f(x_j),$$

where $\alpha_{i,j} \triangleq \sum_{k=j+1}^{i} h_{k,j}$ for $0 \leq j < i \leq N$; this triangular relationship between h and α is invertible, so we can compute h given α and vice versa. The stepsizes h or α may depend on $\mathcal{F}_{0,L}$ and the value of N, but are otherwise predetermined.

Instantiated with these choices, the worst-case performance $\mathcal{R}(M)$ of a method $M \in \mathcal{M}_N$ in (1) becomes

$$\mathcal{R}(M) = \left(\begin{array}{l} \text{maximize} \quad \|\nabla f(x_N)\|^2 \\ \text{subject to} \\ f \in \mathcal{F}_{0,L}, \\ x_\star \text{ is an optimal solution to (7) satisfying } \nabla f(x_\star) = 0, \\ \{x_i\}_{1 \leq i \leq N} \text{ is generated by } M \text{ with initial point } x_0, \\ \|x_0 - x_\star\|^2 \leq R^2, \\ x_\star = 0,\ f(x_\star) = 0 \end{array}\right), \qquad (\mathcal{O}^{\mathrm{inner}})$$

where f, $x_0, \ldots, x_N$, and $x_\star$ are the decision variables. We set $x_\star = 0$ and $f(x_\star) = 0$ without any loss of generality because the function class $\mathcal{F}_{0,L}$ is closed and invariant under shifting variables and function values.

The optimal FSFOM $M_N^\star \in \mathcal{M}_N$ for this setup solves the outer design problem $\mathcal{R}^\star(\mathcal{M}_N) = \min_{M\in\mathcal{M}_N} \mathcal{R}(M)$ of (1).

Following the BnB-PEP reformulation outlined in Section 3.1, the derivation reduces this design problem to a finite-dimensional nonconvex QCQP in two steps: it first formulates the inner problem ($\mathcal{O}^{\text{inner}}$) as a finite-dimensional convex SDP, and then optimizes the stepsizes against its dual [14].

As the first step, the derivation formulates the inner problem ($\mathcal{O}^{\text{inner}}$) as a finite-dimensional convex SDP using the smooth-convex interpolation result of [65, Corollary 1]. This result exactly characterizes the sampled function values and gradients that arise from a function in $\mathcal{F}_{0,L}$.

In other words, we impose the smooth-convex interpolation inequalities at the sampled points x_i, $i \in I_N^\star$, using (6), with the notation $g_i \triangleq \nabla f(x_i)$ and $f_i \triangleq f(x_i)$. This finite representation is lossless by [65, Corollary 1]. Thus, the derivation casts ($\mathcal{O}^{\text{inner}}$) as the following finite-dimensional but nonconvex problem:

$$\mathcal{R}(M) = \left(\begin{array}{ll} \text{maximize} & \|g_N\|^2 \\ \text{subject to} & f_i \geq f_j + \langle g_j, x_i - x_j \rangle + \frac{1}{2L} \|g_i - g_j\|^2, \quad i,j \in I_N^\star, \\ & g_\star = 0, \\ & x_i = x_0 - \frac{1}{L}\sum_{j=0}^{i-1} \alpha_{i,j} g_j, \quad 1 \leq i \leq N, \\ & \|x_0 - x_\star\|^2 \leq R^2, \\ & x_\star = 0,\ f_\star = 0 \end{array} \right), \tag{29}$$

where the decision variables are $\{x_i, g_i, f_i\}_{i\in I_N^\star} \subseteq \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}$.

Next, (29) is formulated as a convex SDP. Let $P \triangleq [x_0 \mid g_0 \mid g_1 \mid \ldots \mid g_N] \in \mathbb{R}^{d\times(N+2)}$, $G \triangleq P^\top P \in \mathbb{S}_+^{N+2}$, and $F \triangleq [f_0 \mid f_1 \mid \ldots \mid f_N] \in \mathbb{R}^{1\times(N+1)}$. Note that $\mathbf{rank}\, G \leq d$. Define the following notation for selecting columns and elements of P and F:

$$\begin{aligned} &\mathbf{g}_\star = 0 \in \mathbb{R}^{N+2},\ \mathbf{g}_i = e_{i+2} \in \mathbb{R}^{N+2}, \quad 0 \leq i \leq N, \\ &\mathbf{x}_0 = e_1 \in \mathbb{R}^{N+2},\ \mathbf{x}_\star = 0 \in \mathbb{R}^{N+2},\ \mathbf{x}_i = \mathbf{x}_0 - \frac{1}{L}\sum_{j=0}^{i-1} \alpha_{i,j}\mathbf{g}_j \in \mathbb{R}^{N+2}, \quad 1 \leq i \leq N, \\ &\mathbf{f}_\star = 0 \in \mathbb{R}^{N+1},\ \mathbf{f}_i = e_{i+1} \in \mathbb{R}^{N+1}, \quad 0 \leq i \leq N. \end{aligned}$$

This notation is defined so that $x_i = P\mathbf{x}_i,\ g_i = P\mathbf{g}_i,\ f_i = F\mathbf{f}_i$ for $i \in I_N^\star$. Note that $\mathbf{x}_i$ depends on $\{\alpha_{i,j}\}_{0\leq j\leq i-1}$ linearly for $1 \leq i \leq N$. For $i,j \in I_N^\star$, now define

$$\begin{aligned} A_{i,j}(\alpha) &= \mathbf{g}_j \odot (\mathbf{x}_i - \mathbf{x}_j) \in \mathbb{S}^{N+2}, \\ B_{i,j}(\alpha) &= (\mathbf{x}_i - \mathbf{x}_j) \odot (\mathbf{x}_i - \mathbf{x}_j) \in \mathbb{S}_+^{N+2}, \\ C_{i,j} &= (\mathbf{g}_i - \mathbf{g}_j) \odot (\mathbf{g}_i - \mathbf{g}_j) \in \mathbb{S}_+^{N+2}. \end{aligned} \tag{30}$$

Note that $A_{i,j}(\alpha)$ is affine and $B_{i,j}(\alpha)$ is quadratic as functions of $\{\alpha_{i,j}\}_{0\leq j<i\leq N}$. Under this notation, $\langle g_j, x_i - x_j \rangle = \mathbf{tr} G A_{i,j}(\alpha)$, $\|x_0 - x_\star\|^2 = \mathbf{tr} G B_{0,\star}$, and $\|g_i - g_j\|^2 = \mathbf{tr} G C_{i,j}$ for $i,j \in I_N^\star$. In these terms, (29) becomes a maximization over F and $G = P^\top P$, at the price of the rank constraint $\mathbf{rank}\, G \leq d$: the equivalence relies on the fact that any $G \in \mathbb{S}_+^{N+2}$ with $\mathbf{rank}\, G \leq d$

factors as $G = P^\top P$ with $P \in \mathbb{R}^{d\times(N+2)}$; see [65, §3.2]. The rank constraint is removed with the following large-scale assumption.

Assumption 1. *We have $d \geq N+2$.*

Under this assumption, the constraint $\mathbf{rank}\, G \leq d$ becomes vacuous, since $G \in \mathbb{S}_+^{N+2}$. The bnb-pep-skill derivation drops the rank constraint and formulates ($\mathcal{O}^{\text{inner}}$) as a convex SDP

$$\mathcal{R}(M) = \left(\begin{array}{l} \text{maximize} \quad \mathbf{tr} G\left(C_{N,\star}\right) \\ \text{subject to} \\ F(\mathbf{f}_j - \mathbf{f}_i) + \mathbf{tr} G\Big[A_{i,j}(\alpha) + \frac{1}{2L} C_{i,j}\Big] \leq 0, \quad i,j \in I_N^\star : i \neq j, \quad \triangleright\, \text{dual var. } \lambda_{i,j} \geq 0 \\ -G \preceq 0, \quad \triangleright\, \text{dual var. } Z \succeq 0 \\ \mathbf{tr} G B_{0,\star} \leq R^2, \quad \triangleright\, \text{dual var. } \nu \geq 0 \end{array} \right), \tag{31}$$

where $F \in \mathbb{R}^{N+1}$, $G \in \mathbb{S}^{N+2}$ are the decision variables. We denote the corresponding dual variables on the right-hand side of the constraints for later use. Note that (31) is free from problem dimension d.

Next the bnb-pep-skill derivation uses convex duality to formulate the SDP (31) in the previous step, originally a maximization problem, as a minimization problem. Taking the dual of (31) gives

$$\overline{\mathcal{R}}(M) = \left(\begin{array}{l} \text{minimize} \quad \nu R^2 \\ \text{subject to} \\ \sum_{i,j \in I_N^\star : i \neq j} \lambda_{i,j}(\mathbf{f}_j - \mathbf{f}_i) = 0, \\ \nu B_{0,\star} - C_{N,\star} + \sum_{i,j\in I_N^\star : i\neq j} \lambda_{i,j}\left[A_{i,j}(\alpha) + \frac{1}{2L} C_{i,j}\right] = Z, \\ Z \succeq 0, \\ \nu \geq 0,\ \lambda_{i,j} \geq 0, \quad i,j \in I_N^\star : i \neq j \end{array} \right), \tag{32}$$

where $\nu \in \mathbb{R}$, $\lambda = \{\lambda_{i,j}\}_{i,j\in I_N^\star : i \neq j}$, and $Z \in \mathbb{S}_+^{N+2}$ are the decision variables. We call λ, ν, and Z the *inner-dual variables*. By weak duality of convex SDPs, we have $\mathcal{R}(M) \leq \overline{\mathcal{R}}(M)$.

In convex SDPs, strong duality holds often but not always. For the sake of simplicity, we will assume strong duality holds in our setup.

Assumption 2. *Strong duality holds between* (31) *and* (32)*, i.e., $\mathcal{R}(M) = \overline{\mathcal{R}}(M)$.*

As the second step, with the inner problem ($\mathcal{O}^{\text{inner}}$) replaced by its dual minimization (32), the outer design problem becomes a joint minimization over the stepsizes $\{\alpha_{i,j}\}_{0\leq j<i\leq N}$ and the inner-dual variables. This joint problem is nonconvex, and, following [14], the derivation formulates it as a QCQP by replacing the semidefinite constraint $Z \succeq 0$ with the Cholesky factorization $Z = VV^\top$, where V is lower triangular with nonnegative diagonals [26, Corollary 7.2.9]. The problem of finding

the optimal FSFOM then takes the QCQP form

$$\mathcal{R}^\star(\mathcal{M}_N) = \left(\begin{array}{l} \text{minimize} \quad \nu R^2 \\ \text{subject to} \\ \sum_{i,j \in I_N^\star : i \neq j} \lambda_{i,j}(\mathbf{f}_j - \mathbf{f}_i) = 0, \\ \nu B_{0,\star} - C_{N,\star} + \sum_{i,j \in I_N^\star : i \neq j} \lambda_{i,j} \left[A_{i,j}(\alpha) + \frac{1}{2L} C_{i,j} \right] = Z, \\ V \text{ is lower triangular with nonnegative diagonals,} \\ Z = VV^\top, \\ \nu \geq 0, \ \lambda_{i,j} \geq 0, \quad i,j \in I_N^\star : i \neq j \end{array} \right), \tag{33}$$

where $\lambda = \{\lambda_{i,j}\}_{i,j \in I_N^\star : i \neq j}$, ν, Z, V, $\{\alpha_{i,j}\}_{0 \leq j < i \leq N}$ are the decision variables.

Upon user approval of the derivation above, bnb-pep-skill solves (33) numerically; the resultant optimal $\alpha^\star$ gives the optimal stepsizes $h^\star$ for any large-scale setup $d \geq N + 2$. By the scale invariance discussed in [65, §3.5], it suffices to solve the QCQP for $L = 1$ and $R = 1$: for any other $L > 0$ and $R > 0$ the optimal stepsizes are unchanged, and the optimal worst-case value scales by the factor $L^2 R^2$.

In Stage 2 of the pipeline, the frontier-llm-consult skill fitted symbolic expressions to the numerical solutions of (33), yielding an analytic parametrization of the numerically optimal stepsizes in the design variables α, equivalently h, together with an analytic feasible point of the inner dual (32), whose feasibility already certifies the worst-case bound by weak duality. From this stepsize parametrization, the consultation then recast the method into the memory-efficient form (LemniAcc) presented in Section 4.1. In continued consultation, Stage 2 converted the multipliers of this feasible point into the Lyapunov proof of Theorem 1 given in Section 4.1.1, the form in which this paper presents the convergence analysis; the Lean coverage of the presented results is described in Section 4.1.5.

A.4 Details for continuous-time analysis of lemniscate ODE

In this section, we develop a leading-order continuous-time model of the lemniscate acceleration algorithm and prove Theorem 2. For an L_f-smooth objective with $L_f > 0$, we apply the method with the algorithmic smoothness upper bound $\widehat{L}_N = N^2/T^2$ and use the iterate grid $t = kT/N$. At fixed L_f and T, the condition $L_f \leq \widehat{L}_N$ holds for all sufficiently large N. We retain only the leading-order terms as $N \to \infty$.

Recall the continuous-time coefficients $\rho(t)$ and $\sigma(t)$ are given by

$$\begin{aligned} \rho(t) &= \mathrm{cl}^2\left(\frac{\varpi_T}{2} t\right), \\ \sigma(t) &= \sqrt{\frac{1 - \rho(t)^2}{\rho(t)}}, \end{aligned} \tag{34}$$

where $\varpi_T \triangleq \varpi/T$ and $\mathrm{cl}(x)$ is the lemniscate cosine function in Section 4.1. We prove some identities of $\rho(t)$ and $\sigma(t)$ used in the derivation of the ODE and the Lyapunov function.

We first collect the properties of the lemniscate elliptic functions that these derivations use. Recall from Section 4.1.3 that $\mathrm{arcsl}(u) = \int_0^u dx/\sqrt{1 - x^4}$, that sl is the inverse of arcsl, and that $\mathrm{cl}(x) = \mathrm{sl}(\varpi/2 - x)$.

Lemma 10 (Lemniscatic calculus)**.** *The following hold.*

1. sl *is a continuous strictly increasing bijection of* $[0, \varpi/2]$ *onto* $[0, 1]$ *with* $\mathrm{sl}(0) = 0$ *and* $\mathrm{sl}(\varpi/2) = 1$*; consequently* cl *is a continuous strictly decreasing bijection of* $[0, \varpi/2]$ *onto* $[0, 1]$*, and* $\mathrm{sl}(x), \mathrm{cl}(x) \in (0, 1)$ *for* $x \in (0, \varpi/2)$*.*
2. $\mathrm{sl}'(x) = \sqrt{1 - \mathrm{sl}^4(x)}$ *and* $\mathrm{cl}'(x) = -\sqrt{1 - \mathrm{cl}^4(x)}$ *for* $x \in (0, \varpi/2)$*.*
3. $\mathrm{cl}(\varpi/2 - x) = \mathrm{sl}(x)$ *for* $x \in [0, \varpi/2]$*.*
4. $\mathrm{cl}^2(x) = \dfrac{1 - \mathrm{sl}^2(x)}{1 + \mathrm{sl}^2(x)}$ *for* $x \in [0, \varpi/2]$*; equivalently,* $\mathrm{cl}^2(x) + \mathrm{sl}^2(x) + \mathrm{cl}^2(x)\,\mathrm{sl}^2(x) = 1$*.*
5. $\mathrm{sl}(x) = x + o(x)$ *as* $x \downarrow 0$*.*
6. *For* $T > 0$*, the coefficient (34) satisfies* $0 < \rho(t) < 1$ *for* $t \in (0, T)$*. Its symmetry identity* $\rho(T - t) = (1 - \rho(t)) \,/\, (1 + \rho(t))$ *holds for* $t \in [0, T]$*.*

Proof. Part 1. On $[0, 1)$ the integrand $1/\sqrt{1 - x^4}$ is positive, and near $x = 1$ it is bounded by $1/\sqrt{1 - x}$, so arcsl is a continuous strictly increasing bijection of $[0, 1]$ onto $[0, \varpi/2]$; here $\mathrm{arcsl}(1) = \varpi/2$ is the definition of ϖ. Its inverse sl inherits these properties, and $\mathrm{cl}(x) = \mathrm{sl}(\varpi/2 - x)$ inherits the rest.

Part 2. For $x \in (0, \varpi/2)$ the inverse function theorem gives $\mathrm{sl}'(x) = 1/\,\mathrm{arcsl}'(\mathrm{sl}(x)) = \sqrt{1 - \mathrm{sl}^4(x)}$, and the chain rule gives $\mathrm{cl}'(x) = -\,\mathrm{sl}'(\varpi/2 - x) = -\sqrt{1 - \mathrm{cl}^4(x)}$.

Part 3. Substitute $\varpi/2 - x$ for x in the definition $\mathrm{cl}(x) = \mathrm{sl}(\varpi/2 - x)$.

Part 4. For $u \in [0, 1]$ set $v(u) \triangleq \sqrt{(1 - u^2)/(1 + u^2)}$, so that $v(u)(1 + u^2) = \sqrt{1 - u^4}$ and $\sqrt{1 - v(u)^4} = 2u/(1 + u^2)$. For $u \in (0, 1)$,

$$v'(u) = \frac{-2u}{v(u)\,(1 + u^2)^2}, \qquad \text{so} \qquad \frac{v'(u)}{\sqrt{1 - v(u)^4}} = \frac{-1}{v(u)\,(1 + u^2)} = \frac{-1}{\sqrt{1 - u^4}}.$$

Hence $\frac{d}{du}\,[\mathrm{arcsl}(u) + \mathrm{arcsl}(v(u))] = 0$ on $(0, 1)$, and by continuity and the value $\mathrm{arcsl}(0) + \mathrm{arcsl}(1) = \varpi/2$ at $u = 0$,

$$\mathrm{arcsl}(u) + \mathrm{arcsl}\,(v(u)) = \frac{\varpi}{2}, \qquad u \in [0, 1].$$

Taking $u = \mathrm{sl}(x)$ for $x \in [0, \varpi/2]$ gives $\mathrm{arcsl}(v(\mathrm{sl}(x))) = \varpi/2 - x$, that is, $v(\mathrm{sl}(x)) = \mathrm{sl}(\varpi/2 - x) = \mathrm{cl}(x)$ by part 3. Squaring yields the first identity, and clearing denominators yields the second.

Part 5. By part 1, $\mathrm{sl}(0) = 0$, and by the mean value theorem $\mathrm{sl}(x) = \mathrm{sl}'(\xi_x)\,x$ for some $\xi_x \in (0, x)$, where $\mathrm{sl}'(\xi_x) = \sqrt{1 - \mathrm{sl}^4(\xi_x)} \to 1$ as $x \downarrow 0$ by part 1 and continuity.

Part 6. For $t \in (0, T)$ we have $\varpi_T t/2 \in (0, \varpi/2)$, so $\rho(t) = \mathrm{cl}^2(\varpi_T t/2) \in (0, 1)$ by part 1. For the symmetry, parts 3 and 4 give

$$\rho(T - t) = \mathrm{cl}^2\left(\frac{\varpi}{2} - \frac{\varpi_T t}{2}\right) = \mathrm{sl}^2\left(\frac{\varpi_T t}{2}\right) = \frac{1 - \mathrm{cl}^2(\varpi_T t/2)}{1 + \mathrm{cl}^2(\varpi_T t/2)} = \frac{1 - \rho(t)}{1 + \rho(t)}. \qquad \square$$

Lemma 11. *The following identities hold for* $t \in (0, T)$*:*

$$\dot{\rho}(t) = -\varpi_T \sqrt{\rho(t)(1 - \rho(t)^2)} = -\varpi_T \sigma(t) \rho(t),$$

$$
\begin{aligned}
\dot\sigma(t) &= \frac{\varpi_T}{2}\left(\rho(t) + \frac{1}{\rho(t)}\right),\\
\ddot\sigma(t) &= \frac{\varpi_T^2}{2}\sigma(t)^3,\\
\sigma(t)\,\sigma(T-t) &= 2.
\end{aligned}
$$

Proof. Throughout the proof, $0 < \rho(t) < 1$ and hence $\sigma(t) > 0$ on $(0, T)$, by part 6 of Lemma 10, so every division below is legitimate. By the derivative $\mathrm{cl}'(x) = -\sqrt{1-\mathrm{cl}^4(x)}$ (part 2 of Lemma 10, with $\mathrm{cl} \geq 0$ from part 1) and (34), $\rho(t)$ satisfies the first identity: $\dot\rho(t) = -\varpi_T\sqrt{\rho(t)(1-\rho(t)^2)} = -\varpi_T\sigma(t)\rho(t)$. Differentiating $\sigma(t)^2 = 1/\rho(t) - \rho(t)$ and using the first identity gives

$$
2\sigma(t)\dot\sigma(t) = -\left(1 + \frac{1}{\rho(t)^2}\right)\dot\rho(t) = \varpi_T\left(1 + \frac{1}{\rho(t)^2}\right)\rho(t)\sigma(t),
$$

and dividing both sides by $2\sigma(t)$ gives the second identity. Differentiating the second identity and using the first identity gives the third identity:

$$
\ddot\sigma(t) = \frac{\varpi_T}{2}\left(1 - \frac{1}{\rho(t)^2}\right)\dot\rho(t) = \frac{\varpi_T^2}{2}\frac{(1-\rho(t)^2)^{3/2}}{\rho(t)^{3/2}} = \frac{\varpi_T^2}{2}\sigma(t)^3.
$$

For the last identity, we use the symmetry $\rho(T-t) = (1-\rho(t))\,/\,(1+\rho(t))$ from part 6 of Lemma 10. Thus

$$
\sigma(T-t)^2 = \frac{1-\rho(T-t)^2}{\rho(T-t)} = \frac{4\rho(t)}{1-\rho(t)^2} = \frac{4}{\sigma(t)^2},
$$

and since $\sigma(t) > 0$, this proves the last identity. □

The continuous-time ODE. We sketch a derivation of the ODE by identifying the leading-order behavior of (LemniAcc) as $N \to \infty$ with $\widehat{L}_N = N^2/T^2$. On the iterate grid $t_k^x = k\,T/N$, we identify $X(t_k^x) \approx x_k$ and $Z(t_k^x) \approx \big(N/(\sqrt{\Omega_N}\,T)\big)\,z_k$. The coefficient ρ_k is naturally placed at $t_k^\rho = k\,T/(N{+}1)$; for $0 \leq k \leq N$, $|t_k^x - t_k^\rho| \leq T/(N+1) = O(1/N)$, so we identify $\rho(t_k^x) \approx \rho_k$ at leading order. The updates of (LemniAcc) are

$$
\begin{aligned}
z_{k+1} &= z_k - \Omega_N\Big(\frac{1+\rho_{k+1}^2}{2\rho_{k+1}} - \frac{1+\rho_k^2}{2\rho_k}\Big)\frac{1}{\widehat{L}_N}\nabla f(x_k),\\
x_k^+ &= x_k - \frac{1}{\widehat{L}_N}\nabla f(x_k),\\
x_{k+1} &= x_k^+ + \Big(\frac{1+\rho_{k+1}^2}{1-\rho_{k+1}^2} - \frac{1+\rho_{k+2}^2}{1-\rho_{k+2}^2}\Big)z_{k+1}.
\end{aligned}
$$

The z-update coefficient, after division by $\sqrt{\Omega_N}$ to account for the rescaling of Z, has the following leading-order form:

$$\sqrt{\Omega_N}\,\frac{1+x^2}{2x}\bigg|_{\rho_k}^{\rho_{k+1}} \approx \sqrt{\Omega_N}(\rho_k-\rho_{k+1})\frac{1-\rho_k^2}{2\rho_k^2} \approx \sqrt{\rho(1-\rho^2)}\frac{1-\rho^2}{2\rho^2} = \frac{1}{2}\left(\frac{1-\rho^2}{\rho}\right)^{3/2} = \frac{1}{2}\sigma^3,$$

which follows from the mean value theorem, the continuity of $\rho(t)$, and the recurrence (9). For the x-update, define $\phi_k = (1+\rho_k^2)/(2\rho_k)$. The symmetry $\rho_{N+1-k} = (1-\rho_k)/(1+\rho_k)$ from Section 4.1 rewrites the coefficient in the x-update as $\phi_{N-k}-\phi_{N-k-1}$. Applying the preceding calculation at $T-t$ gives $\sqrt{\Omega_N}(\phi_{N-k}-\phi_{N-k-1}) \approx \sigma(T-t)^3/2$. Since $1/\widehat{L}_N = T^2/N^2$ and $\Delta t = T/N$, the updates give $\Delta Z \approx -(\sigma^3/2)\nabla f(X)\,\Delta t$ and $\Delta X \approx (\sigma(T-t)^3/2)Z\,\Delta t$ at leading order. The direct gradient step $x_k^+ - x_k$ carries the smaller factor $1/\widehat{L}_N = O(\Delta t^2)$ and hence does not contribute at this order.

This development suggests the coupled ODE for $X(t)$ and $Z(t)$ on $(0,T)$:

$$\begin{aligned}\dot{Z}(t) &= -\frac{\sigma(t)^3}{2}\nabla f(X(t)),\\ \dot{X}(t) &= \frac{\sigma(T-t)^3}{2}Z(t) = \frac{4}{\sigma(t)^3}Z(t).\end{aligned}$$

Differentiating $Z(t) = \sigma(t)^3\dot{X}(t)/4$ and using $\dot{Z}(t) = -\sigma(t)^3\nabla f(X(t))/2$ gives the following second-order ODE for $X(t)$:

$$\ddot{X}(t) + \gamma(t)\dot{X}(t) + 2\nabla f\left(X(t)\right) = 0, \quad 0 < t < T, \tag{35}$$

where $\gamma(t) \triangleq 3\dot{\sigma}(t)/\sigma(t)$ is the coefficient of the friction term.

Lemma 12. *The ODE in (35) can be written as*

$$\ddot{X}(t) + \frac{3\varpi}{2T}\left[\frac{\mathrm{cl}\left(\frac{\varpi t}{2T}\right)}{\mathrm{sl}\left(\frac{\varpi t}{2T}\right)} + \frac{\mathrm{sl}\left(\frac{\varpi t}{2T}\right)}{\mathrm{cl}\left(\frac{\varpi t}{2T}\right)}\right]\dot{X}(t) + 2\nabla f\left(X(t)\right) = 0, \quad 0 < t < T.$$

Proof. We only need to calculate the friction term $\gamma(t) = 3\dot{\sigma}(t)/\sigma(t)$ in (35). By the second identity of Lemma 11, we have

$$\frac{\dot{\sigma}}{\sigma} = \frac{\varpi_T}{2}\frac{\rho+\rho^{-1}}{\sqrt{(1-\rho^2)/\rho}} = \frac{\varpi_T}{2}\frac{1+\rho^2}{\sqrt{\rho(1-\rho^2)}}.$$

Substituting $\rho(t) = \mathrm{cl}^2(\varpi_T t/2)$ gives

$$\begin{aligned}\gamma(t) &= \frac{3\varpi_T}{2}\frac{1+\mathrm{cl}^4(\varpi_T t/2)}{\mathrm{cl}(\varpi_T t/2)\sqrt{1-\mathrm{cl}^4(\varpi_T t/2)}}\\ &= \frac{3\varpi_T}{2}\frac{1+\mathrm{cl}^4(\varpi_T t/2)}{\mathrm{cl}(\varpi_T t/2)\,\mathrm{sl}(\varpi_T t/2)\left(1+\mathrm{cl}^2(\varpi_T t/2)\right)}\\ &= \frac{3\varpi_T}{2}\frac{\mathrm{cl}^2(\varpi_T t/2)+\mathrm{sl}^2(\varpi_T t/2)}{\mathrm{cl}(\varpi_T t/2)\,\mathrm{sl}(\varpi_T t/2)}\\ &= \frac{3\varpi_T}{2}\left[\frac{\mathrm{cl}(\varpi_T t/2)}{\mathrm{sl}(\varpi_T t/2)} + \frac{\mathrm{sl}(\varpi_T t/2)}{\mathrm{cl}(\varpi_T t/2)}\right].\end{aligned}$$

where we use $\mathrm{cl}^2(x)+\mathrm{sl}^2(x)+\mathrm{cl}^2(x)\mathrm{sl}^2(x)=1$ (part 4 of Lemma 10, which also gives $\sqrt{1-\mathrm{cl}^4}=\mathrm{sl}\,(1+\mathrm{cl}^2)$ for the second equality) to simplify the form. ☐

Proof of Theorem 2. Assume that $X\in C^2([0,T];\mathbb{R}^d)$. Denote $x_T\triangleq X(T)$ and $\varpi_T\triangleq\varpi/T$, and set $Z(t)\triangleq\sigma(t)^3\dot{X}(t)/4$ for $t\in(0,T)$. Then the second-order ODE and the definition of γ imply $\dot{Z}(t)=-\sigma(t)^3\nabla f(X(t))/2$. The Lyapunov function $\mathcal{V}(t)$ is defined for $t\in(0,T)$ as

$$\begin{aligned}\mathcal{V}(t)\triangleq{}&\underbrace{\frac{1-\rho^2}{2\rho}\left(f(X(t))-f(x_\star)\right)-\frac{(1-\rho)^2}{2\rho}\left(f(x_T)-f(x_\star)\right)}_{A(t)}\\&+\underbrace{\frac{\varpi_T^2}{2}\left(\left\|X(t)-x_\star+\frac{1+\rho^2}{1-\rho^2}\frac{Z(t)}{\varpi_T}\right\|^2-\left\|x_T-x_\star+\frac{Z(t)}{\varpi_T}\right\|^2\right)}_{B(t)}.\end{aligned}\tag{36}$$

Here and below, ρ, σ, X, Z, and their derivatives are evaluated at t when their arguments are omitted. We calculate the derivative of function-value terms $A(t)$ and squared-norm terms $B(t)$ separately. The function-value terms can be simplified as follows:

$$\begin{aligned}A(t)&=\frac{1-\rho^2}{2\rho}\left(f(X)-f(x_\star)\right)-\frac{(1-\rho)^2}{2\rho}\left(f(x_T)-f(x_\star)\right)\\&=(1-\rho)\left(f(X)-f(x_\star)\right)+\frac{(1-\rho)^2}{2\rho}\left(f(X)-f(x_T)\right).\end{aligned}$$

Using the identity $\dot{\rho}=-\varpi_T\sigma\rho$ of Lemma 11 and $\sigma^2=\rho^{-1}-\rho$, we have

$$\frac{d}{dt}\left(\frac{(1-\rho)^2}{2\rho}\right)=\frac{\varpi_T\sigma^3}{2},$$

so the derivative of the function-value terms is

$$\dot{A}(t)=\varpi_T\sigma\rho\left(f(X)-f(x_\star)\right)+\frac{\varpi_T\sigma^3}{2}\left(f(X)-f(x_T)\right)+\frac{2}{\sigma}\left\langle Z,\nabla f(X)\right\rangle.$$

For the squared-norm terms $B(t)$, the identities of Lemma 11 give

$$\frac{d}{dt}\left(\frac{1+\rho^2}{1-\rho^2}\right)=-\frac{4\varpi_T}{\sigma^3}.$$

Consequently,

$$\frac{d}{dt}\left(X-x_\star+\frac{1+\rho^2}{1-\rho^2}\frac{Z}{\varpi_T}\right)=\frac{1+\rho^2}{1-\rho^2}\frac{\dot{Z}}{\varpi_T},$$

where $\dot{X}=4Z/\sigma^3$ cancels $-4Z/\sigma^3$. Differentiating the squared-norm terms $B(t)$, substituting $\dot{Z}=-\sigma^3\nabla f(X)/2$, and using $\sigma^2=\rho^{-1}-\rho$ gives

$$\dot{B}(t)=-\varpi_T\sigma\rho\left\langle X-x_\star,\nabla f(X)\right\rangle-\frac{\varpi_T\sigma^3}{2}\left\langle X-x_T,\nabla f(X)\right\rangle-\frac{2}{\sigma}\left\langle Z,\nabla f(X)\right\rangle.$$

Adding the two derivatives $\dot{A}(t)$ and $\dot{B}(t)$, the terms involving $\langle Z, \nabla f(X)\rangle$ cancel, and we obtain $\dot{\mathcal{V}}(t) = -\varpi_T \sigma\rho D_f(x_\star, X) - (\varpi_T\sigma^3/2)D_f(x_T, X) \le 0$, where $D_f(y,x) \triangleq f(y) - f(x) - \langle y-x, \nabla f(x)\rangle$, and the inequality follows from convexity of f, in the form $D_f(y,x) \ge 0$. Thus $\mathcal{V}(t)$ is non-increasing on $(0,T)$.

For limits of the endpoints $t \downarrow 0$ and $t \uparrow T$, we substitute $Z = \sigma^3 \dot{X}/4$ and expand the two squared norms in (36) to expose the cancellation between potentially singular terms as $t \uparrow T$. The expanded form of (36) is:

$$\begin{aligned} \mathcal{V}(t) = {} & (1-\rho)\left(f(X) - f(x_\star)\right) + \frac{(1-\rho)^2}{2\rho}\left(f(X) - f(x_T)\right) \\ & + \frac{\varpi_T^2}{2}\left(\|X - x_\star\|^2 - \|x_T - x_\star\|^2\right) + \frac{\sigma^2}{8}\left\|\dot{X}\right\|^2 \\ & + \frac{\varpi_T \sigma\rho}{2}\left\langle X - x_\star, \dot{X}\right\rangle + \frac{\varpi_T\sigma^3}{4}\left\langle X - x_T, \dot{X}\right\rangle . \end{aligned} \tag{37}$$

For the limit $t \downarrow 0$, $\rho(t) \to 1$, $\sigma(t) \to 0$, $X(t) \to x_0$, and $\dot{X}(t) \to 0$, so $\mathcal{V}(t)$ has the limit:

$$\lim_{t\downarrow 0} \mathcal{V}(t) = \frac{\varpi_T^2}{2}\left(\|x_0 - x_\star\|^2 - \|x_T - x_\star\|^2\right). \tag{38}$$

For the limit $t \uparrow T$, denote $\delta = T - t$ and $g_T \triangleq \nabla f(x_T)$. The leading order terms of the coefficients ρ, σ, and γ as $\delta \downarrow 0$ are

$$\begin{aligned} \rho(T-\delta) &= \frac{\varpi_T^2}{4}\delta^2 + o(\delta^2), \\ \sigma(T-\delta) &= \frac{2}{\varpi_T \delta} + o(\delta^{-1}), \\ \gamma(T-\delta) &= \frac{3}{\delta} + o(\delta^{-1}). \end{aligned} \tag{39}$$

Indeed, the relation $\mathrm{cl}\left(\varpi_T(T-\delta)/2\right) = \mathrm{sl}(\varpi_T\delta/2)$ and $\mathrm{sl}(x) = x + o(x)$ (parts 3 and 5 of Lemma 10) gives the expression for ρ; the other two follow from the definitions of σ and γ. Moreover, differentiating the identity $\sigma(t)\,\sigma(T-t) = 2$ of Lemma 11 gives $\dot{\sigma}(t)/\sigma(t) = \dot{\sigma}(T-t)/\sigma(T-t)$, so $\gamma(t) = \gamma(T-t)$; hence $\gamma(t) = 3/t + o(t^{-1})$ as $t \downarrow 0$ as well.

We next obtain the corresponding expansion of the trajectory. Since $X \in C^2([0,T])$, the right-hand side of $\gamma(t)\dot{X}(t) = -\ddot{X}(t) - 2\nabla f(X(t))$ is bounded as $t \uparrow T$. Because $\gamma(t) \to \infty$, this forces $\dot{X}(T) = 0$. Taylor expansion of $\dot{X}(t)$ gives $\dot{X}(T-\delta) = -\ddot{X}(T)\delta + o(\delta)$. Hence the left-hand side above converges to $-3\ddot{X}(T)$, whereas its right-hand side converges to $-\ddot{X}(T) - 2g_T$. Thus $\ddot{X}(T) = g_T$, and further Taylor expansion gives

$$\begin{aligned} \dot{X}(T-\delta) &= -g_T\delta + o(\delta), \\ X(T-\delta) &= x_T + \frac{1}{2}g_T\delta^2 + o(\delta^2), \\ f\left(X(T-\delta)\right) - f(x_T) &= \frac{1}{2}\|g_T\|^2\delta^2 + o(\delta^2). \end{aligned} \tag{40}$$

Using (39) and (40), the limit of $\mathcal{V}(t)$ in (37) is calculated term by term:

$$\begin{aligned}\lim_{t\uparrow T}\mathcal{V}(t) &= f(x_T) - f(x_\star) + \frac{\|g_T\|^2}{\varpi_T^2} + 0 + \frac{\|g_T\|^2}{2\varpi_T^2} + 0 - \frac{\|g_T\|^2}{\varpi_T^2} \\ &= \frac{1}{2\varpi_T^2}\|\nabla f(x_T)\|^2 + f(x_T) - f(x_\star).\end{aligned} \tag{41}$$

For any $0 < s < t < T$, monotonicity gives $\mathcal{V}(s) \geq \mathcal{V}(t)$. Letting $s \downarrow 0$ and then $t \uparrow T$ in this inequality, (38) and (41) gives the stronger estimate

$$\|\nabla f(x_T)\|^2 + 2\varpi_T^2 (f(x_T) - f(x_\star)) + \varpi_T^4 \|x_T - x_\star\|^2 \leq \varpi_T^4 \|x_0 - x_\star\|^2.$$

Dropping nonnegative terms and substituting $\varpi_T = \varpi/T$ proves both stated bounds. □

B Deferred proofs and details for ITEM-f

B.1 Existence and uniqueness of the ITEM-f construction

We first derive the target angle and describe the shooting computation of the ITEM-f coefficients, then prove that it produces the unique construction of Lemma 4. Throughout, $N \geq 1$ and $q \in (0,1)$ are fixed, $\Upsilon > 1$ denotes a candidate value of the constant, and we write

$$C(\Upsilon) \triangleq (\Upsilon, 0), \quad R(\Upsilon) \triangleq \sqrt{\Upsilon^2 - 1}, \quad p(\Upsilon) \triangleq \frac{q\,R(\Upsilon)}{\Upsilon} = q\sqrt{1-\Upsilon^{-2}},$$

abbreviated to C, R, and p when the argument is clear. The map $\Upsilon \mapsto 1 - \Upsilon^{-2}$ is continuous and strictly increasing on $(1,\infty)$; hence so is p, with $0 < p(\Upsilon) < q$, $p(\Upsilon) \to 0$ as $\Upsilon \downarrow 1$, and $p(\Upsilon) \to q$ as $\Upsilon \to \infty$.

Target angle. For a prospective admissible configuration at a candidate $\Upsilon > 1$, write $P_1 - C = R(\cos\theta_1, \sin\theta_1)$ with $\theta_1 \in (0,\pi)$. The endpoint formula gives $P_0 - C = \left(-R^2/\Upsilon,\ \sqrt{1-q}\,R/\Upsilon\right)$, so the inner-product condition at $k = 0$ evaluates on both sides; equating the two evaluations and simplifying (tedious but straightforward algebra) gives

$$\sin\theta_1 = \sqrt{1-q}\,(\Upsilon + R\cos\theta_1) = \sqrt{1-q}\,a_1. \tag{42}$$

Since $b_1 = R\sin\theta_1$, this says that P_1 lies on the line $b = \sqrt{1-q}\,R\,a$ through the origin and P_0. Substitution into $(a-\Upsilon)^2 + b^2 = R^2$, using $\Upsilon^2 - R^2 = 1$, gives the quadratic $\left(1 + (1-q)R^2\right)a^2 - 2\Upsilon a + 1 = 0$, whose roots are $a_\pm = \left(\Upsilon \pm \sqrt{q}\,R\right) / \left(1 + (1-q)R^2\right)$. The quadratic is negative at $a_0 = 1/\Upsilon$, so $a_- < a_0 < a_+$. Hence the ordering constraint selects the point $(a_+, \sqrt{1-q}\,R(\Upsilon)a_+)$. Its angle about $C(\Upsilon)$ is the *target angle* $\vartheta(\Upsilon) \in (0,\pi)$, where

$$\cos\vartheta(\Upsilon) \triangleq \frac{a_+ - \Upsilon}{R} = \frac{\sqrt{q} - (1-q)R\,\Upsilon}{1 + (1-q)R^2}. \tag{43}$$

We also define $\vartheta(\Upsilon)$ for any $\Upsilon > 1$ by this formula, even if the corresponding construction does not exist.

Shooting computation. For each candidate $\Upsilon > 1$, define its backward shooting orbit by

$$\begin{aligned} &\theta_N(\Upsilon) \triangleq \arccos(-\sqrt{q}), \\ &\theta_k(\Upsilon) \triangleq \theta_{k+1}(\Upsilon) + \arccos(1 - q - p(\Upsilon)\cos\theta_{k+1}(\Upsilon)), \quad k = N-1, \dots, 1. \end{aligned} \tag{44}$$

Every admissible construction obeys this recurrence. Indeed, if a candidate Υ admits the required points, then every intermediate point has a unique representation $P_k - C(\Upsilon) = R(\Upsilon)(\cos\theta_k, \sin\theta_k)$ with $\theta_k \in (0, \pi)$. The ordering of the abscissas gives $\theta_1 > \cdots > \theta_N$, while the inner-product conditions give $\cos(\theta_k - \theta_{k+1}) = 1 - q - p(\Upsilon)\cos\theta_{k+1}$ for $1 \le k \le N-1$. At $k = N$, the same condition gives $\sin\theta_N = \sqrt{1-q}$, and $a_N < a_{N+1}$ selects $\cos\theta_N = -\sqrt{q}$. Therefore admissible constructions satisfy (44). Define the shooting residual by $\Xi_N(\Upsilon) \triangleq \theta_1(\Upsilon) - \vartheta(\Upsilon)$ for $\Upsilon > 1$. Lemma 16 below shows that the candidate closes exactly when $\Xi_N(\Upsilon) = 0$.

The numerical value of the root Υ_N can be computed by bisection. Initialize $\Upsilon_l = 1$ as an unevaluated sentinel, since Ξ_N is defined only on $(1, \infty)$, and set $\Upsilon_r = 2$. Double Υ_r until $\Xi_N(\Upsilon_r) < 0$. Then bisect the interval $[\Upsilon_l, \Upsilon_r]$ by computing the midpoint $\Upsilon_m = (\Upsilon_l + \Upsilon_r)/2$, replace Υ_l by Υ_m when $\Xi_N(\Upsilon_m) > 0$, and replace Υ_r by Υ_m when $\Xi_N(\Upsilon_m) < 0$. For a prescribed tolerance $\epsilon > 0$, return Υ_m if $\Xi_N(\Upsilon_m) = 0$ or $|\Upsilon_r - \Upsilon_l| < \epsilon$. Corollary 3 proves that the doubling terminates and the bracket always contains Υ_N. If the bisection is continued indefinitely, its midpoints converge to Υ_N; if it stops with bracket width below ϵ, the returned value is within ϵ of Υ_N.

Lemma 13 (One-step map)**.** *Fix $q \in (0,1)$ and $p \in (0, q)$, and define $F_p(\theta) \triangleq \theta + \arccos(1 - q - p\cos\theta)$. Then:*

1. *F_p is well defined on $\mathbb{R}$: the argument of the* arccos *lies in $(1-2q, 1) \subset (-1, 1)$, so $F_p(\theta) - \theta \in (0, \pi)$;*
2. *F_p is strictly increasing in θ;*
3. *for fixed θ with $\cos\theta < 0$, the map $p \mapsto F_p(\theta)$ is strictly decreasing.*

Proof. Write $u \triangleq 1 - q - p\cos\theta$. Claim 1 follows from $|p\cos\theta| \le p < q$ leading to $-1 < 1 - 2q < 1 - q - p \le u \le 1 - q + p < 1$.

For claim 2, differentiating in θ gives $F_p'(\theta) = 1 - (p\sin\theta/\sqrt{1-u^2})$, so it suffices to prove $p^2\sin^2\theta < 1 - u^2$ expanding $u = 1 - q - p\cos\theta$, tedious but straightforward algebra gives $p^2\sin^2\theta + u^2 \le (p + 1 - q)^2 < 1$, the last inequality because $0 < p + 1 - q < 1$ by $p < q$. This proves claim 2. For claim 3, differentiating in p gives for $\cos\theta < 0$, $\partial F_p(\theta)/\partial p = \cos\theta/\sqrt{1-u^2} < 0$. □

By Lemma 13, the orbit in (44) which is written as $\theta_k(\Upsilon) = F_{p(\Upsilon)}(\theta_{k+1}(\Upsilon))$ is defined for every $\Upsilon > 1$, each $\theta_k(\Upsilon)$ is a continuous function of Υ, and $\theta_1(\Upsilon) > \theta_2(\Upsilon) > \cdots > \theta_N(\Upsilon) = \arccos\left(-\sqrt{q}\right) > \pi/2$.

Lemma 14 (Orbit comparison)**.** *Let $N \ge 2$ and $1 < \Upsilon < \Upsilon'$. If $\theta_1(\Upsilon) \le \pi$, then $\theta_k(\Upsilon') < \theta_k(\Upsilon)$ for $1 \le k \le N-1$.*

Proof. The hypothesis gives $\theta_k(\Upsilon) \in (\pi/2, \pi]$, hence $\cos\theta_k(\Upsilon) < 0$, for every $1 \le k \le N$. We induct downward on k. For the base case $k = N-1$, since $\cos\theta_N = -\sqrt{q} < 0$ and $p(\Upsilon') > p(\Upsilon)$, claim 3 of Lemma 13 gives $\theta_{N-1}(\Upsilon') = F_{p(\Upsilon')}(\theta_N) < F_{p(\Upsilon)}(\theta_N) = \theta_{N-1}(\Upsilon)$. For the step from $k+1$ to $k \le N-2$, claims 2 and 3 of Lemma 13 give $\theta_k(\Upsilon') = F_{p(\Upsilon')}(\theta_{k+1}(\Upsilon')) < F_{p(\Upsilon')}(\theta_{k+1}(\Upsilon)) <$

$F_{p(\Upsilon)}(\theta_{k+1}(\Upsilon)) = \theta_k(\Upsilon)$, where the first inequality uses the induction hypothesis and the second uses $\cos\theta_{k+1}(\Upsilon) < 0$. $\square$

Lemma 15 (Target angle)**.** *The target angle* (43) *is well defined, continuous, and strictly increasing on* $(1,\infty)$*, with* $\vartheta(\Upsilon) \in (0,\pi)$ *for every* $\Upsilon > 1$*,* $\vartheta(\Upsilon) \to \arccos\left(\sqrt{q}\right)$ *as* $\Upsilon \downarrow 1$*, and* $\vartheta(\Upsilon) \to \pi$ *as* $\Upsilon \to \infty$*.*

Proof. Write $c(\Upsilon)$ for the fraction in (43), and note that its denominator is $1 + (1-q)R^2 = q + (1-q)\Upsilon^2$. Then $c(\Upsilon) < 1$, since the numerator is at most $\sqrt{q} < 1 < q + (1-q)\Upsilon^2$. Moreover, using $\Upsilon(\Upsilon - R) = \Upsilon/(\Upsilon + R)$ (from $\Upsilon^2 - R^2 = 1$), tedious but straightforward algebra writes $c(\Upsilon)+1$ as a ratio of positive quantities, so $c(\Upsilon) > -1$: hence $\vartheta(\Upsilon) = \arccos c(\Upsilon)$ is well defined in $(0,\pi)$, and it is continuous. Differentiating the quotient and simplifying (tedious but straightforward algebra) gives, for $\Upsilon > 1$,

$$c'(\Upsilon) = -\frac{(1-q)\left((1+q)\Upsilon^2 - q + 2\sqrt{q}\,\Upsilon R\right)}{R\left(q + (1-q)\Upsilon^2\right)^2},$$

whose bracketed factor equals $(1+q)\left(\Upsilon^2 - 1\right) + 2\sqrt{q}\,\Upsilon R + 1 > 1$. Every remaining factor is positive, so $c'(\Upsilon) < 0$ and c is strictly decreasing on $(1,\infty)$, and composing with the strictly decreasing arccos shows that ϑ is strictly increasing. The limits follow by substitution: as $\Upsilon \downarrow 1$, $R \to 0$ and $c \to \sqrt{q}$; as $\Upsilon \to \infty$, dividing the numerator and denominator of c by $(1-q)\Upsilon^2$ gives $c \to -1$. $\square$

For $\Upsilon > 1$, call a collection of points an *admissible configuration at* Υ if it satisfies the ordering, positivity, and conditions of Lemma 4 with Υ_N replaced by Υ (and hence with $C = C(\Upsilon)$ and $R = R(\Upsilon)$).

Lemma 16 (Shooting equivalence)**.** *An admissible configuration at* $\Upsilon > 1$ *exists if and only if* $\Xi_N(\Upsilon) = 0$*. When it exists, it is unique.*

Proof. Suppose an admissible configuration exists at Υ. By the angle parametrization, its angles satisfy $\theta_N = \arccos\left(-\sqrt{q}\right)$ and the backward recurrence (44), so they coincide with the orbit $\theta_k(\Upsilon)$; in particular $\theta_1(\Upsilon) < \pi$. Moreover, the shooting equation (42) together with $a_1 > a_0$ places P_1 at the farther intersection, so $\cos\theta_1(\Upsilon) = \cos\vartheta(\Upsilon)$; both angles lie in $(0,\pi)$, hence $\theta_1(\Upsilon) = \vartheta(\Upsilon)$. Every coordinate is forced (P_0 and P_{N+1} by their formulas, P_k by $\theta_k(\Upsilon)$), which is the uniqueness claim.

Conversely, suppose $\theta_1(\Upsilon) = \vartheta(\Upsilon)$. Define $P_k \triangleq C + R(\cos\theta_k(\Upsilon), \sin\theta_k(\Upsilon))$ for $1 \le k \le N$ and the endpoints by their formulas. Then all $\theta_k(\Upsilon)$ lie in $(0,\pi)$ because $0 < \theta_N \le \theta_k(\Upsilon) \le \theta_1(\Upsilon) = \vartheta(\Upsilon) < \pi$, so $b_k = R\sin\theta_k(\Upsilon) > 0$; the angles decrease strictly in k, so $a_1 < \cdots < a_N$; and $a_N = \Upsilon - \sqrt{q}\,R < \Upsilon = a_{N+1}$. The norm condition holds by construction. For the inner-product condition at $1 \le k \le N-1$, taking cosines in (44) gives $\cos(\theta_k - \theta_{k+1}) = 1 - q - p\cos\theta_{k+1} = 1 - q\,a_{k+1}/\Upsilon$, and $(P_k - C)\cdot(P_{k+1} - C) = R^2\cos(\theta_k - \theta_{k+1})$ since both points lie on the circle; at $k = N$ the condition holds because $\sin\theta_N = \sqrt{1-q}$. At $k = 0$: the point $\left(a_+, \sqrt{1-q}\,R\,a_+\right)$ lies on the circle with positive ordinate, and P_1 is the point of the circle with the same abscissa $a_1 = \Upsilon + R\cos\vartheta(\Upsilon) = a_+$ and positive ordinate, so the two coincide; hence $\sin\theta_1(\Upsilon) = \sqrt{1-q}\,a_1$, which is (42) and, reversing the algebra that produced it, the inner-product condition at $k = 0$. Finally $a_0 = 1/\Upsilon < a_+ = a_1$. $\square$

Proof of Lemma 4. Existence. We first show $\Xi_N(\Upsilon) < 0$ for all large Υ. Consider the boundary parameter $p = q$: the map $F_q(\theta) = \theta + \arccos(1 - q(1 + \cos\theta))$ is still well defined, since $1 - q(1 + \cos\theta) \in [1 - 2q, 1]$. For $\theta \in (0, \pi)$,

$$\begin{aligned} F_q(\theta) < \pi &\Leftrightarrow \arccos(1 - q(1 + \cos\theta)) < \pi - \theta \\ &\Leftrightarrow 1 - q(1 + \cos\theta) > -\cos\theta \\ &\Leftrightarrow (1 - q)(1 + \cos\theta) > 0, \end{aligned}$$

which holds; so the orbit $\overline{\theta}_N \triangleq \theta_N$, $\overline{\theta}_k \triangleq F_q(\overline{\theta}_{k+1})$ satisfies $\overline{\theta}_1 < \pi$ by induction. Each backward step $(p, \theta) \mapsto F_p(\theta)$ is jointly continuous for $p \in [0, q]$ because the arccos argument stays in $[1 - 2q, 1]$; at the endpoint $p = 0$ it reads $F_0(\theta) = \theta + \arccos(1 - q)$. Hence $\theta_1(\Upsilon)$, which depends on Υ only through $p(\Upsilon)$, converges to $\overline{\theta}_1$ as $p(\Upsilon) \to q$, that is, as $\Upsilon \to \infty$. Combining with Lemma 15, we have $\lim_{\Upsilon\to\infty} \Xi_N(\Upsilon) = \overline{\theta}_1 - \pi < 0$, so $\Xi_N(b) < 0$ for some $b > 1$.

Next we produce a point where $\Xi_N > 0$. As $\Upsilon \downarrow 1$ we have $p(\Upsilon) \to 0$, so by the same joint continuity, now at the endpoint $p = 0$, $\theta_1(\Upsilon) \to \theta_N + (N - 1)\arccos(1 - q)$, while $\vartheta(\Upsilon) \to \arccos(\sqrt{q})$ by Lemma 15. Therefore

$$\lim_{\Upsilon\downarrow 1} \Xi_N(\Upsilon) = \theta_N + (N - 1)\arccos(1 - q) - \arccos(\sqrt{q}) \geq \pi - 2\arccos(\sqrt{q}) = \arccos(1 - 2q) > 0,$$

where the middle inequality drops the nonnegative term $(N - 1)\arccos(1 - q)$ and uses $\theta_N = \arccos(-\sqrt{q}) = \pi - \arccos(\sqrt{q})$, and the last equality holds because $\cos(\pi - 2\arccos\sqrt{q}) = -(2q - 1) = 1 - 2q$ and $\pi - 2\arccos\sqrt{q} \in (0, \pi)$. By continuity, $\Xi_N(a) > 0$ for some $a > 1$. Hence $\Xi_N(a) > 0 > \Xi_N(b)$ with $a < b$ (enlarging b if necessary), so the intermediate value theorem yields $\Upsilon_N \in (a, b)$ with $\Xi_N(\Upsilon_N) = 0$. By Lemma 16, an admissible configuration exists at Υ_N.

Uniqueness. Suppose $\Upsilon < \Upsilon'$ both admit admissible points, so that $\theta_1(\Upsilon) = \vartheta(\Upsilon) < \pi$ and $\theta_1(\Upsilon') = \vartheta(\Upsilon')$ by Lemma 16. For $N = 1$ the orbit is the constant $\arccos(-\sqrt{q})$, and the strict monotonicity of ϑ makes the two equalities incompatible. For $N \geq 2$, Lemma 14 applies at Υ (its hypothesis $\theta_1(\Upsilon) \leq \pi$ holds) and gives, with Lemma 15, $\theta_1(\Upsilon') < \theta_1(\Upsilon) = \vartheta(\Upsilon) < \vartheta(\Upsilon') = \theta_1(\Upsilon')$, a contradiction. Hence the admissible Υ is unique, and Lemma 16 shows that the points are uniquely determined by it. $\square$

The same objects also give a sign characterization of the residual that makes the numerical evaluation of Υ_N straightforward.

Corollary 3 (Bisection correctness)**.** *Let Υ_N be as in Lemma 4. Then the residual Ξ_N defined above satisfies $\Xi_N(\Upsilon) > 0$ for $1 < \Upsilon < \Upsilon_N$ and $\Xi_N(\Upsilon) < 0$ for $\Upsilon > \Upsilon_N$. Consequently both update rules of the bisection described above are correct, the doubling of the right endpoint terminates and produces a bracket containing Υ_N, the invariant $\Upsilon_l < \Upsilon_N < \Upsilon_r$ is maintained, and, if the bisection is continued indefinitely, the midpoints converge to Υ_N. For any $\epsilon > 0$, if it stops with bracket width below ϵ, the returned value is within ϵ of Υ_N.*

Proof. By Lemma 16, $\theta_1(\Upsilon_N) = \vartheta(\Upsilon_N) < \pi$, so $\Xi_N(\Upsilon_N) = 0$. For $\Upsilon > \Upsilon_N$, Lemma 14 applies at Υ_N (its hypothesis $\theta_1(\Upsilon_N) \leq \pi$ holds) and gives, with the strictly increasing target angle of Lemma 15, $\Xi_N(\Upsilon) = \theta_1(\Upsilon) - \vartheta(\Upsilon) < \theta_1(\Upsilon_N) - \vartheta(\Upsilon_N) = 0$. For $1 < \Upsilon < \Upsilon_N$, either $\theta_1(\Upsilon) \geq \pi > \vartheta(\Upsilon)$ and $\Xi_N(\Upsilon) > 0$ directly, or $\theta_1(\Upsilon) < \pi$ and Lemma 14, now applied at Υ, gives in the same way $\Xi_N(\Upsilon) > \Xi_N(\Upsilon_N) = 0$. (For $N = 1$ the orbit is the constant $\arccos\left(-\sqrt{q}\right)$ and both inequalities hold through the target angle alone.) For the bisection, the doubling terminates, since $\Xi_N < 0$ once Υ_r exceeds Υ_N, producing a bracket $\Upsilon_l = 1 < \Upsilon_N < \Upsilon_r$; by the sign characterization each update rule fires exactly on its side of Υ_N, and the bisection bookkeeping is exactly parallel to the argument of Appendix A.1, so the invariant $\Upsilon_l < \Upsilon_N < \Upsilon_r$ is preserved. If the bisection is continued indefinitely, its bracket width tends to zero, so the midpoints converge to Υ_N. At finite termination with $|\Upsilon_r - \Upsilon_l| < \epsilon$, the bracket invariant puts the returned value within ϵ of Υ_N. $\square$

Lemma 17 (Symmetry of the construction)**.** *For* $0 \le k \le N+1$*,*

$$\begin{aligned} a_k a_{N+1-k} &= 1, \\ a_k b_{N+1-k} &= b_k. \end{aligned} \tag{45}$$

Proof. The circle equation is equivalent to $a^2 + b^2 - 2\Upsilon_N a + 1 = 0$, and the map, defined for $a > 0$, $T(a,b) = (1/a, b/a)$, which satisfies $T \circ T = \mathrm{id}$, maps this circle to itself, with $T(P_0) = P_{N+1}$. Moreover, if $P = (a,b)$ and $P' = (a',b')$ satisfy

$$(P - C)\cdot(P' - C) = R_N^2\left(1 - \frac{qa'}{\Upsilon_N}\right),$$

then direct expansion gives

$$(T(P') - C)\cdot(T(P) - C) = \frac{(P - C)\cdot(P' - C) + R_N^2(aa' - 1)}{aa'} = R_N^2\left(1 - \frac{q}{\Upsilon_N a}\right).$$

This is the recurrence for the reversed pair, since the first coordinate of $T(P)$ is $1/a$. Hence $\widetilde{P}_k \triangleq T(P_{N+1-k})$ satisfies the same endpoint conditions and recurrence as P_k. Its first coordinates are strictly increasing and its second coordinates are positive, so uniqueness in Lemma 4 gives $T(P_k) = P_{N+1-k}$. Comparing coordinates proves (45). $\square$

The construction also gives the explicit relaxation used in Theorem 3.

Lemma 18. *Let* Υ_N *be as in Lemma 4. Then, for every* $N \ge 1$*, we have* $1/\Upsilon_N^2 < 4(1-\sqrt{q})^{2N}$*.*

Proof. For the construction angles θ_k, define $t_k \triangleq \cot(\theta_k/2)$ for $1 \le k \le N$. The half-angle formula gives

$$t_k = \frac{\sin\theta_k}{1 - \cos\theta_k} = \frac{b_k}{\Upsilon_N + R_N - a_k}.$$

By the symmetry of Lemma 17, $a_1 a_N = 1$ and $b_1 = a_1 b_N$; substituting these together with $a_N = \Upsilon_N - \sqrt{q}\, R_N$ and $b_N = \sqrt{1-q}\, R_N$ into the half-angle formula, tedious but straightforward algebra gives $t_1 = \left(\sqrt{1-q}\right) / \left((1-\sqrt{q})(\Upsilon_N + R_N)\right)$ and $t_N = \left(\sqrt{1-q}\right) / \left(1 + \sqrt{q}\right)$. For $1 \le k \le N-1$, the inner-product condition, $\cos\theta_{k+1} < 0$, and $R_N < \Upsilon_N$ give

$$\begin{aligned} \cos(\theta_k - \theta_{k+1}) &= 1 - q\left(1 + \frac{R_N}{\Upsilon_N}\cos\theta_{k+1}\right) \\ &< 1 - q(1 + \cos\theta_{k+1}) = 1 - 2q\cos^2\left(\frac{\theta_{k+1}}{2}\right), \end{aligned}$$

and hence, taking half-angle sines, expanding $\sin((\theta_k-\theta_{k+1})/2)$, and dividing by $\sin(\theta_k/2)\sin(\theta_{k+1}/2)$ (tedious but straightforward algebra), $t_k < (1-\sqrt{q})t_{k+1}$ for $1 \le k \le N-1$, which gives $t_1 \le (1-\sqrt{q})^{N-1} t_N$.

Combining the contraction with these boundary values,

$$\frac{1}{2\Upsilon_N} < \frac{1}{\Upsilon_N + R_N} = t_1 t_N \le t_N^2 (1-\sqrt{q})^{N-1} = \frac{(1-\sqrt{q})^N}{1+\sqrt{q}}.$$

It follows that $1/\Upsilon_N^2 < \big(2/(1+\sqrt{q})\big)^2 (1-\sqrt{q})^{2N} < 4(1-\sqrt{q})^{2N}$, which completes the proof. $\square$

B.2 Norm identities for the ITEM-f Lyapunov sequence

This appendix proves Lemma 5 and shows how the update equation of (ITEM-f) enters the analysis.

Proof of Lemma 5. We use the coordinate relations

$$\begin{aligned} c_k a_k + s_k b_k &= (1-q) a_{k+1}, \quad 1 \le k \le N, \\ a_k &= (c_k + q) a_{k+1} - s_k b_{k+1}, \quad 1 \le k \le N-1. \end{aligned} \tag{46}$$

For $1 \le k \le N-1$, these are the coordinate relations for the rotation from $P_k - C$ to $P_{k+1} - C$ and its inverse. At $k = N$, the first identity follows from $c_N = 1-q$, $s_N = \sqrt{q(1-q)}$, $a_{N+1} = \Upsilon_N$, and the boundary facts $b_N = \sqrt{1-q}\, R_N$ and $\Upsilon_N - a_N = \sqrt{q}\, R_N$ of the construction (Appendix B.1).

First identity. Let $g_0 \triangleq \nabla f(x_0)$. Since $x_{-1}^+ = x_0$, the first update (ITEM-f) and the definition of ϕ_N with $a_1 a_N = 1$, $a_0 = 1/\Upsilon_N$ give

$$x_1 - x_\star = x_0^+ - x_\star - \frac{\phi_{N-1}}{\phi_N}\left(1 + \frac{a_0}{(1-q)a_1}\right)\frac{g_0}{L} = x_0^+ - x_\star - \frac{\phi_{N-1}}{\Upsilon_N\sqrt{1-q}}\frac{g_0}{L}.$$

The definition of s_1 gives

$$\frac{(1-q)a_2}{s_1}\frac{\phi_{N-1}}{\Upsilon_N\sqrt{1-q}} = \sqrt{\frac{1-q}{q}}.$$

Substituting this update into the definition (20) of W_1 and using the first coordinate relation (46), we have

$$W_1 = \begin{bmatrix} a_1(x_0^+ - x_\star) \\ b_1(x_0^+ - x_\star) - \sqrt{\dfrac{1-q}{q}}\dfrac{g_0}{L} \end{bmatrix}.$$

The boundary relations in the construction give $b_1 = \sqrt{1-q}\, R_N a_1$ (the shooting relation (42)) and $\Upsilon_N a_1 - 1 = \sqrt{q}\, R_N a_1$ (equivalent to $a_1 a_N = 1$ from Lemma 17). Together with the circle equation for P_1, these imply

$$\begin{aligned} b_1 &= \sqrt{\frac{1-q}{q}}(\Upsilon_N a_1 - 1), \\ a_1^2 + b_1^2 &= 2\Upsilon_N a_1 - 1. \end{aligned}$$

Expanding the squared norm and using $\mu = qL$, $\Upsilon_N a_0 = 1$, and $(1-q)g_0 = \widetilde{g}_0 + \mu(x_0^+ - x_\star)$, we obtain

$$\begin{aligned}
\|W_1\|^2 &= (2\Upsilon_N a_1 - 1)\left\|x_0^+ - x_\star\right\|^2 + \frac{1-q}{qL^2}\|g_0\|^2 - \frac{2(\Upsilon_N a_1 - 1)}{\mu}\left\langle x_0^+ - x_\star, (1-q)g_0\right\rangle \\
&= \left\|x_0^+ - x_\star\right\|^2 + \frac{1-q}{qL^2}\|g_0\|^2 - \frac{2\Upsilon_N}{\mu}(a_1 - a_0)\left\langle x_0^+ - x_\star, \widetilde{g}_0\right\rangle.
\end{aligned}$$

Finally, direct expansion of $x_0^+ - x_\star = x_0 - x_\star - L^{-1}g_0$ and $\widetilde{g}_0 = g_0 - qL(x_0 - x_\star)$ gives

$$\left\|x_0^+ - x_\star\right\|^2 + \frac{1-q}{qL^2}\|g_0\|^2 = (1-q)\|x_0 - x_\star\|^2 + \frac{1}{qL^2}\|\widetilde{g}_0\|^2.$$

Since $1/(qL^2) = \Upsilon_N a_0/(\mu L)$, substitution proves the first identity.

Second identity. For $1 \le k \le N-1$, the definition $s_k = \sqrt{q}\, a_{k+1}\phi_{N-k}/\Upsilon_N$ gives $\phi_{N-k-1}/\phi_{N-k} = (a_{k+1}s_{k+1})/(a_{k+2}s_k)$. Thus the update of (ITEM-f) becomes

$$x_{k+1} = x_k^+ + \frac{a_k s_{k+1}}{(1-q)a_{k+2}s_k}(x_k^+ - x_{k-1}^+) + \frac{a_{k+1}s_{k+1}}{a_{k+2}s_k}(x_k^+ - x_k).$$

Substituting this into the definition (20) of W_{k+1} and using the first coordinate relation (46) gives

$$W_{k+1} = \begin{bmatrix} a_{k+1}(x_k^+ - x_\star) \\ b_{k+1}(x_k^+ - x_\star) + \dfrac{a_k}{s_k}(x_k^+ - x_{k-1}^+) + \dfrac{(1-q)a_{k+1}}{s_k}(x_k^+ - x_k) \end{bmatrix}.$$

Subtracting $\mathcal{O}_k W_k$ of (21), the first component of $W_{k+1} - \mathcal{O}_k W_k$ is:

$$a_{k+1}(x_k^+ - x_\star) - (1-q)a_{k+1}(x_k - x_\star) = -\frac{a_{k+1}}{L}\widetilde{g}_k.$$

The second component is:

$$\begin{aligned}
&b_{k+1}(x_k^+ - x_\star) + \frac{a_k}{s_k}(x_k^+ - x_{k-1}^+) + \frac{(1-q)a_{k+1}}{s_k}(x_k^+ - x_k) \\
&\quad - \frac{(1-q)c_k a_{k+1}(x_k - x_\star) - a_k(x_{k-1}^+ - x_\star)}{s_k} \\
&= \left(b_{k+1} + \frac{a_k}{s_k} + \frac{(1-q)a_{k+1}}{s_k}\right)(x_k^+ - x_\star) \\
&\quad - \frac{(1-q)a_{k+1}(1+c_k)}{s_k}(x_k - x_\star) \\
&= \frac{a_{k+1}(1+c_k)}{s_k}\left(x_k^+ - x_\star - (1-q)(x_k - x_\star)\right) \\
&= -\frac{\Upsilon_N s_k}{\mu}\widetilde{g}_k,
\end{aligned}$$

where the second equality uses the second coordinate relation (46) and the last equality uses $a_{k+1}(1+c_k)/s_k = a_{k+1}s_k/(1-c_k) = \Upsilon_N s_k/q$ and $x_k^+ - x_\star = (1-q)(x_k - x_\star) - L^{-1}\widetilde{g}_k$. Since $1 - c_k = qa_{k+1}/\Upsilon_N$, the two components give

$$W_{k+1} - \mathcal{O}_k W_k = -\frac{\Upsilon_N}{\mu}\begin{bmatrix}(1-c_k)\widetilde{g}_k \\ s_k\widetilde{g}_k\end{bmatrix} = -\frac{\Upsilon_N}{\mu}\mathcal{O}_k\begin{bmatrix}(c_k-1)\widetilde{g}_k \\ s_k\widetilde{g}_k\end{bmatrix},$$

where the last equality uses $c_k^2 + s_k^2 = 1$. Rearrangement gives

$$W_{k+1} = \mathcal{O}_k\left(W_k - \frac{\Upsilon_N}{\mu}\begin{bmatrix}(c_k-1)\widetilde{g}_k \\ s_k\widetilde{g}_k\end{bmatrix}\right).$$

Taking norms and using the orthogonality of $\mathcal{O}_k$ proves the second identity.

Third identity. At $k = N$, the definition (20) of W_N and $c_N = 1-q$, $s_N = \sqrt{q(1-q)}$, and $a_{N+1} = \Upsilon_N$ give

$$W_N = \begin{bmatrix} a_N(x_{N-1}^+ - x_\star) \\ \sqrt{\frac{1-q}{q}}\big(\Upsilon_N(x_N - x_\star) - a_N(x_{N-1}^+ - x_\star)\big)\end{bmatrix}.$$

Expanding and regrouping its squared norm yields

$$\begin{aligned}\|W_N\|^2 &= a_N^2\left\|x_{N-1}^+ - x_\star\right\|^2 + \frac{1-q}{q}\left\|\Upsilon_N(x_N - x_\star) - a_N(x_{N-1}^+ - x_\star)\right\|^2 \\ &= (1-q)\Upsilon_N^2\left\|x_N - x_\star\right\|^2 + \frac{1}{q}\left\|(1-q)\Upsilon_N(x_N - x_\star) - a_N(x_{N-1}^+ - x_\star)\right\|^2,\end{aligned}$$

which is the third identity. □

B.3 Computer-assisted algorithm design by PEP

PEP formulation of the problem. This subsection presents the Stage-1 derivation that bnb-pep-skill wrote for the ITEM-f design problem, again lightly edited for presentation. The derivation follows the PEP methodology of Section 3.1 [19, 65, 14], developed in detail for lemniscate acceleration in Appendix A.3, whose notation (Section 4) it reuses. Below we spell out the case-specific ingredients, namely the function class, the function-value performance measure with its normalized initial condition, and the reduction to a transformed function, and refer to Appendix A.3 for the steps that carry over unchanged.

The design problem is instantiated with the function class $\mathcal{F} = \mathcal{F}_{\mu,L}$ and the method class $\mathcal{M}_N$ of N-step FSFOMs. Applying a method in $\mathcal{M}_N$ to $f \in \mathcal{F}_{\mu,L}$ from a starting point x_0, with the stepsizes normalized by L as in Appendix A.3, produces the iterates

$$x_i = x_{i-1} - \frac{1}{L}\sum_{j=0}^{i-1} h_{i,j}\nabla f(x_j), \quad 1 \le i \le N, \tag{47}$$

with stepsizes $\{h_{i,j}\}_{0\le j<i\le N}$ fixed in advance, independently of f. We rate a method by its worst-case function-value suboptimality after N steps, relative to the initial suboptimality. Since $\mathcal{F}_{\mu,L}$ is invariant under translations of the variable and shifts of the function value, we set $x_\star = 0$ and

$f_\star = 0$. Optimality conditions give $\nabla f(x_\star) = 0$, and we normalize the initial gap by $f(x_0) - f_\star \le 1$. The worst-case performance of $M \in \mathcal{M}_N$ is

$$\mathcal{R}(M) = \left(\begin{array}{ll} \text{maximize} & f(x_N) - f_\star \\ \text{subject to} & f \in \mathcal{F}_{\mu,L}, \\ & \{x_i\}_{1\le i\le N} \text{ generated from } x_0 \text{ by } M, \\ & f(x_0) - f_\star \le 1, \\ & x_\star = 0,\ f_\star = 0,\ \nabla f(x_\star) = 0 \end{array} \right), \tag{$\mathcal{O}_f^{\mathrm{in}}$}$$

where the decision variables are the function f and the iterates $x_0, \dots, x_N$. The optimal FSFOM $M_N^\star \in \mathcal{M}_N$ for this setup solves the outer design problem $\mathcal{R}^\star(\mathcal{M}_N) = \min_{M\in\mathcal{M}_N} \mathcal{R}(M)$ of (1).

Following [62], the derivation passes to a transformed function. By scale invariance [65, §3.5] fix $L = 1$, so $\mu = q$. For $f \in \mathcal{F}_{q,1}$ with $x_\star = 0$, $f_\star = 0$, the shifted function $\tilde{f} \triangleq f - (q/2)\left\|\cdot\right\|^2$ belongs to $\mathcal{F}_{0,1-q}$, with gradients $\tilde{g}_i = \nabla f(x_i) - q\, x_i$ and $\tilde{g}_\star = 0$; the original suboptimality is recovered from the transformed values $\tilde{f}_i = \tilde{f}(x_i)$ through

$$f(x_i) - f_\star = \tilde{f}_i + \tfrac{q}{2}\left\|x_i\right\|^2. \tag{48}$$

Re-expressing (47) through $\tilde{g}_i$, the iterates take the form

$$x_i = \Big(1 - q\sum_{j=0}^{i-1} \alpha_{i,j}\Big) x_0 - \sum_{j=0}^{i-1} \alpha_{i,j}\, \tilde{g}_j, \quad 1 \le i \le N, \tag{49}$$

where the transformed stepsizes $\{\alpha_{i,j}\}_{0\le j<i\le N}$ are defined from the original stepsizes $\{h_{i,j}\}_{0\le j<i\le N}$ through the triangular (hence invertible) system of equations [62, §3.2] $\alpha_{i,i-1} = h_{i,i-1}$ and $\alpha_{i,j} = h_{i,j} + \alpha_{i-1,j} - q \displaystyle\sum_{k=j+1}^{i-1} h_{i,k}\alpha_{k,j}$, for $0 \le j \le i-2$. The transformed function obeys (6). We impose these smooth-convex interpolation inequalities at the sampled iterates, with $1-q$ in place of L.

Collect the data into $P = [\, x_0 \mid \tilde{g}_0 \mid \cdots \mid \tilde{g}_N \,] \in \mathbb{R}^{d\times(N+2)}$, $G = P^\top P \in \mathbb{S}_+^{N+2}$, and $F = [\, \tilde{f}_0 \mid \cdots \mid \tilde{f}_N \,]$. With the selectors $\mathbf{x}_0 = e_1$, $\mathbf{g}_i = e_{i+2}$ (so $\tilde{g}_i = P\mathbf{g}_i$), $\mathbf{f}_i = e_{i+1}$, $\mathbf{x}_\star = \mathbf{g}_\star = 0$, and $\mathbf{f}_\star = 0$, define the iterate selectors by mirroring the coefficients of (49),

$$\mathbf{x}_i \triangleq \Big(1 - q\sum_{j=0}^{i-1} \alpha_{i,j}\Big) \mathbf{x}_0 - \sum_{j=0}^{i-1} \alpha_{i,j}\, \mathbf{g}_j, \quad 1 \le i \le N, \tag{50}$$

so that $x_i = P\mathbf{x}_i$ by (49), and define $A_{i,j}(\alpha)$, $B_{i,j}(\alpha)$, and $C_{i,j}$ for $i,j \in I_N^\star$ exactly as in (30).

By (50) $\mathbf{x}_i$ is affine in α, so $A_{i,j}$ is affine and $B_{i,j}$ quadratic in α, while $C_{i,j}$ is constant. By (48) the objective is $f(x_N) - f_\star = F\mathbf{f}_N + (q/2)\mathbf{tr}(GB_{N,\star})$ and the initial gap is $f(x_0) - f_\star = F\mathbf{f}_0 + (q/2)\mathbf{tr}(GB_{0,\star})$. Under the large-scale assumption $d \ge N+2$ (Assumption 1) the rank constraint on G is vacuous, and ($\mathcal{O}_f^{\mathrm{in}}$) relaxes to the convex SDP

$$\mathcal{R}(M) = \left(\begin{array}{l} \text{maximize} \quad F\mathbf{f}_N + \frac{q}{2}\mathbf{tr} GB_{N,\star} \\ \text{subject to} \\ F(\mathbf{f}_j - \mathbf{f}_i) + \mathbf{tr} G\big[A_{i,j}(\alpha) + \frac{1}{2(1-q)}C_{i,j}\big] \le 0, \quad i,j \in I_N^\star : i \ne j, \quad \triangleright \text{dual var. } \lambda_{i,j} \ge 0 \\ F\mathbf{f}_0 + \frac{q}{2}\mathbf{tr} GB_{0,\star} \le 1, \quad \triangleright \text{dual var. } \nu \ge 0 \\ G \succeq 0, \quad \triangleright \text{dual var. } Z \succeq 0 \end{array} \right),$$

with decision variables $F \in \mathbb{R}^{1\times(N+1)}$ and $G \in \mathbb{S}^{N+2}$, and the indicated dual variables. Taking the dual gives

$$\overline{\mathcal{R}}(M) = \left(\begin{array}{l} \text{minimize} \quad \nu \\ \text{subject to} \\ \mathbf{f}_N - \nu\, \mathbf{f}_0 + \sum_{i,j\in I_N^\star : i\neq j} \lambda_{i,j}\,(\mathbf{f}_i - \mathbf{f}_j) = 0, \\ \frac{q}{2}\big(\nu B_{0,\star} - B_{N,\star}(\alpha)\big) + \sum_{i,j\in I_N^\star : i\neq j} \lambda_{i,j}\big[A_{i,j}(\alpha) + \frac{1}{2(1-q)} C_{i,j}\big] = Z, \\ Z \succeq 0, \quad \nu \geq 0, \quad \lambda_{i,j} \geq 0 \;\; (i,j \in I_N^\star : i \neq j) \end{array} \right), \tag{51}$$

with decision variables $\nu \in \mathbb{R}$, $\lambda = \{\lambda_{i,j}\}$, and $Z \in \mathbb{S}^{N+2}$. By weak duality, we have $\mathcal{R}(M) \leq \overline{\mathcal{R}}(M)$.

The derivation then recasts the outer problem, in the design variable α, as a QCQP in exact parallel with (33): it substitutes (51) for the inner problem, adjoins $\{\alpha_{i,j}\}_{0\leq j<i\leq N}$ to the decision variables ν, λ, and Z, and replaces the constraint $Z \succeq 0$ with the Cholesky factorization $Z = VV^\top$, where V is lower triangular with nonnegative diagonal again [26, Corollary 7.2.9]. Every constraint of the resulting problem is at most quadratic in its decision variables; upon user approval of the derivation, bnb-pep-skill solves the problem numerically. The reformulation presumes strong duality between the inner SDP and its dual (51), the analogue of Assumption 2; weak duality alone already makes its optimal value an upper bound on $\mathcal{R}^\star(\mathcal{M}_N)$.

In Stage 2 of the pipeline, the frontier-llm-consult skill fitted symbolic expressions to the numerical solutions of this QCQP, yielding analytic formulas for the transformed stepsizes and an analytic feasible point of the inner dual (51), expressed in terms of the geometric construction of Lemma 4 (at $L = 1$, hence $\mu = q$); dual feasibility alone already certifies the contraction bound. The consultation then recast the method from the transformed-stepsize representation into the momentum form (ITEM-f) in which Section 4.2 presents it. From the multipliers of this feasible point, continued consultation produced the Lyapunov proof of Theorem 3 presented in Section 4.2.1, stated on the momentum form, and Section 4.2.4 describes the Lean coverage of the presented results.