PEPPY: AN AI-ASSISTED WORKFLOW FOR TIGHT CONVERGENCE ANALYSIS OF OPTIMIZATION ALGORITHMS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper presents Peppy, an AI-assisted workflow for discovering tight, analytic convergence proofs for first-order optimization algorithms. Generic approaches to using LLMs to conduct mathematical research target an unspecified, broad spectrum of problems and sometimes use the Lean 4 proof assistant for formalization. On the other hand, Peppy leverages domain-specific knowledge more heavily and is thereby capable of constructing the proofs in a more structured manner, which allow a minimal and accessible verification through SymPy. We experimentally demonstrate through examples that Peppy provides a rigorous, practical, and reproducible paradigm for AI-assisted theorem synthesis in optimization. We further highlight its capability of closing several open problems on tight convergence analysis of first-order optimization algorithms, including conjectures for Nesterov's FGM. Overall, Peppy is designed to turn the art of optimization algorithm analysis into a science.

1 INTRODUCTION

Recent advances in large language models (LLMs) have fueled growing interest in AI-assisted research (Lu et al., 2024; Bubeck et al., 2025), including mathematical discoveries (Romera-Paredes et al., 2024; Georgiev et al., 2025). In theorem proving, several systems pursue broad applicability through general-purpose proof assistants (Yang et al., 2023; Thakur et al., 2024; Tsoukalas et al., 2026). However, such approaches are not necessarily tailored to structures and computational tools specific to individual mathematical subfields. This motivates domain-specific workflows that integrate LLMs with a specialized set of tools and knowledge, enabling proof discovery to exploit the structure of the subfield at hand. Such workflows can make the process of discovery more controlled and targeted.

This paper presents a complementary AI-assisted workflow Peppy, focusing on searching for convergence proofs in optimization theory. We newly and extensively combine AI with the mathematical framework developed by Yoon et al. (2026), which detects the Lyapunov function structure numerically identifiable via Performance Estimation Problem (PEP) (Drori & Teboulle, 2014; Taylor et al., 2017c) and streamlines proofs alongside code. This enables Peppy to leverage AI agents to perform concrete, step-by-step guidelines, and produce novel results such as the first analytic proof of the tight convergence rate of Nesterov's FGM.

PEP itself is often viewed as a computer-assisted proof framework, providing numerical evidence and guidance for convergence proofs of first-order optimization methods. It which has been proved useful for both finding tight convergence rates and designing novel optimization algorithms (Kim & Fessler, 2016; Taylor et al., 2018; Taylor & Bach, 2019; Ryu et al., 2020; Kim, 2021; Dragomir et al., 2022; Taylor & Drori, 2023; Das Gupta et al., 2024; Upadhyaya et al., 2025).

However, PEP provides numerical—rather than analytic—certificates for a proof, and traditional research processes utilizing PEP typically involved a largely manual art of converting numerical data into analytic proofs, requiring significant effort from human experts. Packages such as PESTO (Taylor et al., 2017b), PEPit (Goujaud et al., 2024), AutoLyap (Upadhyaya et al., 2026) or PEPFlow

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(Suh et al., 2025) support the computational side of helping users formulate PEP. Applications of PEP to analytic proof synthesis remain scarce. Kim et al. (2026) is among the very few examples.

Peppy guides AI to perform the procedure of Yoon et al. (2026), which simulate the steps that were traditionally performed by humans. Within the workflow, PEPFlow (Suh et al., 2025) serves as the backend for implementing and solving PEPs while exposing the information needed for analytic extensions. The completion of each step of the workflow can be verified with simple numerical or symbolic checks. As a final output, Peppy provides a rigorous *theorem* stating an optimization algorithm's tight convergence rate with its proof and code scripts verifying it, with varying degrees of human guidance.

1.1 Contributions

We present Peppy, an AI-assisted workflow for discovering analytic proofs of tight convergence bounds for first-order optimization algorithms. Using Peppy, we solve several open problems, including tight convergence rates for Nesterov's FGM. Details are provided below.

- Peppy has several advantages over existing approaches in "AI4Math" that typically combine LLMs with formal verification. We list them as follows:
  - Peppy draws on the powerful **domain-specific machinery** of PEP to provide a tailored approach to AI-assisted optimization research. More specifically, AI agents are guided by concrete instructions based on the systematic framework of Yoon et al. (2026) for converting numerical PEP certificates into analytic Lyapunov analyses. Users can exploit the power of this well-established strategy sharpened for optimization analysis while needing only a high-level understanding of how Peppy operates.
  - Peppy provides AI agents with the **domain-specific library** PEPFlow, which abstracts the underlying PEP components behind an accessible interface. Consequently, AI agents can efficiently handle PEP implementation from conceptual principles, based on an organized source of the necessary domain-specific language.
  - Peppy uses **domain-specific proof structures for rapid verification,** which boils down to simple checks of symbolic matrix/vector identities and positivity of algebraically expressed coefficients, accelerating the search for verified proofs.[1]
- **Using Peppy, we close several conjectures.** Specifically, Peppy yields Lyapunov analyses that, to the best of our knowledge, provide the **first analytic proofs of tight convergence bounds for Nesterov's FGM**. These FGM analyses, together with our OGM result, resolve Conjectures 4–5 of Taylor et al. (2017c). Our analyses of FGM and FISTA with rational coefficients also resolve the corresponding conjectures in Table 1 of Taylor et al. (2017a). Table 1 summarizes these results.

Notation and a detailed discussion of prior work are provided in Appendices A and B, respectively. The code and accompanying examples are included in the supplementary material.

Table 1: Conjectures closed by Peppy. See Sections 5.1 and 5.2 for algorithm definitions.

Conjecture	Corresponding theorem
Taylor et al. (2017c), Conjecture 4, FGM, y_N	Theorem 5.1
Taylor et al. (2017c), Conjecture 5, FGM, x_N	Theorem 5.2
Taylor et al. (2017c), Conjecture 4, OGM, y_N	Theorem 5.3
Taylor et al. (2017a), conjecture in Table 1, FGM-rational, y_N	Theorem 5.4
Taylor et al. (2017a), conjecture in Table 1, FISTA-rational, y_N	Theorem 5.5

[1] As the main requirement for verification is establishing symbolic identities, we present them through Jupyter notebooks which contain the outputs based on SymPy, a symbolic calculation engine. Nevertheless, we provide an auxiliary skill \lean-from-ipynb for users who wish to proceed with Lean formalization.

2 PEPPY: DESIGN PRINCIPLE WITH AN EXAMPLE

The goal of Peppy is to produce a convergence theorem with its proof based on Lyapunov analysis. Given a user input of the form described in Section 2.1, a successful run produces a theorem and its proof as in Section 2.5, with varying degrees of human intervention. Figure 1 summarizes the five stages of Peppy. Below, we outline the theoretical design principle of each step of the workflow via a representative example of gradient descent (GD) for smooth convex minimization.

2.1 PEP IMPLEMENTATION OF INPUT ALGORITHM

In the first stage of Peppy, the user feeds in the problem class, algorithm's update rule, performance metric and initial condition into the coding agent. Input format is flexible, as agents can handle various inputs; we provide a standardized example format in Section 3.1. Once the agent understands all this information, it writes `PEPFlow` code for setting up the PEP construction, which can be numerically solved in the next stage.[2]

Setup: f is L-smooth convex function, characterized by

$$I(x,y) := f(y) - f(x) + \langle \nabla f(y), x - y\rangle + \frac{1}{2L}\left\|\nabla f(x) - \nabla f(y)\right\|^2 \le 0, \quad \forall x, y. \tag{1}$$

Performance metric: $f(x_N) - f(x_\star)$
Initial condition: $\|x_0 - x_\star\| \le R$
Algorithm definition:

$$x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k).$$

2.2 SOLVING PEP AND DETERMINING CERTIFICATES

Given the PEP implementation, the second stage actually builds and solves the PEP problem to obtain numerical certificates, which provide guidance toward the analytical convergence proof. The PEP certificates obtained here can be understood as the numerical version of the following generic identity:

$$\begin{aligned}&\texttt{(Performance metric)} - \tau_N \times \texttt{(Initial condition)}\\ &= \texttt{(Weighted sum of } I(\cdot,\cdot)\texttt{'s)} - \texttt{(Sum of squares)}.\end{aligned}$$

Specifically for GD, we can write

$$\begin{aligned}&f(x_N) - f(x_\star) - \frac{L}{4N+2}\|x_0 - x_\star\|^2 \\ &= \sum_{i=1}^{N} \lambda_{i-1,i} I(x_{i-1}, x_i) + \sum_{i=0}^{N} \lambda_{\star,i} I(x_\star, x_i) - \sum_{i=0}^{N} d_i \left\|s_i\right\|^2\end{aligned} \tag{2}$$

If the above identity is analytically determined, with forms of PEP certificates $\lambda_{i,i+1}, \lambda_{\star,i}, d_i \ge 0$ and s_i clarified, then from (1) we conclude that $f(x_N) - f(x_\star) \le \frac{L}{4N+2}\|x_0 - x_\star\|^2$. However, the PEP certificates are only numerical and searching for their analytical form traditionally required

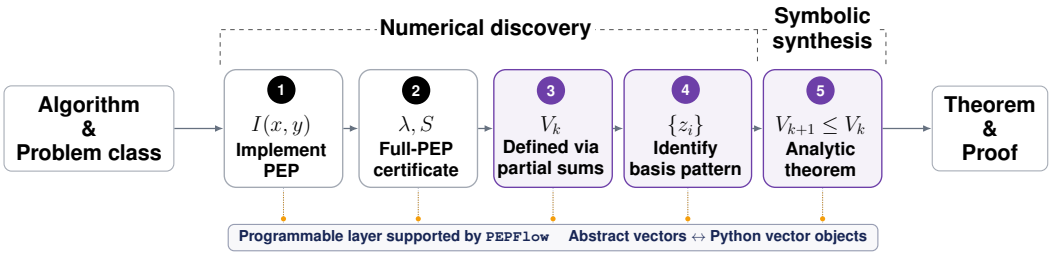


Figure 1: A schematic description of Peppy's 5-stage workflow.

[2]The characterization of the function class by inequality (1) is due to Taylor et al. (2017c).

human effort (Taylor et al., 2017c; Kim & Fessler, 2016). In Peppy, this is often successfully carried out by the coding agent instead.

Because the AI agent works with the identity (2) for some fixed value of N, it has to be converted into a pattern that can be generalized to arbitrary N for an analytic proof. Our strategy is to employ the Lyapunov-style proof approach, outlined in the subsequent sections. Notably, the analytic formulas for certificates $\lambda_{i,i+1}, \lambda_{\star,i}, d_i \geq 0$ and s_i may or may not be easily inferred from the numerical values. In our workflow, this is *not a requirement*—one may proceed to the next stage using only numerical information, which is sufficient for inferring a right form of Lyapunov function to use. Missing closed forms can be determined symbolically in the final stage.

2.3 Defining Lyapunov function

In this stage, we decompose the PEP proof (2) into partial sums V_k for $k = 1, \ldots, N$, as:

$$V_k := \sum_{i=1}^{k} \lambda_{i-1,i} I(x_{i-1}, x_i) + \sum_{i=0}^{k} \lambda_{\star,i} I(x_\star, x_i) - \sum_{i=0}^{k} d_i \|s_i\|^2$$

By definition, due to (1), V_k is a nonincreasing *Lyapunov function*. This construction and the subsequent steps of Peppy follow Yoon et al. (2026), which provided a systematic procedure for utilizing the Lyapunov function construction of the above form and deriving a concise and interpretable analytic proof from it. Notably, Goujaud et al. (2023) presented a similar idea of using a partial sum of nonpositive terms as a Lyapunov function, while Yoon et al. (2026) can be viewed as an extension and completion of the theory surrounding it.

2.4 Finding a summation-free expression for V_k

In this stage, we systematically simplify V_k into a summation-free form. This is done via nontrivial linear algebraic techniques that we have implemented. These core functions are packaged into the new `lyapunov_utils.py` component, built upon the `PEPFlow` library. Section C explains the mathematical construction and its implementation.

Utilizing them, we can extract the basis vectors that can express V_k as a simple (low-rank) quadratic form. This is the most novel component introduced by Peppy, which enables the conversion of (2) into an analytical, N-independent proof. Applying this step to GD, in particular, we obtain:

$$\begin{aligned} V_k = &\frac{k+1}{2N-k}\big(f(x_k) - f(x_\star)\big) - \frac{L}{4N+2}\|x_0 - x_\star\|^2 \\ &- \frac{k+1}{2L(2N-k)}\|\nabla f(x_k)\|^2 + \frac{L(2N-2k-1)}{2(2N-k)^2}\|x_{k+1} - x_\star\|^2 \end{aligned}$$

Note that this stage, on its own, used to be a research-level task requiring human experts, as done in prior work (Park et al., 2023; d'Aspremont et al., 2021; Lee et al., 2021).

2.5 Analytic proof by Lyapunov analysis

In this final step, we verify $V_{k+1} \leq V_k$ through symbolic calculation. In a successful use case, we let Peppy additionally produce LaTeX code for the corresponding proof and the resulting theorem statement, as shown below. Examples of proofs generated by Peppy are provided in Section 4.

Theorem. Let f be convex and L-smooth, and let $x_\star$ minimize f. If $\|x_0 - x_\star\|^2 \leq R^2$, then the gradient descent iterates

$$x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$$

satisfy the following stronger Lyapunov statement. Set $V_0 = 0$ and, for $1 \leq k \leq N-1$,

$$\begin{aligned} V_k = &\frac{k+1}{2N-k}\big(f(x_k) - f(x_\star)\big) - \frac{L}{4N+2}\|x_0 - x_\star\|^2 \\ &- \frac{k+1}{2L(2N-k)}\|\nabla f(x_k)\|^2 + \frac{L(2N-2k-1)}{2(2N-k)^2}\|x_{k+1} - x_\star\|^2, \end{aligned}$$

and $V_N = f(x_N) - f(x_\star) - \frac{L}{4N+2}\|x_0 - x_\star\|^2$. Then $V_0 \geq V_1 \geq \cdots \geq V_N$. Consequently, for every $N \geq 1$,

$$f(x_N) - f(x_\star) \leq \frac{LR^2}{4N+2}.$$

3 PEPPY: AGENTIC IMPLEMENTATION

Peppy is implemented as a sequence of reusable commands, each written in a Markdown file. Here, a *command* is a prompt specification for executing each modular block of the theoretical workflow described in Section 2.

The user feeds Peppy with a prompt including the command name and the required input arguments. Below, we briefly outline the function of each command at a high level, deferring detailed explanation to Section D. See Section 3.1 for a concrete prompt example.

(1) **`/pep-implement`:** Turns the algorithm description into an executable `PEPFlow` setup module.

(2) **`/pep-full-proof`:** Solves PEP with a fixed N and extracts proof certificates, e.g., λ, d, and s in (2), and guesses their closed-form expressions.

(3) **`/lyap-define`:** Constructs a candidate Lyapunov function V_k as partial sums of weighted inequalities and square terms from the PEP proof.

(4) **`/lyap-vectors`:** Identifies the vectors that are used for expressing V_k, and thereby simplifies it into a summation-free expression.

(5) **`/lyap-closed-form`:** Finalizes the analytic coefficients in V_k, symbolically verifies the Lyapunov analysis, and drafts a theorem with a proof outline.

The role of AI assistance in Peppy is two-fold:

- AI generates `PEPFlow`-based codes for structured tasks such as solving PEPs, constructing Lyapunov groupings, and searching for basis vectors.
- AI assists in automating tasks that require mathematical insight, including selecting sparsity patterns, guessing closed-form rates and symbolic formulas, and analytically identifying V_k. Here, to prevent proceeding with hallucinated outputs, Peppy performs specific numerical or symbolic verification steps on the AI agent's outputs.

At the end of each block, the outputs are saved into a `.json` state file, and a human-readable summary is written into a Jupyter notebook, which enables both Peppy and the user to keep track of intermediate results. The user may inspect the progress and provide additional guidance to Peppy when needed.

3.1 HOW TO USE

We describe direct use of the five-stage workflow.

The user specifies an algorithm, a problem class, an initial condition, and a performance metric, then runs the five commands in sequence.

While the input prompts can be flexible, we provide a standardized example format that we found successful, for the case of GD for smooth convex minimization. We start by running the first command, `/pep-implement`, together with the required arguments.

```
/pep-implement

Function: f is convex and L-smooth
Parameters: L, R
Initial condition: ||x_0 - x_star|| <= R, where grad f(x_star) = 0
Performance metric: f(x_N) - f(x_star)
```

```
Algorithm: Gradient descent with fixed step size 1/L

For k >= 0:
x_{k+1} = x_k - (1/L) * grad f(x_k)

Conjectured rate: unknown
```

Alternatively, one can choose to input a screenshot of the algorithm's information, e.g., the box in Section 2.1.

In successful cases, `/pep-implement` produces an output that can be used as an input to the next block, `/pep-full-proof`. This can be triggered by prompting "command name + algorithm name." For example,

```
/pep-full-proof gd
```

will let `Peppy` proceed to the next stage. A simpler alternative for interactive use is:

```
proceed with the next step
```

This is repeated until the final block is reached.

4 Evaluations on Known Results

We provide a simple evaluation of `Peppy`'s ability to recover known convergence rates through analytic Lyapunov proofs for PGM, OGM (Kim & Fessler, 2016), BPPM, FEG (Lee & Kim, 2021), and Dual-FEG (Yoon et al., 2024). PGM is tested with and without a supplied sum-of-squares (SOS) decomposition, yielding six experiment settings with five attempts each. Proof generation and evaluation both use Codex with GPT-6 Astra, medium reasoning.

We design a controlled clean-room experiment protocol, where the AI agent is instructed not to "cheat" by consulting external material or using outputs of other runs, and to strictly adhere to a specified Python environment. The purpose of the experiments is to demonstrate the capability and consistency of `Peppy` in simulating an optimization-specific workflow that has been performed by domain experts. All 30 trials generated verifiable proofs establishing prescribed convergence bounds, streamlined through Jupyter Notebooks.[3] We report the time and generation tokens consumed in Table 2.

Notably, while these runs consider algorithms whose tight rates are already known for the purpose of evaluating the workflow, the *Lyapunov proof* has not appeared in the literature for the case of PGM, which we present below.

Table 2: Evaluation results. Block times (minutes) and generation tokens (millions) are means $\pm$ one sample standard deviation over five attempts.

Condition	Verified	Block 1	Block 2	Block 3	Block 4	Block 5	Tokens (M)
OGM	5/5	7.7 ± 0.8	11.3 ± 1.8	6.8 ± 0.8	7.8 ± 1.6	12.5 ± 1.7	11.5 ± 2.4
BPPM	5/5	7.4 ± 1.1	10.7 ± 2.6	8.1 ± 1.3	6.8 ± 1.2	9.5 ± 2.1	9.9 ± 1.3
FEG	5/5	7.4 ± 0.8	11.1 ± 1.5	7.2 ± 1.0	5.6 ± 0.4	9.2 ± 1.4	10.7 ± 1.1
Dual-FEG	5/5	7.6 ± 0.3	13.1 ± 2.8	7.4 ± 0.7	7.4 ± 1.7	11.2 ± 0.5	12.5 ± 1.7
PGM (no SOS)	5/5	7.1 ± 0.9	10.5 ± 1.5	11.3 ± 4.2	8.0 ± 1.4	23.5 ± 4.7	15.7 ± 2.2
PGM (given SOS)	5/5	6.9 ± 1.0	12.7 ± 0.7	13.6 ± 2.3	10.2 ± 1.1	14.6 ± 3.7	15.4 ± 1.0

[3] A clean-room protocol issue was flagged in one trial each for BPPM, FEG, and PGM (without SOS), during final notebook execution within Block 5 *after* a Jupyter notebook containing a correct proof had been generated. The incidents involved generic Python/Jupyter dependencies outside the designated environment. No prohibited mathematical exposure was identified, and subsequent verification confirmed the submitted proofs.

4.1 PROXIMAL GRADIENT DESCENT METHOD (PGM)

Let f be convex and L-smooth, let g be closed proper convex, and let $x_\star$ minimize $h = f + g$. Consider

$$x_{k+1} = \mathrm{prox}_{g/L}\left(x_k - L^{-1}\nabla f(x_k)\right).$$

Both PGM conditions recover $h(x_N) - h(x_\star) \leq \frac{L}{4N}\|x_0 - x_\star\|^2$ for every $N \geq 1$ (Teboulle & Vaisbourd, 2023).

To illustrate the mathematical output, the first attempt without supplied SOS yields, for $1 \leq k < N$,

$$V_k = \frac{k+1}{2N-k}\left(f(x_k) - f(x_\star)\right) + \frac{k}{2N-k}\left(g(x_k) - g(x_\star)\right) - \frac{1}{2N-k}\langle x_k - x_\star, \nabla f(x_k)\rangle$$
$$- \frac{L}{4N}\|x_0 - x_\star\|^2 + \frac{L(N-k)}{(2N-k)^2}\|x_k - x_\star\|^2 - \frac{k}{2L(2N-k)^2}\|\nabla f(x_k) - \nabla f(x_\star)\|^2.$$

With separately defined endpoints, the proof establishes $V_N \leq \cdots \leq V_0 = 0$ and the desired bound. The first supplied-SOS attempt has a different gradient-square coefficient; both constructions are reproduced in Section E.3.

5 CASE STUDIES ON OPEN PROBLEMS

In this section, we show that `Peppy` can obtain novel convergence analyses for settings in which analytic analyses had not previously been developed, including some problems that had remained open for about a decade.

We close Conjecture 4 and Conjecture 5 of Taylor et al. (2017c) by providing the first analytic proofs of tight convergence bounds for Nesterov's Fast Gradient Method (FGM) (Nesterov, 1983). FGM is a classical accelerated method dating back to 1983, yet its tight worst-case bounds had not been established analytically. These bounds were conjectured in Taylor et al. (2017c), based on numerical values obtained by PEP roughly a decade ago, and had remained open since then. We resolve these conjectures by providing Lyapunov analyses obtained by `Peppy`. Additionally, we close related conjectures for OGM raised in Taylor et al. (2017c, Conjecture 4) and for FGM and FISTA with rational coefficients raised in Taylor et al. (2017a, Table 1).

For the conjectures considered in this section, we used the same input format as in Section 3.1, setting only the conjectured-rate field to the rate stated in the corresponding conjecture. We used `GPT-6 Astra` with `Ultra` reasoning effort.

As the proofs grew more complex, the Codex-generated drafts required human editing to make them readable. In particular, the drafts introduced many unnecessary substitutions; we had to simplify these and choose variable names and notation that made the arguments easier to follow. Detailed proofs are provided in Appendix F.

5.1 TIGHT CONVERGENCE ANALYSIS OF FGM USING PEPPY: CLOSING CONJECTURES 4 AND 5 OF TAYLOR ET AL. (2017C)

For a parameter sequence θ_k and a scalar η, FGM, OGM, and FGM with rational coefficients start from $x_0 = y_0 = z_0$ and use, for $0 \leq k < N$, the following updates:

$$\begin{aligned} y_{k+1} &= x_k - \frac{1}{L}\nabla f(x_k), \\ z_{k+1} &= z_k - \frac{\eta\theta_k}{L}\nabla f(x_k), \\ x_{k+1} &= \left(1 - \frac{1}{\theta_{k+1}}\right) y_{k+1} + \frac{1}{\theta_{k+1}} z_{k+1}. \end{aligned} \tag{3}$$

We first consider FGM, which selects

$$\theta_0 = 1, \qquad \theta_{k+1} = \frac{1 + \sqrt{1 + 4\theta_k^2}}{2} \quad (k \geq 0), \qquad \eta = 1. \tag{FGM}$$

Theorem 5.1 (FGM at y_N; Conjecture 4 in Taylor et al. (2017c))**.** *Let $f \in \mathcal{F}_L$. Fix an integer $N \geq 1$ and $R \geq 0$, and assume $\|x_0 - x_\star\| \leq R$. Then FGM satisfies*

$$f(y_N) - f(x_\star) \leq \frac{LR^2\theta_{N-1}^2}{2\left(2\sum_{j=0}^{N-1}\theta_j^3 + \theta_{N-1}^2\right)}. \tag{4}$$

Proof outline. Set $D_j = 2C_N + 1 - C_j$ and $\tau_N = 1/(2D_0)$. For the interior iterates, set

$$\boldsymbol{w}_k = \begin{bmatrix} z_{k+1} - x_\star \\ y_{k+1} - z_{k+1} \end{bmatrix}$$

and use $\mathbf{Q}_k$, whose diagonal entries are rational expressions in $a_k, C_{k+1}, D_0, \theta_k$, and θ_{N-1}. Define $V_0 = 0$,

$$V_k = a_k\big(f(x_k) - f(x_\star)\big) - \tau_N L\,\|x_0 - x_\star\|^2 - \frac{a_k}{2L}\,\|\nabla f(x_k)\|^2 + L\boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k$$

for the interior iterates, and

$$V_N = f(y_N) - f(x_\star) - \tau_N L\,\|x_0 - x_\star\|^2.$$

Each decrement is an exact positive weighted sum of smooth convex interpolation residuals minus one or two nonnegative squares. The explicit coefficients and their positivity are verified in Appendix F.2. Hence $V_{k+1} \leq V_k$, so $V_N \leq V_0 = 0$, which gives (4).

□

Theorem 5.2 (FGM at x_N; Conjecture 5 in Taylor et al. (2017c))**.** *Let $f \in \mathcal{F}_L$. Fix an integer $N \geq 1$ and $R \geq 0$, and assume $\|x_0 - x_\star\| \leq R$. Then FGM satisfies*

$$f(x_N) - f(x_\star) \leq \frac{LR^2\theta_N^2}{2\left(2\sum_{j=0}^{N}\theta_j^3 - \theta_N^2\right)}. \tag{5}$$

Proof. See Appendix F.3 for the proof. □

The bounds in Theorems 5.1 and 5.2 are tight, matching the lower bounds established in Lemma F.12 using the examples of Taylor et al. (2017c, Section 4.2, Conjectures 4–5). Lemma F.11 gives the correspondence with their notation.

5.1.1 OGM AT y_N

For the bound at y_N, OGM uses the same recursive parameters as FGM through θ_{N-1} and doubles the coefficient in the z-update:

$$\theta_0 = 1, \qquad \theta_{k+1} = \frac{1 + \sqrt{1 + 4\theta_k^2}}{2} \quad (0 \leq k < N-1), \qquad \eta = 2. \tag{OGM}$$

Theorem 5.3 (OGM at y_N; Conjecture 4 in Taylor et al. (2017c))**.** *Let $f \in \mathcal{F}_L$. Fix an integer $N \geq 1$ and $R \geq 0$, and assume $\|x_0 - x_\star\| \leq R$. Then OGM satisfies*

$$f(y_N) - f(x_\star) \leq \frac{LR^2}{2(2\theta_{N-1}^2 + 1)}. \tag{6}$$

Proof. See Appendix F.4 for the proof. □

The bound in Theorem 5.3 matches the lower bound in Kim & Fessler (2017, Proposition 4.1) and is therefore tight.

5.2 FGM AND FISTA WITH RATIONAL COEFFICIENTS

We now consider the conjectures in Taylor et al. (2017a, Table 1) for FGM and FISTA with rational coefficients. For FGM, the common update (3) uses the parameters

$$\theta_k = \frac{k+2}{2} \quad (k \geq 0), \qquad \eta = 1. \tag{FGM-rational}$$

It is worth noting that Berg Thomsen et al. (2026, Section 4.2) investigate sparsity patterns in numerical PEP proof certificates for this FGM bound. Theorem 5.4 establishes this conjectured rate.

Theorem 5.4 (FGM with rational coefficients; conjecture in Table 1 of Taylor et al. (2017a))**.** *Let $f \in \mathcal{F}_L$. Fix an integer $N \geq 1$ and $R \geq 0$, and assume $\|x_0 - x_\star\| \leq R$. Then FGM-rational satisfies*

$$f(y_N) - f(x_\star) \leq \frac{2LR^2}{(N+2)(N+3)}. \tag{7}$$

Proof. See Appendix F.1 for the proof. □

Combining this theorem with Lemma F.12 shows that (7) is attained and therefore sharp.

We next consider FISTA with rational coefficients for a composite objective $F = f + g$. Starting from $x_0 = y_0$, the method uses, for $0 \leq k < N$,

$$\begin{aligned} y_{k+1} &= \operatorname{prox}_{g/L}\left(x_k - \frac{1}{L}\nabla f(x_k)\right), \\ x_{k+1} &= y_{k+1} + \frac{k}{k+3}(y_{k+1} - y_k), \end{aligned} \tag{FISTA-rational}$$

We report y_N. This is the FPGM1 variant in Taylor et al. (2017a, Table 1); the following theorem resolves its nonsmooth composite conjecture.

Theorem 5.5 (FISTA with rational coefficients; nonsmooth FPGM1 conjecture in Table 1 of Taylor et al. (2017a))**.** *Let $f \in \mathcal{F}_L$, and let $g : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ be proper, closed, and convex. Set $F = f + g$. Fix an integer $N \geq 1$ and $R \geq 0$, and suppose $0 \in \nabla f(x_\star) + \partial g(x_\star)$ and $\|x_0 - x_\star\| \leq R$. Then FISTA-rational satisfies*

$$F(y_N) - F(x_\star) \leq \frac{2LR^2}{N^2 + 5N + 2}.$$

Proof. See Appendix F.5 for the proof. □

The bound in Theorem 5.5 is tight, matching the lower bound established in Lemma F.13 using the constrained linear example of Taylor et al. (2017a, Section 4.2.2, Table 1 and the following paragraph).

6 CONCLUSION

We present `Peppy`, an AI-assisted workflow for first-order optimization, built on the domain-specific machinery of PEP and the systematic framework of Yoon et al. (2026) for converting numerical PEP certificates into analytic Lyapunov analyses. `Peppy` equips AI agents with the `PEPFlow` library, which abstracts the domain-specific context and language into an accessible interface. Our particular Lyapunov proof structures support simple and rapid verification of the proposed proofs. This enables users to efficiently search for verified proofs with `Peppy` using only a high-level understanding of the workflow, as demonstrated by our resolution of conjectures on Nesterov's FGM and other methods. Future work could extend `Peppy` to support a broader range of proof structures and thereby cover more algorithms and forms of convergence analysis. We believe that `Peppy` is an important step toward exploiting AI for making systematic and human-understandable advances in optimization.

AI USE STATEMENT

To demonstrate what AI agents can contribute to optimization analysis, we used Codex for substantive research assistance in implementing PEPs, analyzing certificates, conjecturing coefficient formulas, constructing Lyapunov functions, and drafting and checking proofs. It also assisted with workflow and evaluation-protocol development, interpretation of experimental records, manuscript drafting and editing, and preparation of notebooks and supporting code. Model and reasoning settings for the reported runs are specified in Section 4 and Appendix F. Verification combined numerical certificate checks, symbolic identity checks, and the analytic sign, domain, and endpoint arguments reported in the proofs. The authors edited the proof presentations and take responsibility for the final claims and manuscript.

REPRODUCIBILITY STATEMENT

Sections 2 and 3, with Appendices C and D, describe the workflow and its numerical and symbolic verification procedures. Appendix E records the experimental inputs, model settings, protocol, and representative generated proofs. The assumptions and detailed case-study proofs appear in Appendix F, with references to the corresponding notebook checks. These materials support verification of the reported arguments; rerunning AI-assisted discovery can produce different proof expressions and computational costs.

REFERENCES

Daniel Berg Thomsen, Manu Upadhyaya, Baptiste Goujaud, Aymeric Dieuleveut, and Adrien Taylor. Finding simple proofs for first-order optimization. *arXiv:2607.08753*, 2026.

Nizar Bousselmi, Julien M. Hendrickx, and François Glineur. Interpolation conditions for linear operators and applications to performance estimation problems. *SIAM Journal on Optimization*, 34(3):3033–3063, 2024.

Sébastien Bubeck, Christian Coester, Ronen Eldan, Timothy Gowers, Yin Tat Lee, Alexandru Lupsasca, Mehtaab Sawhney, Robert Scherrer, Mark Sellke, Brian K. Spears, Derya Unutmaz, Kevin Weil, Steven Yin, and Nikita Zhivotovskiy. Early science acceleration experiments with GPT-5. *arXiv:2511.16072*, 2025.

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*, 204:567–639, 2024.

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

Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis, and Pushmeet Kohli. Advancing mathematics by guiding human intuition with AI. *Nature*, 600:70–74, 2021.

Radu-Alexandru Dragomir, Adrien B Taylor, Alexandre d'Aspremont, and Jérôme Bolte. Optimal complexity and certification of Bregman first-order methods. *Mathematical Programming*, 194 (1):41–83, 2022.

Yoel Drori. The exact information-based complexity of smooth convex minimization. *Journal of Complexity*, 39:1–16, 2017.

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.

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. *arXiv:2602.10177*, 2026.

Bogdan Georgiev, Javier Gómez-Serrano, Terence Tao, and Adam Zsolt Wagner. Mathematical exploration and discovery at scale. *arXiv:2511.02864*, 2025.

Baptiste Goujaud, Aymeric Dieuleveut, and Adrien Taylor. On fundamental proof structures in first-order optimization. In *Conference on Decision and Control*, 2023.

Baptiste Goujaud, Céline Moucer, Francois Glineur, Julien Hendrickx, Adrien Taylor, and Aymeric Dieuleveut. PEPit: Computer-assisted worst-case analyses of first-order optimization methods in Python. *Mathematical Programming Computation*, 16:337–367, 2024.

Benjamin Grimmer and Alex L. Wang. Implicit primal-dual guarantees in unconstrained first-order minimization. *arXiv:2607.20875*, 2026.

Uijeong Jang and Ernest K. Ryu. Point convergence of Nesterov's accelerated gradient method: An AI-assisted proof. *arXiv:2510.23513*, 2026.

Uijeong Jang, Shuvomoy Das Gupta, and Ernest K. Ryu. Computer-assisted design of accelerated composite optimization methods: OptISTA. *Mathematical Programming*, 2025.

Donghwan Kim. Accelerated proximal point method for maximally monotone operators. *Mathematical Programming*, 190(1–2):57–87, 2021.

Donghwan Kim and Jeffrey A. Fessler. Optimized first-order methods for smooth convex minimization. *Mathematical Programming*, 159(1–2):81–107, 2016.

Donghwan Kim and Jeffrey A. Fessler. On the convergence analysis of the optimized gradient method. *Journal of Optimization Theory and Applications*, 172(1):187–205, 2017.

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.

Heechang Kim, Ernest K. Ryu, and Shuvomoy Das Gupta. A domain-specific harness for end-to-end automation of optimization research. *arXiv:2608.07407*, 2026.

Jaeyeon Kim, Asuman E. Ozdaglar, Chanwoo Park, and Ernest K. Ryu. Time-reversed dissipation induces duality between minimizing gradient norm and function value. In *Neural Information Processing Systems*, 2023.

Jungbin Kim. A proof of the exact convergence rate of gradient descent. *arXiv:2412.04427*, 2025.

Jongmin Lee, Chanwoo Park, and Ernest K. Ryu. A geometric structure of acceleration and its role in making gradients small fast. In *Neural Information Processing Systems*, 2021.

Sucheol Lee and Donghwan Kim. Fast extra gradient methods for smooth structured nonconvex-nonconcave minimax problems. In *Neural Information Processing Systems*, 2021.

Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

Felix Lieder. On the convergence rate of the Halpern-iteration. *Optimization Letters*, 15(2):405–418, 2021.

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

Jianhao Ma and Yuxin Chen. A lower bound for stepsize-based acceleration of gradient descent. *arXiv:2608.10418*, 2026.

Shiqian Ma. Convergence of BDRS as a matrix scaling algorithm. Optimization Online, 2026.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Doklady Akademii Nauk SSSR*, 269(3):543–547, 1983.

Chanwoo Park, Jisun Park, and Ernest K. Ryu. Factor-$\sqrt{2}$ acceleration of accelerated gradient methods. *Applied Mathematics & Optimization*, 88(3):77, 2023.

Jisun Park and Ernest K Ryu. Exact optimal accelerated complexity for fixed-point iterations. In *International Conference on Machine Learning*, 2022.

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*, 625:468–475, 2024.

Teodor Rotaru, Francois Glineur, and Panagiotis Patrinos. Exact worst-case convergence rates of gradient descent: a complete analysis for all constant stepsizes over nonconvex and convex functions. *arXiv:2406.17506*, 2024.

Ernest K. Ryu, Adrien B. Taylor, Carolina Bergeling, and Pontus Giselsson. Operator splitting performance estimation: Tight contraction factors and optimal parameter selection. *SIAM Journal on Optimization*, 30(3):2251–2271, 2020.

Jaewook J. Suh, Bicheng Ying, Xin Jiang, and Edward Duc Hien Nguyen. PEPFlow: A Python library for the workflow of performance estimation of optimization algorithms. In *Neural Information Processing Systems Workshop on GPU-accelerated and Scalable Optimization*, 2025.

Adrien Taylor and Francis Bach. Stochastic first-order methods: Non-asymptotic and computer-aided analyses via potential functions. In *Conference on Learning Theory*, 2019.

Adrien Taylor, Bryan Van Scoy, and Laurent Lessard. Lyapunov functions for first-order methods: Tight automated convergence guarantees. In *International Conference on Machine Learning*, 2018.

Adrien B. Taylor and Yoel Drori. An optimal gradient method for smooth strongly convex minimization. *Mathematical Programming*, 199(1–2):557–594, 2023.

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, 2017a.

Adrien B. Taylor, Julien M. Hendrickx, and Francois Glineur. Performance estimation toolbox (PESTO): Automated worst-case analysis of first-order optimization methods. In *Conference on Decision and Control*, 2017b.

Adrien B. Taylor, Julien M. Hendrickx, and François Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. *Mathematical Programming*, 161:307–345, 2017c.

Marc Teboulle and Yakov Vaisbourd. An elementary approach to tight worst case complexity analysis of gradient based methods. *Mathematical Programming*, 201(1):63–96, 2023.

Amitayush Thakur, George Tsoukalas, Yeming Wen, Jimmy Xin, and Swarat Chaudhuri. An in-context learning agent for formal theorem-proving. In *First Conference on Language Modeling*, 2024.

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625:476–482, 2024.

George Tsoukalas, Anton Kovsharov, Sergey Shirobokov, Anja Surina, Moritz Firsching, Gergely Bérczi, Francisco J. R. Ruiz, Arun Suggala, Adam Zsolt Wagner, Eric Wieser, Lei Yu, Aja Huang, Miklós Z. Horváth, Andrew Ferraiuolo, Henryk Michalewski, Edward Lockhart, Codrut Grosu, Thomas Hubert, Matej Balog, Pushmeet Kohli, and Swarat Chaudhuri. Advancing mathematics research with AI-driven formal proof search. *arXiv:2605.22763*, 2026.

Manu Upadhyaya, Sebastian Banert, Adrien B. Taylor, and Pontus Giselsson. Automated tight Lyapunov analysis for first-order methods. *Mathematical Programming*, 209(1):133–170, 2025.

Manu Upadhyaya, Shuvomoy Das Gupta, Adrien B. Taylor, Sebastian Banert, and Pontus Giselsson. The AutoLyap software suite for computer-assisted Lyapunov analyses of first-order methods. *arXiv:2506.24076*, 2026.

Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. LeanDojo: Theorem proving with retrieval-augmented language models. In *Neural Information Processing Systems*, 2023.

Yuhan Ye and Kaizhao Liu. Improved gradient descent lower bounds beyond Nesterov. *arXiv:2609.02855*, 2026.

TaeHo Yoon and Ernest K. Ryu. Accelerated algorithms for smooth convex-concave minimax problems with $\mathcal{O}(1/k^2)$ rate on squared gradient norm. In *International Conference on Machine Learning*, 2021.

TaeHo Yoon, Jaeyeon Kim, Jaewook J. Suh, and Ernest K. Ryu. Optimal acceleration for minimax and fixed-point problems is not unique. In *International Conference on Machine Learning*, 2024.

TaeHo Yoon, Jaewook J. Suh, Edward Duc Hien Nguyen, Bicheng Ying, and Shiqian Ma. Toward a systematic understanding and interactive search of Lyapunov-style proofs in optimization. *arXiv:2606.26077*, 2026.

A NOTATION

We write $\mathcal{H}$ for a nonzero real Hilbert space, equipped with inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\|\cdot\|$. For $L > 0$, $\mathcal{F}_L$ denotes the class of differentiable convex functions $f : \mathcal{H} \to \mathbb{R}$ that are L-smooth, i.e., $\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|$ for all $x, y \in \mathcal{H}$. For a proper, closed, convex function $g : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ and $\gamma > 0$, we write

$$\operatorname{prox}_{\gamma g}(v) := \operatorname*{arg\,min}_{w \in \mathcal{H}} \left\{ \gamma g(w) + \frac{1}{2} \|w - v\|^2 \right\}.$$

For positive integers m, n, $\mathbb{R}^{m \times n}$ denotes the space of real $m \times n$ matrices. We write $\mathbb{S}^n$ and $\mathbb{S}^n_+$ for the spaces of real symmetric and real symmetric positive semidefinite $n \times n$ matrices, respectively. For a matrix A, A^T, $\operatorname{Tr}(A)$, $\operatorname{rank}(A)$, and $\operatorname{col}(A)$ denote its transpose, trace, rank, and column space. The i-th standard basis vector in a Euclidean space of dimension at least i is denoted by $\mathbf{e}_i$.

For minimization problems, we assume that the objective attains its minimum and denote a minimizer by $x_\star$. We use $\star$ as an index for an optimal point and reserve N for the terminal iteration, or horizon, at which the performance measure is evaluated. For endpoint coefficient formulas, we adopt the convention $0/0 = 0$.

B PRIOR WORK

B.1 PERFORMANCE ESTIMATION PROBLEMS

The performance estimation problem (PEP) framework formulates the worst-case performance of a prescribed optimization algorithm as an optimization problem over admissible functions and iterates. Drori & Teboulle (2014) introduced this approach, and Taylor et al. (2017c) developed exact smooth strongly convex interpolation conditions that yield finite-dimensional semidefinite formulations. Extensions cover composite convex optimization (Taylor et al., 2017a), Bregman geometry (Dragomir et al., 2022), operator splitting (Ryu et al., 2020), and methods involving linear operators (Bousselmi et al., 2024).

Dual feasible solutions supply weights on interpolation inequalities and positive semidefinite matrix certificates that together certify a performance bound. Numerical solutions thus provide concrete guidance for analytic proof construction. Turning these certificates into formulas valid for arbitrary iteration horizons remains a separate mathematical task. This is the step that `Peppy` addresses through a sequence of numerical, structural, and symbolic computations.

B.2 DISCOVERIES ENABLED BY PEP

PEP has supported both sharper analyses of existing methods and the design of new methods. Early examples include tight bounds for constant-step GD (Drori & Teboulle, 2014) and the optimized gradient method (OGM) (Kim & Fessler, 2016), whose worst-case objective bound for the horizon-dependent secondary output x_N was later matched by an oracle lower bound (Drori, 2017). Subsequent developments include OGM-G for reducing the gradient norm (Kim & Fessler, 2021), the information-theoretic exact method (ITEM) for smooth strongly convex minimization (Taylor & Drori, 2023), and OptISTA for composite optimization (Jang et al., 2025). The branch-and-bound PEP framework (Das Gupta et al., 2024) extends this design process to nonconvex quadratically constrained quadratic formulations in which the algorithm coefficients themselves are optimization variables.

Beyond minimization, this line of work includes the accelerated proximal point method (Kim, 2021), tight Halpern iteration bounds (Lieder, 2021; Park & Ryu, 2022), and accelerated methods for minimax problems such as EAG and FEG (Yoon & Ryu, 2021; Lee & Kim, 2021). Numerical relations between algorithm coefficients have also motivated H-duality, connecting function-value and gradient-norm guarantees (Kim et al., 2023), with extensions to minimax and fixed-point problems (Yoon et al., 2024). These results illustrate the mathematical insight required to interpret PEP outputs, and motivate the structured proof-search tasks used in `Peppy`.

B.2.1 TIGHT ANALYSIS OF FIXED ALGORITHMS

Performance estimation has also shown that obtaining a tight rate for a prescribed algorithm can be difficult even when that algorithm is not minimax optimal. For gradient descent (GD), Drori & Teboulle (2014) derived a tight analytical bound in the smooth convex setting, and Taylor et al. (2017c) subsequently used exact smooth strongly convex interpolation to formulate finite-dimensional PEPs and numerically conjecture worst-case rates under several performance criteria. It was only recently that Kim (2025) proved the exact function-value convergence rate of GD for smooth strongly convex optimization, settling conjectures posed in the aforementioned works. In parallel, Rotaru et al. (2024) obtained a complete exact analysis of the minimum gradient norm for all constant stepsizes across convex, strongly convex, and weakly convex classes. Their proof requires inequalities coupling nonconsecutive iterates. This progression from numerical evidence to analytic proofs for GD itself illustrates that the tight analysis of a fixed, suboptimal method is a nontrivial problem. Conventional convergence rate analysis does not in general give the sharp finite-horizon constant or a matching worst-case instance.

B.2.2 RECENT EVIDENCE AND EXACT ANALYSES FOR ACCELERATED METHODS

In their study of implicit primal–dual guarantees, Grimmer & Wang (2026, Section 6.2) note that an exactly optimal PEP certificate for the Nesterov FGM variant they analyze had not been determined analytically, and they instead use a feasible, nearly optimal certificate. Using chained, iteration-dependent quadratic Lyapunov functions over prescribed state and oracle-evaluation vectors, Upadhyaya et al. (2026, Section 3.2 and Figure 3) obtain numerical finite-horizon constants for Nesterov's FGM that are smaller than the classical rates in their comparison. Identifying Grimmer and Wang's t_n with our θ_n, their initialization of z_1 is the first z-update in (3), and their subsequent recurrences agree with (3) and (FGM). Their criterion $f(x_N) - f(x_\star)$, under $\|x_0 - x_\star\| \leq D$, therefore corresponds to the FGM output x_N in Conjecture 5 of Taylor et al. (2017c), rather than the output y_N in Conjecture 4. Theorem 5.2 gives an analytic Lyapunov proof of the corresponding tight finite-horizon bound for every N. Berg Thomsen et al. (2026) investigate the sparsification of PEP multipliers to obtain simpler proof certificates.

For the primary OGM output y_N, Kim & Fessler (2017, Theorem 4.1) prove the upper bound $LR^2/(4\theta_{N-1}^2)$, while their Proposition 4.1 constructs an instance attaining $LR^2/[2(2\theta_{N-1}^2+1)]$. These expressions agree asymptotically but leave a finite-horizon gap. Theorem 5.3 proves the latter expression as an upper bound for the same output y_N, closing the finite-horizon gap. This result is separate from the already exact guarantee $LR^2/(2\theta_N^2)$ for OGM's secondary last iterate x_N, which uses the horizon-dependent terminal parameter $\theta_N = (1+\sqrt{1+8\theta_{N-1}^2})/2$ and is attained in Kim & Fessler (2017, Theorem 5.1).

B.3 COMPUTER-ASSISTED LYAPUNOV ANALYSIS

A Lyapunov proof expresses progress through a scalar quantity V_k whose decrease implies the desired convergence guarantee. Identifying an appropriate form of V_k is often the central difficulty. Integral quadratic constraints (Lessard et al., 2016) and semidefinite searches over potential functions (Taylor et al., 2018; Taylor & Bach, 2019) provide systematic approaches to this task. Upadhyaya et al. (2025) give necessary and sufficient conditions for a Lyapunov inequality to exist within a prescribed family. These approaches allow automated searches once the candidate state vectors and functional form have been chosen. Berg Thomsen et al. (2026) use a candidate-lemma SDP to identify Lyapunov potentials and their one-step decrements for proximal point and accelerated proximal point methods.

A complementary approach starts from a PEP certificate and seeks a concise Lyapunov representation of it. The history of accelerated methods illustrates this need: matrix-based certificates for methods such as OGM and OGM-G were subsequently developed into potential-function or geometric arguments (d'Aspremont et al., 2021; Lee et al., 2021; Park et al., 2023). Goujaud et al. (2023, Section 6) discuss forming potentials from partial sums of the terms in a PEP proof. Building on this perspective, Yoon et al. (2026) develop a systematic characterization and procedure for converting PEP proofs into Lyapunov analyses while preserving their certificates, including the search for a concise representation in suitable basis vectors.

That conversion framework provides the theoretical basis for Peppy. The present paper organizes its application into tasks that an AI agent can execute: implementing the PEP, inferring analytic certificate formulas, constructing and simplifying V_k, and checking its difference and boundary conditions. The resulting workflow complements searches over predefined Lyapunov families by using the structure of the PEP certificate to guide the choice of representation.

B.4 Software for performance estimation

Software packages make these computational approaches accessible. PESTO (Taylor et al., 2017b) provides a MATLAB toolbox for PEP, while PEPit (Goujaud et al., 2024) offers an open-source Python interface for modeling and solving such problems. AutoLyap (Upadhyaya et al., 2026) supports computer-assisted searches over Lyapunov templates. Its fixed-horizon numerical certificates do not give explicit formulas for the certificate coefficients or a single analytic proof valid for every horizon. Numerical agreement with a tight FGM rate sequence does not by itself identify the same algorithm, reported iterate, or Lyapunov function as in the conjectures of Taylor et al. (2017c), and therefore does not settle them. The approaches are complementary: AutoLyap searches over a prescribed Lyapunov family in a general-purpose state and evaluation basis, while Peppy starts from a tight PEP certificate and seeks a compressed basis adapted to the algorithm, summation-free formulas, and proofs of the required identities and coefficient signs valid for every horizon.

Peppy uses PEPFlow (Suh et al., 2025) as its computational backend. Its tagging of iterates, function values, and gradient or operator evaluations makes the resulting vectors and matrices accessible in mathematical notation. This is useful when manipulating numerical certificates and recovering analytic expressions. The certificate-processing routines in `lyapunov_utils.py`, discussed in Section C.2, support the conversion steps. Peppy adds reusable agent instructions, intermediate state records, and numerical and symbolic checks around these operations.

B.5 AI for mathematics

AI-assisted mathematical research includes discovering conjectures, searching for explicit constructions, and developing proofs. Davies et al. (2021) combine machine learning with attribution methods to identify relationships among mathematical objects and guide mathematicians toward conjectures and theorems in topology and representation theory. FunSearch (Romera-Paredes et al., 2024) combines a language model with program evaluation and evolutionary search, producing constructions for the cap-set problem and heuristics for online bin packing. Feng et al. (2026) introduce Aletheia, a mathematical research agent that iteratively generates, verifies, and revises solutions and is capable of solving research-level problems in mathematics. Tsoukalas et al. (2026) use agents to leverage LLMs to generate formal proofs in languages like Lean to successfully solve Erdős problems.

A related direction couples learned search with symbolic reasoning. AlphaGeometry (Trinh et al., 2024) uses a language model to propose auxiliary geometric constructions and a symbolic engine to derive their consequences. LeanDojo (Yang et al., 2023) provides tools for interacting with Lean and develops a retrieval-augmented prover that selects relevant premises from a mathematical library. The feedback in these systems takes different forms, including numerical evaluation, symbolic deduction, and proof-assistant checking, depending on the task and its representation.

Peppy applies this general approach to convergence analysis. Its search is organized around numerical PEP certificates and their conversion into analytic Lyapunov proofs. The agent seeks formulas valid beyond the finite horizons used in numerical experiments and checks the resulting identities symbolically, together with the inequalities and boundary arguments required for the claimed rate. The central proof objects are a Lyapunov function, its one-step difference, and the endpoint estimates that together yield the convergence analysis.

Kim et al. (2026) integrate the PEP framework with LLMs. Their approach differs from Peppy in its level of granularity. Given a user-specified problem class, their AutoOpt domain-specific harness uses the branch-and-bound performance estimation programming (BnB-PEP) framework to automatically construct a nonconvex quadratically constrained quadratic program (QCQP). Solving this QCQP provides numerical guidance about an optimal algorithm for the problem class and its convergence-rate proof. The harness uses high-level prompts to an LLM to carry out the typically

human step of converting these numerical outputs into an analytic result. The resulting convergence analysis is then formally verified in Lean and left for humans to interpret and verify.

B.6 AI FOR OPTIMIZATION THEORY

There have been various works that have leveraged LLMs to produce research-level results in optimization, specifically first-order methods. Jang & Ryu (2026) leverage GPT-5 Pro to assist in proving the point convergence of Nesterov's accelerated gradient method. Ma (2026) uses a proof generated by ChatGPT 5.5 and verified by the author to establish convergence of BDRS under a verifiable condition when it is viewed as a matrix scaling algorithm. Ma & Chen (2026) improve the lower bound for the last-iterate convergence rate of gradient descent with predetermined nonnegative stepsize schedules to show that stepsize schedules alone cannot accelerate GD to the convergence rate achieved by Nesterov's accelerated method through using GPT-5.6 Sol Pro. Ye & Liu (2026) further improve the lower bound established by Ma & Chen (2026) for GD with assistance from GPT-5.6 Sol Pro. The aforementioned works are a non-exhaustive representation of works in convergence analysis of first-order methods that employ LLMs to achieve state-of-the-art results.

C OMITTED DETAILS OF SECTION 2.4

Here, we briefly explain Step 4, which is implemented under the hood in `Peppy`, using GD as a representative example. For a more detailed explanation, we refer the readers to Yoon et al. (2026).

C.1 MATHEMATICAL CONSTRUCTION: THE GD EXAMPLE

The PEP proof for gradient descent (GD) $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$ for convex and L-smooth f is equivalent to establishing the *Lyapunov analysis* $V_{k+1} \le V_k$ for each $k = 0, \dots, N-1$, where we define

$$V_k = \sum_{i=1}^{k} \lambda_{i-1,i} I(x_{i-1}, x_i) + \sum_{i=0}^{k} \lambda_{\star,i} I(x_\star, x_i) - \sum_{i=0}^{k} d_i \|s_i\|^2 \tag{8}$$

where

$$\lambda_{i-1,i} = \frac{i}{2N+1-i}, \quad i = 1, \dots, N, \qquad \lambda_{\star,i} = \begin{cases} \lambda_{0,1} & i = 0 \\ \lambda_{i,i+1} - \lambda_{i-1,i} & i = 1, \dots, N-1 \\ 1 - \lambda_{i-1,i} & i = N, \end{cases}$$

$$d_i = \frac{L}{2} \frac{4Ni + 2N - 2i^2 + 1}{(2N-i)^2}, \quad i = 0, \dots, N-1$$

$$d_N = \frac{L}{2},$$

$$s_i = \frac{1}{2N+1-i}(x_i - x_\star) - \frac{1}{L}\nabla f(x_i)$$

$$I(x_{i-1}, x_i) = \left(f(x_i) - f(x_{i-1}) + \langle \nabla f(x_i), x_{i-1} - x_i \rangle + \frac{1}{2L} \|\nabla f(x_{i-1}) - \nabla f(x_i)\|^2 \right) \le 0$$

$$I(x_\star, x_i) = \left(f(x_i) - f(x_\star) + \langle \nabla f(x_i), x_\star - x_i \rangle + \frac{1}{2L} \|\nabla f(x_i)\|^2 \right) \le 0.$$

Chaining the inequalities, we obtain $V_N \le \cdots \le V_1 \le 0$, and this would imply $f(x_N) - f(x_\star) \le \frac{LR^2}{4N+2}$. Stage 4 checks whether each V_k can be simplified into a summation-free formula written in terms of a small number of vectors, not increasing with k. Direct computation, either numerical or symbolic, quickly reveals that the function value terms in (8) simplify to

$$\lambda_{k,k+1}(f(x_k) - f(x_\star)) = \frac{k+1}{2N-k}(f(x_k) - f(x_\star)).$$

However, systematically checking that the sum of all inner product and squared norm terms can be simplified is a more involved process. Below, we outline how this is performed in our code `gd_recover_example_lyap.ipynb`.

Define the sum of only the inner product terms, excluding the function values, within (8) as

$$\begin{aligned}\widetilde{V}_k :=& \sum_{i=1}^{k} \lambda_{i-1,i} \left(\langle \nabla f(x_i), x_{i-1} - x_i \rangle + \frac{1}{2L} \|\nabla f(x_{i-1}) - \nabla f(x_i)\|^2 \right) \\ &+ \sum_{i=0}^{k} \lambda_{\star,i} \left(\langle \nabla f(x_i), x_\star - x_i \rangle + \frac{1}{2L} \|\nabla f(x_i)\|^2 \right) \\ &- \sum_{i=0}^{k} d_i \left\| \frac{1}{2N+1-i}(x_i - x_\star) - \frac{1}{L}\nabla f(x_i) \right\|^2 .\end{aligned} \tag{9}$$

The Gram formulation used in PEP allows us to rewrite this quantity in terms of tangible matrices that we can handle with linear algebraic tricks. Without loss of generality, assume $x_\star = \mathbf{0}$. We let

$$\mathbf{P} = [x_0 \quad \nabla f(x_0) \quad \cdots \quad \nabla f(x_N)] \in \mathbb{R}^{d \times (N+2)},$$

and define $\mathbf{x}_0 = \mathbf{e}_1$ and $\mathbf{g}_i = \mathbf{e}_{i+2}$ for $i = 0, \ldots, N$ so that $\mathbf{P}\mathbf{x}_0 = x_0$ and $\mathbf{P}\mathbf{g}_i = \nabla f(x_i)$. Define $\mathbf{x}_1, \ldots, \mathbf{x}_N$ by

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \frac{1}{L}\mathbf{g}_i = \mathbf{x}_0 - \frac{1}{L}\sum_{j=0}^{i} \mathbf{g}_j$$

for $i = 0, \ldots, N-1$ and let $\mathbf{x}_\star = \mathbf{0}$, so that $\mathbf{P}\mathbf{x}_i = x_i$ for all $i \in \{0, \ldots, N, \star\}$. Now consider the Gram matrix $\mathbf{G} = \mathbf{P}^\mathsf{T}\mathbf{P} \in \mathbb{S}_+^{N+2}$. Then we can express the inner product terms as, for example, for $i = 1, \ldots, k$,

$$\begin{aligned}\langle \nabla f(x_i), x_{i-1} - x_i \rangle &= \langle \mathbf{P}\mathbf{g}_i, \mathbf{P}(\mathbf{x}_{i-1} - \mathbf{x}_i) \rangle \\ &= \mathbf{g}_i^\mathsf{T}\mathbf{P}^\mathsf{T}\mathbf{P}(\mathbf{x}_{i-1} - \mathbf{x}_i) \\ &= \mathbf{g}_i^\mathsf{T}\mathbf{G}(\mathbf{x}_{i-1} - \mathbf{x}_i) = \mathrm{Tr}\left(\mathbf{G}\mathbf{g}_i(\mathbf{x}_{i-1} - \mathbf{x}_i)^\mathsf{T}\right).\end{aligned}$$

The last expression can be symmetrized: $\mathrm{Tr}\left(\mathbf{G}\mathbf{g}_i(\mathbf{x}_{i-1} - \mathbf{x}_i)^\mathsf{T}\right) = \mathrm{Tr}\left(\mathbf{G} \cdot \frac{1}{2}\left(\mathbf{g}_i(\mathbf{x}_{i-1} - \mathbf{x}_i)^\mathsf{T} + (\mathbf{x}_{i-1} - \mathbf{x}_i)\mathbf{g}_i^\mathsf{T}\right)\right)$. Applying the similar trick to all terms in (9), we can express $\widetilde{V}_k = \mathrm{Tr}(\mathbf{G}\mathbf{V}_k)$ with

$$\begin{aligned}\mathbf{V}_k :=& \sum_{i=1}^{k} \lambda_{i-1,i} \left(\mathbf{B}_{i-1,i} + \frac{1}{2L}\mathbf{C}_{i-1,i} \right) + \sum_{i=0}^{k} \lambda_{\star,i} \left(\mathbf{B}_{\star,i} + \frac{1}{2L}\mathbf{C}_{\star,i} \right) \\ &- \sum_{i=0}^{k} d_i \left(\frac{1}{(2N+1-i)^2}\mathbf{A}_{\star,i} + \frac{2}{L(2N+1-i)}\mathbf{B}_{\star,i} + \frac{1}{L^2}\mathbf{C}_{\star,i} \right)\end{aligned}$$

where $\mathbf{A}_{i,j}, \mathbf{B}_{i,j}, \mathbf{C}_{i,j} \in \mathbb{S}^{N+2}$ are the symmetric matrices

$$\begin{aligned}\mathbf{A}_{i,j} &= (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\mathsf{T} \\ \mathbf{B}_{i,j} &= \frac{1}{2}\left(\mathbf{g}_j(\mathbf{x}_i - \mathbf{x}_j)^\mathsf{T} + (\mathbf{x}_i - \mathbf{x}_j)\mathbf{g}_j^\mathsf{T}\right) \\ \mathbf{C}_{i,j} &= (\mathbf{g}_i - \mathbf{g}_j)(\mathbf{g}_i - \mathbf{g}_j)^\mathsf{T}.\end{aligned}$$

Now, if the matrices $\mathbf{V}_k$ have constant low rank r across k, then this indicates that each $\widetilde{V}_k$ can be expressed in terms of r vectors that are linear combinations of $x_0, \nabla f(x_0), \ldots, \nabla f(x_N)$. The process of identifying the low-rank structure of $\mathbf{V}_k$ is based on the following linear algebraic argument. If $\mathrm{rank}(\mathbf{V}_k) = r$ and $\mathcal{B}_k = \{\mathbf{v}_{k,1}, \ldots, \mathbf{v}_{k,r}\}$ is a basis of the column space $\mathrm{col}(\mathbf{V}_k)$, then there exists $\mathbf{C}_k \in \mathbb{R}^{r \times r}$ such that

$$\mathbf{V}_k = [\mathbf{v}_{k,1} \quad \cdots \mathbf{v}_{k,r}]\, \mathbf{C}_k\, [\mathbf{v}_{k,1} \quad \cdots \quad \mathbf{v}_{k,r}]^\mathsf{T}. \tag{10}$$

Then $\widetilde{V}_k$ can be expressed as inner products of $\mathbf{P}\mathbf{v}_{k,1}, \ldots, \mathbf{P}\mathbf{v}_{k,r}$, and the associated coefficients are the entries of $\mathbf{C}_k$. The process of finding $\mathcal{B}_k$ and $\mathbf{C}_k$ is as follows.

1. Compute $r = \text{rank}(\mathbf{V}_k)$. For GD, $r = 3$.
2. Build the set of candidate vectors as

$$\mathcal{T} = \{\mathbf{x}_0, \ldots, \mathbf{x}_N, \mathbf{x}_\star, \mathbf{g}_0, \ldots, \mathbf{g}_N\}, \quad \mathcal{S} = \mathcal{T} \cup \{\mathbf{u} - \mathbf{v} : \mathbf{u}, \mathbf{v} \in \mathcal{T}\}.$$

   Then, use the `vectors_in_column_space` function to determine the candidate vectors that belong to $\text{col}(\mathbf{V}_k)$. Denote the candidates as $\widetilde{\mathcal{B}}_k = \{\mathbf{v} \in \mathcal{S} : \mathbf{v} \in \text{col}(\mathbf{V}_k)\}$.
3. Use the `find_basis_with_sparsest_coefficients` function to search over linearly independent subsets of $\widetilde{\mathcal{B}}_k$ with r elements and select the one that yields the sparsest $\mathbf{C}_k$ in the decomposition (10). This will be our $\mathcal{B}_k$. In case of GD, we find $\mathcal{B}_k = \{\mathbf{x}_0 - \mathbf{x}_\star, \mathbf{x}_{k+1} - \mathbf{x}_\star, \mathbf{g}_k\}$.
4. Identify the closed form of the Lyapunov coefficients, i.e., the entries of $\mathbf{C}_k$. This yields the analytic expression of $\widetilde{V}_k$, and thus completes the simplification of V_k. For GD, we find

$$V_k = \frac{k+1}{2N-k}\big(f(x_k) - f(x_\star)\big) - \frac{L}{4N+2}\|x_0 - x_\star\|^2 - \frac{k+1}{2L(2N-k)}\|\nabla f(x_k)\|^2 + \frac{L(2N-2k-1)}{2(2N-k)^2}\|x_{k+1} - x_\star\|^2$$

C.2 Implementation in `lyapunov_utils.py`

The role of `lyapunov_utils.py` is to expose the linear-algebraic structure of a numerical candidate Lyapunov function in terms of mathematical vectors that an agent and a user can inspect. The inner-product part of a PEP scalar is stored as a matrix in a common Gram coordinate system. The module accesses this matrix and the coordinates of candidate vectors through `ExpressionManager` and then returns a short basis and a coefficient matrix for the same quadratic form. In other words, the module connects the quadratic part of V_k to expressions involving recognizable iterates, gradients, and auxiliary vectors.

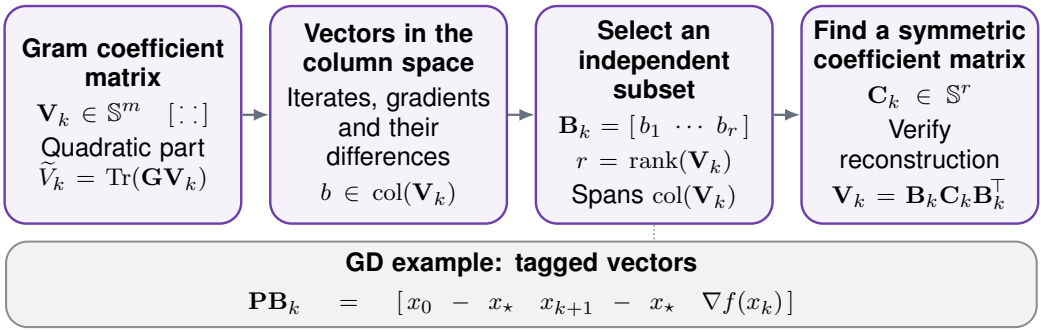


Figure 2: Recovering an interpretable representation of the quadratic part. Given candidates spanning the target column space, the routines select a basis and recover a small symmetric coefficient matrix whose reconstruction is checked numerically. Function value and constant terms are retained in the complete Lyapunov function. The GD example uses the notation of Section C.1.

C.2.1 Preserving mathematical names

The vector-tagging and display mechanisms belong to the underlying domain-specific library. The utilities preserve this interface by accepting and returning the original `Vector` objects, rather than replacing them with anonymous coordinate columns. Thus, when a basis is returned in a notebook, its entries retain their user-facing `repr` strings, and the rows and columns of the associated coefficient matrix can be read in that same order. This link between Gram coordinates and tagged vectors is what makes the numerical basis search useful for proof development.

C.2.2 FINDING AN INTERPRETABLE BASIS

The main basis-search path begins with `vectors_in_column_space`. Given a quadratic scalar and a list of mathematically motivated candidates, it uses an SVD projection test to keep the vectors that lie in the column space of the scalar's inner-product matrix. The user can then call `find_basis_with_sparsest_coefficients`, which enumerates independent subsets whose size is the numerical rank of the candidate coordinates and chooses a basis whose coefficient matrix has the largest number of numerically zero entries. Specified `fixed_vectors` can be required to appear in every tested basis. The simpler `select_independent_subset` routine greedily removes dependent vectors in input order. It is useful when the desired candidate ordering is already known, but it does not perform the sparsity search.

C.2.3 RECONSTRUCTING THE QUADRATIC FORM

For a proposed independent basis $\boldsymbol{v} = (v_1, \ldots, v_r)$, `find_symmetric_coefficient_matrix` checks that the column space of the target matrix is contained in $\mathrm{span}(\boldsymbol{v})$ and solves for the symmetric matrix C satisfying

$$V^{\mathrm{quad}} = \boldsymbol{v}^\top C \boldsymbol{v}.$$

This reconstruction is the basic numerical check used both inside the sparse-basis search and after an agent proposes a basis directly. A second, more specialized routine, `complete_basis_with_sparsifying_last_vector`, applies when all but one basis direction have been fixed. It finds the missing numerical direction in the column space and shifts it by the fixed directions, when possible, so that the last off-diagonal block of C vanishes. This option is useful when the candidate list does not already contain a convenient final vector. The resulting coordinate vector still must be interpreted as a mathematical expression.

C.2.4 FROM FIXED NUMERICAL INSTANCES TO AN ANALYTIC FORMULA

These routines analyze a specified numerical instance. They do not automatically align bases across iteration indices or infer a closed-form, k-dependent pattern. In the `Peppy` workflow, the agent repeats the basis and reconstruction calculations for several values of k and N, compares the tagged vectors and the entries of C, and conjectures their analytic formulas. The final stage then verifies the resulting Lyapunov identity, boundary relations, and coefficient signs symbolically. Keeping this division explicit is important: the module supplies inspectable numerical evidence, while the proof establishes the identity for arbitrary indices and horizons.

C.2.5 AUXILIARY LDL DECOMPOSITION

`ldl_decompose_with_reversed_basis` is a specialized helper for a named slack matrix whose basis is ordered chronologically. It reverses the basis, performs an LDL decomposition, propagates any factorization permutation to the vector names, and can print the labeled factors. This is useful for inspecting certain slack-matrix decompositions, but it is an auxiliary diagnostic rather than the main basis-recovery mechanism used by the current Lyapunov workflow.

C.2.6 ROLE IN THE PEPPY ARCHITECTURE

The module's column-space, basis-selection, and coefficient-reconstruction routines alongside `PEPFlow`'s Gram representation and vector tags are crucial components of Stage 4 of `Peppy`. Without these components, the matrix of the quadratic part of V_k is opaque, and identifying a compact mathematical representation requires case-specific coordinate calculations. Together they let the agent test proposed vectors against this matrix, retain a basis that a human can read, and recover the small matrix that must later be expressed and verified symbolically.

The computations remain subject to numerical tolerances and to the candidate vectors proposed by the agent. In particular, the sparse-basis routine performs an exhaustive search within the supplied candidate family and warns when the number of subsets becomes large. It neither guarantees a globally preferred mathematical basis nor proves the inferred formulas. Its purpose is to make the transition from the matrix of the quadratic part to a readable candidate Lyapunov function systematic and understandable.

D ADDITIONAL DETAILS FOR SECTION 3

This appendix documents the commands used in `Peppy` to turn an algorithm description into an analytic Lyapunov analysis. Each block specifies a mathematical object to construct, checks on that object, and the evidence passed to the next block.

D.1 MODULAR EXECUTION AND STATE

The five commands are executed sequentially, with recovery loops when needed:

`/pep-implement` → `/pep-full-proof` → `/lyap-define` → `/lyap-vectors` → `/lyap-closed-form`.

The Codex plugin provides skill entry points and accompanying workflow instructions under `plugins/pepflow-workflows/skills/`. For Claude, each command is written in a Markdown file in the directory `.claude/commands`.

The first command, `/pep-implement`, starts by parsing the information provided by the user into the fields listed below. It creates the executable setup and initializes the Lyapunov notebook. Each subsequent command typically reads the state generated by the previous block, performs a specific discovery or verification task, writes a new state file, and updates the same Lyapunov notebook.

Step 1 — Parse

From `$ARGUMENTS`, identify:

Field	Description
`ALGO_NAME`	`snake_case` identifier, e.g. `heavy_ball`
`PROBLEM_TYPE`	`smooth_convex` / `smooth_strongly_convex` / `monotone_operator` / `composite`
`OBJ_TAG`	`"f"` (functions) or `"A"` (operators)
`PARAMS`	All symbolic parameters (e.g. `L`, `alpha`, `beta`, `R`)
`PERFORMANCE_METRIC`	e.g. `f(x_N) - f(x_star)` or `||A(x_N)||²`
`INITIAL_CONDITION`	e.g. `||x_0 - x_star||²<= R²`
`CONJECTURED_RATE`	Known bound or `"unknown"`

The following table briefly summarizes the tasks performed by each command and their main outputs, together with the checks before completion. Numerical checks apply to sampled instances. Block 5 establishes the analytic Lyapunov proof.

Block / command	Tasks	Main output	Checks before completion
B1 `/pep-implement`	Parse the algorithm information. Implement the algorithm. Solve PEP problem for various $N = 1, \ldots, 7$ and identify a conjectured convergence rate. Initialize the Lyapunov notebook.	Setup module. Numerical sweep data. Block 1 state `{ALGO_NAME}_b1.json`.	Check problem encoding and successful solves. Label the candidate rate as numerical evidence.

Continued on the next page

Block / command	Tasks	Main output	Checks before completion
B2 `/pep-full-proof`	Find sparsity patterns of the dual variables. Find a decomposition of the PSD matrix corresponding to the sum-of-squares term in (2). Try to find closed-form expressions for the dual variables and the PSD matrix.	Closed-form λ and S, when available. Verified numerical PEP certificate. Separate numerical and analytic verification statuses. Block 2 state `{ALGO_NAME}_b2.json`.	Check the full numerical identity, dual signs, PSD slack, decomposition, and dense/relaxed bound consistency. Record the initial formula search.
B3 `/lyap-define`	Build Lyapunov partial sums V_k from the full-PEP certificate. Compute the rank profile of the matrices representing the quadratic parts of $V_0, \ldots, V_N$.	Partial sums V_k, stored in the list `lyap`. Rank profile `[r_0, r_1, ..., r_N]`. Rank evidence across horizons. Block 3 state `{ALGO_NAME}_b3.json`.	Check decomposition, coverage, numerical step identities, and required signs. Confirm consistent interior rank at two or more horizons with fixed boundary treatment.
B4 `/lyap-vectors`	Find special vectors that yield a summation-free expression of V_k. Find basis patterns: special vector patterns indexed by k. Extract and verify coefficient matrices. Optionally infer closed-form coefficient patterns.	Basis patterns. Coefficient matrices and available formulas or explicit unknowns. Block 4 state `{ALGO_NAME}_b4.json`.	Check independence and reconstruction of quadratic, function value, and constant terms at every index, including exceptions.
B5 `/lyap-closed-form`	Derive unresolved coefficients and assemble the final closed-form Lyapunov proof. Verify the result symbolically.	Final notebook. Block 5 state `{ALGO_NAME}_b5.json`. LaTeX theorem/proof.	Establish exact step, initial, and terminal identities and sign/domain conditions. Verify that the proof yields the stated convergence bound. Complete notebook delivery checks.

D.2 Discovery and recovery using structure

D.2.1 From numerical certificates to analytic coefficients

After Block 2's initial formula search, verified numerical certificates can support Blocks 3 and 4 while global dual formulas remain unresolved. All components of each certificate come from the same solve. Replacing a certificate requires rebuilding and rechecking dependent partial sums, bases, and coefficients. Block 4 passes a reconstructed expression structure, including function value and constant terms and any unknown coefficients, to Block 5. Block 5 derives missing coefficients and local certificate terms from symbolic identities, recurrences, and endpoint conditions.

D.2.2 Structural search using rank

Block 3 uses rank profiles to guide groupings and square assignments. Confirmation requires the same rank at two or more horizons under one construction recipe, fixed boundary offsets, and the same tolerance and scaling conventions. Each confirming horizon retains at least three indices, excluding at most two per end, counting V_0 and V_N. Excluded terms remain subject to coverage, reconstruction, and proof checks. Equal ranks alone establish neither a common basis nor a theorem for arbitrary horizons.

If a candidate is unhelpful, the agent tries alternative groupings, square assignments, or verified decompositions of the same slack matrix, or obtains a new certificate at a chosen horizon from

Block 2. Block 4 checks independence and full reconstruction. If these checks fail, the search returns to Block 3.

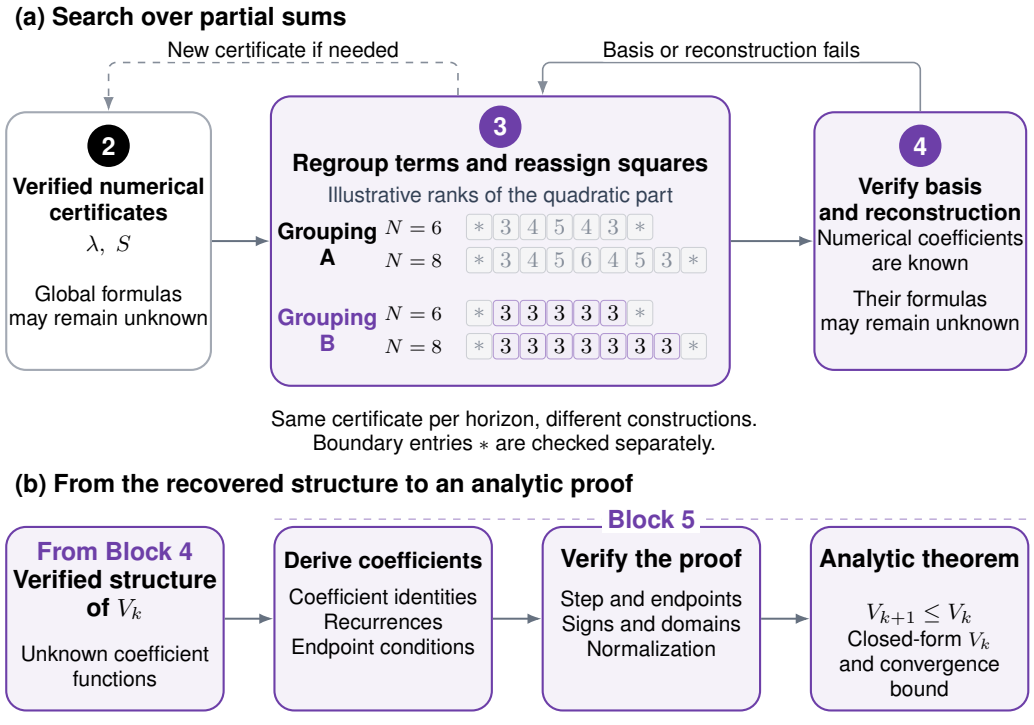


Figure 3: Structural discovery and analytic completion. (a) The same certificate at each horizon supports different groupings and square assignments. The displayed ranks are illustrative, and coverage and exceptional indices still require verification. (b) Verified numerical structure can pass to Block 5 with unresolved coefficient formulas, which are derived and verified before reporting the analytic proof.

D.3 Verification and completion criteria

D.3.1 Numerical evidence and analytic proof

The workflow separately records verification of numerical certificates, analytic full-PEP proofs, and Lyapunov proofs. Numerical agreement alone does not verify either analytic proof. An explicitly requested full-PEP proof remains outstanding until verified.

For analytic completion, Block 5 establishes the step, initial and terminal identities over their claimed index and parameter domains, checking inner product coefficients, function value coefficients, and constant offsets. The proof also justifies required signs and nonvanishing denominators, treats exceptional indices and small horizons, and checks the requested bound's normalization. Together, these arguments establish $V_{k+1} \le V_k$ and the endpoint implications needed for the convergence bound. A zero identity residual establishes equality. Sign and domain arguments establish its inequality consequence.

D.3.2 Inspectable proof artifacts and delivery

The final result is a Jupyter notebook file that contains a LaTeX theorem statement and proof outline, together with the detailed results of the whole workflow. Its contents are organized as follows.

Notebook section	Contents
Problem and result	Algorithm and assumptions, followed by the theorem and proof outline.
Executable setup	Imports, function or operator definitions, and PEP construction.
Numerical evidence	Rate comparison, dense and relaxed solves, and certificate checks.
Lyapunov construction	Partial sums, rank profiles, and basis search.
Analytic verification	Coefficient reconstruction, closed-form V_k, and symbolic step, initial, and terminal identities.

Definitions needed for the proof and executed symbolic checks remain visible in its cells. Mathematical proof status is recorded separately from delivery completion, which additionally requires static checks, successful notebook execution, inspection of rendered mathematics, and an evidence checklist.

D.4 Artifacts and human interaction

Block 1 creates the output file

`examples_peppy/{ALGO_NAME}/state/{ALGO_NAME}_b1.json`,

and initializes a Jupyter notebook

`examples_peppy/{ALGO_NAME}/{ALGO_NAME}_example_lyap.ipynb`.

The setup module resides beside the notebook. `state/` holds block states and certificates. Additional helpers, exploratory evidence, and logs belong in `supplementary/`. Runtime scratch files are kept separately. Paths follow the actual notebook destination.

As additional guidance, `supplementary/` may accumulate files that are unnecessary for distribution, so we used the auxiliary skill `clean-peppy-artifacts` to clean up the released code. Simple proofs can be read directly from the Jupyter notebook, while more involved proofs can be easier to read as a LaTeX document and PDF generated from the notebook using the optional `pdf-from-ipynb` skill.

The agent first performs the prescribed recovery. Unresolved closed forms alone do not require human intervention. In an interactive run, if the search still does not identify partial sums with consistent interior ranks, users may suggest alternative groupings or square decompositions. The resulting construction remains subject to the same reconstruction and verification checks. The code and accompanying examples are included in the supplementary material. Examples of the theorem statements and proofs are provided in Section E.

E Omitted details of Section 4

We provide experimental details omitted from the main text, followed by the input prompts and generated proofs. For each condition, the output boxes reproduce the complete theorem and proof outline Markdown cells from repetition 1, without editorial correction. References to other notebook content refer to the original Jupyter notebook. Only typesetting and equation line breaks are changed. All 30 attempts contribute to the reported results.

E.1 Experimental setup and evaluation

E.1.1 Execution settings

We ran five repetitions for each of six conditions: PGM with and without supplied SOS, OGM, BPPM, FEG, and Dual-FEG. Generation and independent evaluation both used GPT-6 Astra with medium reasoning. Up to three AI agents (workers) generated proofs concurrently, with each repetition completed before the next began. Each generation attempt and evaluation had a limit of two hours and no token cap. The experiment was launched with the following request:

```
Use $evaluate-peppy with GPT-6 Astra and medium reasoning for both
generation and evaluation.
Run all six conditions with 5 repetitions per condition, in parallel.
```

The recorded environment used Windows x86-64 and Python 3.11.13, with NumPy 2.3.1, SymPy 1.14.0, and CVXPY 1.7.1. Numerical solves used Clarabel 0.11.1 with one numerical thread per worker. Workers were assigned separate Python/Jupyter environments.

E.1.2 CLEAN-ROOM SETTING

Here, a clean-room run starts in a fresh context with the selected problem packet, approved generic framework code and workflow instructions, and artifacts created within that run. Workers are instructed not to consult existing target proofs, other repetitions, or other workers' results. BPPM receives its conjectured rate, and PGM with supplied SOS receives a candidate SOS decomposition to verify. The full inputs below specify this assistance. The following excerpt from the worker instructions states the source restriction:

```
Do not inspect, search for, open, quote, copy, or derive hints from
any target-specific pre-existing result in this repository, another
checkout, external sources, or any form of history.
```

The clean-room policy relies on instructions, without filesystem isolation or control over prior model knowledge. Evaluators check each proof in a separate context without other attempts' results.

E.1.3 SUCCESS CRITERIA

A successful proof establishes the prescribed bound for finite horizons, including its constant, under the input assumptions for every integer $N \geq 1$. It need not use a particular Lyapunov function or SOS decomposition. The evaluation checks the following evidence from each block:

Block	Required checks
B1	Correct algorithm, assumptions, metric, and normalization. Valid numerical sweep and rate comparison.
B2	Coherent full-PEP certificate, agreement of dense and relaxed bounds, identities for all scalar components, required dual signs and PSD conditions, and reconstruction of the slack matrix.
B3	Decomposition into partial sums, coverage and increment identities, and consistent interior rank at two or more horizons with fixed boundary treatment.
B4	Independent spanning bases and reconstruction of the full Lyapunov function, including function value terms, constants, and exceptional indices.
B5	General step, initial and terminal identities, coefficient signs and domains, exceptional cases, and implication of the exact target bound.

Finite numerical agreement alone does not establish the theorem. An invalid identity or an unjustified necessary inequality prevents mathematical success. Missing verification and runtime errors are recorded separately. Workflow completion additionally requires the notebook evidence, static checks, and execution from a clean kernel.

E.1.4 COMPLETION OF INTERRUPTED VERIFICATION

Three runs were flagged during final notebook execution in Block 5 for Python/Jupyter environment issues: BPPM repetition 2, FEG repetition 3, and PGM without SOS repetition 2. Each run had already produced its proof. After the experiment, the experimenter requested completion of the omitted verification, treating these generic runtime issues separately from mathematical correctness. The relevant excerpt from that request is:

```
Do not rerun proof generation, alter submitted proofs, or supply
missing mathematical arguments. Verification code may check the
submitted expressions and reasoning.

Treat generic dependency-path and notebook-display issues as
operational issues.
```

This post hoc evaluation retained 28 completed verifications, including BPPM repetition 2, and completed the two remaining evaluations. The evaluators replayed selected submitted proof cells and saved certificates in validation copies, with recorded adjustments confined to execution. No proofs were regenerated or repaired. The mathematical criteria and tolerances were unchanged. All 30 submitted proofs passed, and 27 runs completed the original workflow. No prohibited mathematical exposure was identified in the available records, although the access logs were not exhaustive.

E.2 Computation and observed outputs

E.2.1 Time and token measurements

Table 2 reports means and sample standard deviations over all five attempts per condition, including time consumed by the three incomplete deliveries. Generation time runs from dispatch to stopping and includes service waits and recovery. Evaluation time is separate. Token counts are input plus output, including reused context, with cached input and reasoning counted within those totals. Across all attempts, generation used 24.51 summed hours and 378.47 million tokens. The original evaluations used 2.87 summed hours and 43.67 million tokens. Summed worker times are not elapsed experiment time because workers ran in parallel.

The two additional evaluations used 8.23 minutes and 1.75 million tokens for PGM without SOS repetition 2, and 6.68 minutes and 1.40 million tokens for FEG repetition 3. These are evaluation costs, accounted for separately from generation and excluding orchestration.

E.2.2 Variation in the PGM outputs

Overall generation times were similar: 57.9 ± 4.2 minutes with supplied SOS and 60.4 ± 9.7 minutes without it (mean $\pm$ sample standard deviation). The five supplied-SOS runs produced the same Lyapunov function after notation and normalization were aligned. Without supplied SOS, the five runs produced three distinct forms: one in repetition 1, a shared form in repetitions 2–4, and a third in repetition 5. These differences persist beyond notation or overall scaling, while every proof establishes the same target bound. The outputs from repetitions 1 and 5 below illustrate this variation.

E.3 PGM

The two input conditions share the algorithm and assumptions. The second additionally supplies a candidate SOS decomposition. The archived packets below make this difference explicit.

E.3.1 Input prompt: PGM without supplied SOS

```
# Common assumptions

All cases use real finite-dimensional Euclidean spaces, positive `L`
where present, nonnegative `R`, and integer horizons `N >= 1`.
Proximal updates and stated evaluations are assumed well-defined.
Supply these common assumptions and only the selected case subsection
to its worker.


### `pgm`
```

```
- Problem: composite convex minimization. `f` is convex and
`L`-smooth; `g` is closed, proper, and convex. Write `h = f + g`.
- Initial condition: `x_star` minimizes `h`, and `||x_0 - x_star||^2
<= R^2`.
- Metric: `h(x_N) - h(x_star)`.
- For `k = 0, ..., N-1`, set `x_{k+1} = prox_{g/L}(x_k - grad
f(x_k)/L)`.
- Conjectured rate: unknown.

### `pgm_without_sos`


- No fixed SOS aid is supplied. Derive the certificate and Lyapunov
analysis within this attempt. You may discover and use an SOS
decomposition yourself.
- Do not read the supplied fixed aid, its source, either condition's
pre-existing answers, sibling outputs, or any other attempt's
artifacts, including the paired repetition.
```

E.3.2 INPUT PROMPT: PGM WITH SUPPLIED SOS

```
# Common assumptions

All cases use real finite-dimensional Euclidean spaces, positive `L`
where present, nonnegative `R`, and integer horizons `N >= 1`.
Proximal updates and stated evaluations are assumed well-defined.
Supply these common assumptions and only the selected case subsection
to its worker.


### `pgm`


- Problem: composite convex minimization. `f` is convex and
`L`-smooth; `g` is closed, proper, and convex. Write `h = f + g`.
- Initial condition: `x_star` minimizes `h`, and `||x_0 - x_star||^2
<= R^2`.
- Metric: `h(x_N) - h(x_star)`.
- For `k = 0, ..., N-1`, set `x_{k+1} = prox_{g/L}(x_k - grad
f(x_k)/L)`.
- Conjectured rate: unknown.

### `pgm_with_sos`


- Authorized assistance: the fixed SOS aid below, supplied from the
start. Record it as supplied assistance. Derive the remaining
certificate and Lyapunov analysis within this attempt.
- The aid is a candidate, not a verified certificate or proof. Verify
it against this attempt's certificate and all applicable block gates.

## Fixed PGM SOS Aid


Only a `pgm_with_sos` worker receives this section. This is a
candidate decomposition of the nonnegative full-PEP slack. It does not
provide the convergence bound, dual multipliers, partial sums, or
final Lyapunov function.

For `i = 1, ..., N`, write `a_i = grad f(x_i)`, `a_star = grad
f(x_star)`, and choose the proximal subgradient `s_i in partial
g(x_i)` given by `s_i = L*(x_{i-1}-x_i) - a_{i-1}`. Define:
```

```
'''text
c1_i = N / ((2*N+1-i)*(2*N-i)*L)
u1_i = (i/(2*N))*a_i + ((2*N-i)/(2*N))*a_{i-1} - a_star

c2_i = (((i-1)/(2*N-i))*((2*N+2)/(2*N+1))
        + (2*N)/((2*N-i)^2*(2*N+1))) / (2*L)
u2_i = a_{i-1} + s_i - (L/(2*N+1-i))*(x_{i-1}-x_star)

c3_i = (i/(2*N+1-i))*((2*N)/(2*N+1))/(2*L)
u3_i = ((2*N+1)/(2*N))*(a_i+s_i) - (a_{i-1}+s_i)/(2*N)
        - (L/(2*N-i))*(x_i-x_star)

S_candidate = sum(c1_i*||u1_i||^2 + c2_i*||u2_i||^2 + c3_i*||u3_i||^2
                  for i = 1, ..., N)
'''

These are mathematical formulas, not directly executable Python.
Define 'a_0 = grad f(x_0)'. Use exact arithmetic when implementing
them. Verify their role against the certificate obtained in this
attempt, accounting explicitly for any overall certificate
normalization.
```

E.3.3 GENERATED OUTPUT: PGM WITHOUT SUPPLIED SOS (REPETITION 1)

Theorem
For the stated PGM recurrence with convex L-smooth f, closed proper convex g, $L > 0$, $R \geq 0$, integer $N \geq 1$, a minimizer $x_\star$ and $\|x_0 - x_\star\|^2 \leq R^2$, the following potential is nonincreasing. All stated evaluations and proximal steps are well-defined.
Put $d_k = x_k - x_\star$ and $p_k = (\nabla f(x_k) - \nabla f(x_\star))/L$. For $1 \leq k < N$, define

$$V_k = -\frac{L}{4N}\|d_0\|^2 + \frac{L(N-k)}{(2N-k)^2}\|d_k\|^2 - \frac{\langle d_k, \nabla f(x_k)\rangle}{2N-k} - \frac{Lk}{2(2N-k)^2}\|p_k\|^2 + \frac{k+1}{2N-k}(f(x_k) - f(x_\star)) + \frac{k}{2N-k}(g(x_k) - g(x_\star)).$$

The endpoints are $V_0 = 0$ and

$$V_N = h(x_N) - h(x_\star) - \frac{L}{4N}\|d_0\|^2.$$

Consequently,

$$h(x_N) - h(x_\star) \leq \frac{LR^2}{4N}.$$

Proof outline
Put $d_k = x_k - x_\star$ and $p_k = (\nabla f(x_k) - \nabla f(x_\star))/L$. For $1 \leq k < N$, define

$$V_k = -\frac{L}{4N}\|d_0\|^2 + \frac{L(N-k)}{(2N-k)^2}\|d_k\|^2 - \frac{\langle d_k, \nabla f(x_k)\rangle}{2N-k} - \frac{Lk}{2(2N-k)^2}\|p_k\|^2 + \frac{k+1}{2N-k}(f(x_k) - f(x_\star)) + \frac{k}{2N-k}(g(x_k) - g(x_\star)).$$

The endpoints are $V_0 = 0$ and

$$V_N = h(x_N) - h(x_\star) - \frac{L}{4N}\|d_0\|^2.$$

Define the nonpositive interpolation residuals

$$I_f(u,v) = f(v) - f(u) + \langle \nabla f(v), u - v\rangle + \frac{\|\nabla f(u) - \nabla f(v)\|^2}{2L},$$
$$I_g(u,v) = g(v) - g(u) + \langle s_v, u - v\rangle,$$

where $s_v \in \partial g(v)$ is the selected oracle subgradient. At the minimizer choose $s_\star = -\nabla f(x_\star)$; at a new iterate use the proximal subgradient. Write $a_k = (k+1)/(2N-k)$ and $b_k = k/(2N-k)$. For a step,

put $m = 2N - k$, $p = p_k$, $q = p_{k+1}$, $s = (s_{x_{k+1}} + \nabla f(x_\star))/L$, $z = s - d_k/m$. The recurrence gives $d_{k+1} = d_k - p - s$. Define

$$A_k = \frac{k+1}{2(m-1)^2} + \frac{k+2}{2(m-1)}, \quad B_k = \frac{k+1}{2(m-1)}, \quad C_k = \frac{km+N-k}{(m-1)^2},$$

$$D_k = \frac{k+1}{2m(m-1)}, \quad E_k = \frac{km+1}{2(m-1)^2}, \quad F_k = \frac{a_k}{2} - \frac{N-k-1}{(m-1)^2} - \frac{k}{2m^2}.$$

Set $\alpha_k = A_k$, $\beta_k = C_k - B_k^2/A_k$, $\eta_k = E_k - B_k D_k/A_k$, $\gamma_k = F_k - D_k^2/A_k - \eta_k^2/\beta_k$ and

$$Q_k = L\alpha_k \|q + (B_k/A_k)z + (D_k/A_k)p\|^2 + L\beta_k \|z + (\eta_k/\beta_k)p\|^2 + L\gamma_k \|p\|^2.$$

The sign lemma below proves $\alpha_k, \beta_k > 0$ and $\gamma_k \geq 0$ for $0 \leq k \leq N-2$. For $N \geq 2$, the exact base identity is

$$V_1 = a_0 I_f(x_\star, x_0) + (a_1 - a_0) I_f(x_\star, x_1) + b_1 I_g(x_\star, x_1) + a_0 I_f(x_0, x_1) - \frac{L}{4N}\|p_0\|^2 - Q_0.$$

For $1 \leq k \leq N-2$, the exact recursion is

$$V_{k+1} - V_k = (a_{k+1} - a_k) I_f(x_\star, x_{k+1}) + (b_{k+1} - b_k) I_g(x_\star, x_{k+1}) + a_k I_f(x_k, x_{k+1}) + b_k I_g(x_k, x_{k+1}) - Q_k.$$

At the final step put $p = p_{N-1}$, $q = p_N$, $s = (s_{x_N} + \nabla f(x_\star))/L$, $d = d_{N-1}$ and

$$T_N = \frac{L}{2}\left\|q + s + \frac{p-d}{N+1}\right\|^2 + \frac{L}{2}\left\|s + \frac{Np-d}{N+1}\right\|^2.$$

Then, for $N \geq 2$,

$$V_N - V_{N-1} = (1 - a_{N-1}) I_f(x_\star, x_N) + (1 - b_{N-1}) I_g(x_\star, x_N) + a_{N-1} I_f(x_{N-1}, x_N) + b_{N-1} I_g(x_{N-1}, x_N) - T_N.$$

For $N = 1$, use the separate identity

$$V_1 = \tfrac{1}{2} I_f(x_\star, x_0) + \tfrac{1}{2} I_f(x_\star, x_1) + I_g(x_\star, x_1) + \tfrac{1}{2} I_f(x_0, x_1) - \tfrac{L}{4}\|p_0\|^2 - T_1.$$

All residual multipliers are nonnegative: $a_{k+1} - a_k = (2N+1)/[(2N-k-1)(2N-k)]$, $b_{k+1} - b_k = 2N/[(2N-k-1)(2N-k)]$, $1 - a_{N-1} = 1/(N+1)$ and $1 - b_{N-1} = 2/(N+1)$. Thus $V_N \leq \cdots \leq V_1 \leq V_0 = 0$, and

$$h(x_N) - h(x_\star) \leq \frac{L}{4N}\|d_0\|^2 \leq \frac{LR^2}{4N}.$$

E.3.4 Generated output: PGM with supplied SOS (repetition 1)

Theorem
Let $h = f + g$, with f convex and differentiable with L-Lipschitz gradient, $L > 0$, and g closed, proper and convex. Suppose $x_\star$ minimizes h and $\|x_0 - x_\star\|^2 \leq R^2$, $R \geq 0$. For any integer $N \geq 1$, proximal gradient with step $1/L$ satisfies

$$h(x_N) - h(x_\star) \leq \frac{L\|x_0 - x_\star\|^2}{4N} \leq \frac{LR^2}{4N}.$$

Here is a nonincreasing Lyapunov certificate. Write $a_k = \nabla f(x_k)$, $a_\star = \nabla f(x_\star)$, $d_k = x_k - x_\star$ and $D_k = 2N - k$. Define $V_0 = 0$ and

$$V_k = \frac{k+1}{D_k}[f(x_k) - f(x_\star)] + \frac{k}{D_k}[g(x_k) - g(x_\star)] - \frac{\langle a_k, d_k\rangle}{D_k} + \frac{\|a_k - a_\star\|^2}{2LD_k} + \frac{L(N-k)}{D_k^2}\|d_k\|^2 - \frac{L}{4N}\|d_0\|^2, \qquad 1 \leq k < N.$$

The terminal case is defined separately:

$$V_N = h(x_N) - h(x_\star) - \frac{L}{4N}\|d_0\|^2.$$

Then $V_N \le V_{N-1} \le \cdots \le V_0 = 0$. The statement also covers $N = 1$ and $R = 0$; it requires no finite value of $g(x_0)$.

Proof outline

Use $a_k = \nabla f(x_k)$, $s_i = L(x_{i-1} - x_i) - a_{i-1} \in \partial g(x_i)$, $s_\star = -a_\star$, and $d_k = x_k - x_\star$. The interpolation residuals are

$$I_f(u, v) = f(v) - f(u) + \langle \nabla f(v), u - v\rangle + \frac{\|\nabla f(u) - \nabla f(v)\|^2}{2L} \le 0,$$

$$I_g(u, v) = g(v) - g(u) + \langle s_v, u - v\rangle \le 0.$$

Define $D_k = 2N - k$, $V_0 = 0$, $V_N = h(x_N) - h(x_\star) - L\|d_0\|^2/(4N)$, and, for $1 \le k < N$,

$$V_k = \frac{(k+1)[f(x_k) - f(x_\star)] + k[g(x_k) - g(x_\star)] - \langle a_k, d_k\rangle}{D_k} + \frac{\|a_k - a_\star\|^2}{2LD_k} + \frac{L(N-k)\|d_k\|^2}{D_k^2} - \frac{L\|d_0\|^2}{4N}.$$

For $1 \le i \le N$, set $Q_i = \sum_{r=1}^{3} c_{i,r}\|u_{i,r}\|^2$, where

$$c_{i,1} = \frac{N}{(2N+1-i)(2N-i)L}, \qquad u_{i,1} = \frac{i}{2N}a_i + \frac{2N-i}{2N}a_{i-1} - a_\star,$$

$$c_{i,2} = \frac{1}{2L}\left[\frac{i-1}{2N-i}\frac{2N+2}{2N+1} + \frac{2N}{(2N-i)^2(2N+1)}\right],$$

$$u_{i,2} = a_{i-1} + s_i - \frac{L}{2N+1-i}d_{i-1},$$

$$c_{i,3} = \frac{i}{2N+1-i}\frac{2N}{2N+1}\frac{1}{2L},$$

$$u_{i,3} = \frac{2N+1}{2N}(a_i + s_i) - \frac{a_{i-1} + s_i}{2N} - \frac{L}{2N-i}d_i.$$

Let

$$\alpha_i = \begin{cases} \dfrac{2N+1}{(2N-i)(2N+1-i)} & i < N, \\ \dfrac{1}{N+1} & i = N, \end{cases} \qquad \beta_i = \begin{cases} \dfrac{2N}{(2N-i)(2N+1-i)} & i < N, \\ \dfrac{2}{N+1} & i = N. \end{cases}$$

The exact identity for every step is

$$V_i - V_{i-1} = \mathbf{1}_{\{i=1\}}\frac{I_f(x_\star, x_0)}{2N} + \alpha_i I_f(x_\star, x_i) + \frac{i}{2N+1-i}I_f(x_{i-1}, x_i) + \beta_i I_g(x_\star, x_i) + \frac{i-1}{2N+1-i}I_g(x_{i-1}, x_i) - Q_i.$$

The last I_g term is **omitted** for $i = 1$, avoiding an evaluation of $g(x_0)$. For $N = 1$, use the terminal multipliers and the initial residual together. The notebook verifies the interior, base, terminal, and $N = 1$ identities separately, in all scalar components with symbolic parameters.

All multipliers and square coefficients are nonnegative: $2N - i \ge N > 0$, $2N + 1 - i \ge N + 1 > 0$, $L > 0$, and $i - 1 \ge 0$. Thus each displayed right-hand side is nonpositive. Telescoping yields $V_N \le V_0 = 0$, so $h(x_N) - h(x_\star) \le L\|d_0\|^2/(4N) \le LR^2/(4N)$. The use of a supplied SOS candidate is justified by these exact identities, not by numerical agreement.

E.3.5 Additional generated output: PGM without supplied SOS (repetition 5)

This additional example is selected to illustrate the distinct Lyapunov function using vectors from the next step. The outputs from repetition 1 above retain the uniform selection rule.

Theorem

Let f be convex and L-smooth, g closed proper convex, $L > 0$, $R \ge 0$, and $x_\star \in \arg\min(f + g)$. Assume the stated proximal updates are well-defined and $\|x_0 - x_\star\|^2 \le R^2$. For every integer $N \ge 1$, PGM satisfies

$$h(x_N) - h(x_\star) \le \frac{LR^2}{4N}.$$

Here is an explicit finite-product Lyapunov certificate. Let $s_i \in \partial g(x_i)$ be the proximal subgradient for $i \geq 1$, and set

$$d_i = x_i - x_\star, \quad p_i = \frac{\nabla f(x_i) - \nabla f(x_\star)}{L}, \quad q_i = \frac{s_i + \nabla f(x_\star)}{L},$$
$$F_i = \frac{f(x_i) - f(x_\star) - \langle \nabla f(x_\star), d_i \rangle}{L},$$
$$G_i = \frac{g(x_i) - g(x_\star) + \langle \nabla f(x_\star), d_i \rangle}{L}.$$

Only F_0, not G_0 or s_0, is used. We have $d_{i+1} = d_i - p_i - q_{i+1}$.
For $0 \leq j \leq N-2$, write $m_j = 2N - j$ and define the explicit matrices

$$T_j = \begin{pmatrix} 2(j+1)(m_j-1)[(j+2)m_j(m_j-1) - (j+1)] & (j+1)^2(m_j-2)(2N+1) \\ 4m_j^2(m_j-1)[(2j+3)m_j - (3j+4)] & 2m_j^2(m_j-2)(j+1)(j+2) \end{pmatrix}.$$

Define z_k by the finite ordered product

$$\begin{pmatrix} P_k \\ Q_k \end{pmatrix} = T_k T_{k+1} \cdots T_{N-2} \begin{pmatrix} N \\ 2(N+1)^2 \end{pmatrix}, \qquad z_k = \frac{P_k}{Q_k}, \qquad 0 \leq k \leq N-1.$$

The empty product is the identity, so $z_{N-1} = N/[2(N+1)^2]$. This formula has no recursively unspecified coefficients.
For $1 \leq k \leq N-1$, let $m = 2N - k$ and

$$a_k = \frac{k+1}{m}, \quad b_k = \frac{k}{m}, \quad A_k = \frac{N - k - 1 + (k+1)/(2m) - z_k}{(m-1)^2},$$
$$B_k = \frac{1}{2m} + \frac{z_k}{m-1}, \quad C_k = -\frac{k-1}{2m} - z_k.$$

Then the nonincreasing sequence has the full formula

$$V_k = L\big[-\frac{\|d_0\|^2}{4N} + a_k F_k + b_k G_k - \frac{a_k}{2}\|p_k\|^2 - b_k\langle p_k, q_{k+1}\rangle + A_k\|d_{k+1}\|^2 + 2B_k\langle d_{k+1}, q_{k+1}\rangle + C_k\|q_{k+1}\|^2\big],$$
$$V_0 = 0, \qquad V_N = h(x_N) - h(x_\star) - \frac{L}{4N}\|d_0\|^2.$$

There are no interior indices when $N = 1$. The certificate uses one-step-ahead vectors, as in the numerically discovered Block 4 basis.

Proof outline

Use the coefficient definitions and finite product in the theorem. The potential being proved is, explicitly,

$$\frac{V_k}{L} = -\frac{\|d_0\|^2}{4N} + a_k F_k + b_k G_k - \frac{a_k}{2}\|p_k\|^2 - b_k\langle p_k, q_{k+1}\rangle + A_k\|d_{k+1}\|^2 + 2B_k\langle d_{k+1}, q_{k+1}\rangle + C_k\|q_{k+1}\|^2 \quad (1 \leq k < N),$$

with $V_0 = 0$ and $V_N = h(x_N) - h(x_\star) - L\|d_0\|^2/(4N)$.
The nonpositive interpolation residuals, normalized by L, are

$$I_f(i,j) = \frac{f(x_j) - f(x_i) + \langle \nabla f(x_j), x_i - x_j\rangle + \|\nabla f(x_j) - \nabla f(x_i)\|^2/(2L)}{L},$$
$$I_g(i,j) = \frac{g(x_j) - g(x_i) + \langle s_j, x_i - x_j\rangle}{L}.$$

The star index means $x_\star$ and $s_\star = -\nabla f(x_\star)$. Smooth convex interpolation gives $I_f \leq 0$, and convexity gives $I_g \leq 0$.
For $0 \leq k \leq N-2$, put $m = 2N - k$, $w = z_{k+1}$, and

$$H_k = 2(m-1)^2 w + (k+1)(m-2),$$
$$J_k = 2(m-1)[(2k+3)m - (3k+4)]w + (m-2)(k+1)(k+2),$$
$$\alpha_k = \frac{H_k}{2(m-2)^2}, \quad \beta_k = \frac{J_k}{2(m-1)H_k},$$
$$\gamma_k = \frac{2(m-1)w + (k+1)(m-2)}{H_k}, \quad \delta_k = \frac{k+1}{2m\beta_k}.$$

Define the complete square contribution

$$\mathcal{Q}_k = \alpha_k \left\| q_{k+2} - \frac{d_{k+1}}{m-1} + \gamma_k p_{k+1} \right\|^2 + \beta_k \left\| p_{k+1} + \delta_k \left(q_{k+1} - \frac{d_{k+1}}{m-1} \right) \right\|^2 .$$

The matrix product gives exactly $z_k = (k+1)/(2m) - (k+1)^2/(4m^2\beta_k)$. Substitution of the PGM updates yields, for every $1 \le k \le N-2$,

$$\frac{V_{k+1} - V_k}{L} = a_k I_f(k, k+1) + (a_{k+1} - a_k) I_f(\star, k+1) + b_k I_g(k, k+1) + (b_{k+1} - b_k) I_g(\star, k+1) - \mathcal{Q}_k.$$

For $N \ge 2$, define

$$\mathcal{Q}_{\text{init}} = \frac{\|p_0\|^2}{4N} + \frac{z_0}{(2N-1)^2} \|d_0 - p_0 - 2Nq_1\|^2.$$

The exact base identity is

$$\frac{V_1}{L} = a_0 I_f(\star, 0) + a_0 I_f(0, 1) + (a_1 - a_0) I_f(\star, 1) + b_1 I_g(\star, 1) - \mathcal{Q}_0 - \mathcal{Q}_{\text{init}}.$$

The terminal identity is

$$\frac{V_N - V_{N-1}}{L} = \frac{N}{N+1} I_f(N-1, N) + \frac{1}{N+1} I_f(\star, N) + \frac{N-1}{N+1} I_g(N-1, N) + \frac{2}{N+1} I_g(\star, N) - \mathcal{Q}_{\text{term}},$$

$$\mathcal{Q}_{\text{term}} = \frac{1}{2} \left\| p_N + \frac{N}{N+1} q_N - \frac{d_N}{N+1} \right\|^2 .$$

For $N = 1$, the separate identity is

$$\frac{V_1}{L} = \tfrac{1}{2} I_f(\star, 0) + \tfrac{1}{2} I_f(0, 1) + \tfrac{1}{2} I_f(\star, 1) + I_g(\star, 1) - \tfrac{1}{4}\|p_0\|^2 - \tfrac{1}{8}\|d_0 - p_0 - 2q_1\|^2 - \tfrac{1}{2}\|p_1 + q_1/2 - d_1/2\|^2.$$

All four entries of T_j are strictly positive on $0 \le j \le N-2$: here $m_j \ge j+4$, so both bracketed factors are positive. The terminal vector has positive entries, hence every $z_k > 0$ and every product denominator is nonzero. Also $H_k, J_k, \alpha_k, \beta_k > 0$ and all displayed denominators are positive. The weights a_k, b_k are nonnegative, with

$$a_{k+1} - a_k = \frac{2N+1}{(2N-k)(2N-k-1)} > 0, \qquad b_{k+1} - b_k = \frac{2N}{(2N-k)(2N-k-1)} > 0.$$

Every right side is nonpositive. Thus $V_N \le V_{N-1} \le \cdots \le V_1 \le V_0 = 0$, including the separate $N = 1$ case. Rearranging the explicit terminal potential and using $L > 0$ and $\|d_0\|^2 \le R^2$ proves the theorem. The notebook below verifies every Gram coefficient, function-value coefficient, and offset in these identities exactly with symbolic N, k, L.

E.4 OGM

E.4.1 Input prompt

```
# Common assumptions

All cases use real finite-dimensional Euclidean spaces, positive `L`
where present, nonnegative `R`, and integer horizons `N >= 1`.
Proximal updates and stated evaluations are assumed well-defined.
Supply these common assumptions and only the selected case subsection
to its worker.
```

````
### `ogm`


- Problem: smooth convex minimization. `f` is convex and `L`-smooth.
- Initial condition: `grad f(x_star) = 0`, `||x_0 - x_star||^2 <=
R^2`, and `z_0 = x_0`.
- Metric: `f(x_N) - f(x_star)`.
- Define `theta_{-1} = 0`; for `0 <= k < N`, `theta_k = (1 + sqrt(1 +
4*theta_{k-1}^2))/2`; set `theta_N = (1 + sqrt(1 +
8*theta_{N-1}^2))/2`.
- For `k = 0, ..., N-1`:

```text
y_{k+1} = x_k - grad f(x_k)/L
z_{k+1} = z_k - (2*theta_k/L)*grad f(x_k)
x_{k+1} = (1 - 1/theta_{k+1})*y_{k+1} + z_{k+1}/theta_{k+1}
```

- Conjectured rate: unknown. Preserve the special terminal `theta_N`
recurrence.
````

E.4.2 GENERATED OUTPUT (REPETITION 1)

Theorem
Let f be convex and L-smooth on a finite-dimensional Euclidean space, with $L > 0$, $\nabla f(x_\star) = 0$, and $\|x_0 - x_\star\|^2 \le R^2$, where $R \ge 0$. For every integer $N \ge 1$, run the algorithm above with its special terminal recurrence. Write $g_k = \nabla f(x_k)$, $f_\star = f(x_\star)$, $T = \theta_N$, and $a = x_0 - x_\star$.
A nonincreasing certificate is $V_0 = 0$, with

$$V_k = \frac{2\theta_k^2}{T^2}(f(x_k) - f_\star) - \frac{\theta_k^2}{LT^2}\|g_k\|^2 + \frac{L}{2T^2}\left(\|z_{k+1} - x_\star\|^2 - \|a\|^2\right), \quad 1 \le k < N,$$

and

$$V_N = f(x_N) - f_\star + \frac{L}{2T^2}\left(\left\|z_N - x_\star - \frac{T}{L}g_N\right\|^2 - \|a\|^2\right).$$

The interior range is empty when $N = 1$. These expressions satisfy $V_N \le \cdots \le V_0 = 0$, and therefore

$$\boxed{f(x_N) - f(x_\star) \le \frac{LR^2}{2\theta_N^2}.}$$

Proof outline
For evaluated points u, v, write $g(u) = \nabla f(u)$ and define the nonpositive smooth convex interpolation residual

$$\mathcal{I}(u, v) = f(v) - f(u) + \langle g(v), u - v\rangle + \frac{1}{2L}\|g(u) - g(v)\|^2 \le 0.$$

Use $\mathcal{I}_{i,j} = \mathcal{I}(x_i, x_j)$ and $\mathcal{I}_{\star,j} = \mathcal{I}(x_\star, x_j)$.
To verify the displayed V_k, define auxiliary expressions

$$\Phi_k = 2\theta_k^2(f(x_k) - f_\star) - \frac{\theta_k^2}{L}\|g_k\|^2 + \frac{L}{2}\|z_{k+1} - x_\star\|^2, \quad 0 \le k < N,$$

$$\Phi_N = T^2(f(x_N) - f_\star) + \frac{L}{2}\left\|z_N - x_\star - \frac{T}{L}g_N\right\|^2, \qquad A = \frac{L}{2}\|a\|^2.$$

Thus $V_k = (\Phi_k - A)/T^2$ for $1 \le k \le N$, while $V_0 = 0$.
The exact base, ordinary-step, and terminal-step identities are

$$\Phi_0 - A = 2\mathcal{I}_{\star,0},$$

$$\Phi_{j+1} - \Phi_j = 2\theta_{j+1}\mathcal{I}_{\star,j+1} + 2\theta_j^2\mathcal{I}_{j,j+1}, \quad 0 \le j \le N - 2,$$

$$\Phi_N - \Phi_{N-1} = T\mathcal{I}_{\star,N} + 2\theta_{N-1}^2\mathcal{I}_{N-1,N}.$$

The ordinary identity uses $\theta_{j+1}^2 - \theta_{j+1} = \theta_j^2$; the terminal identity uses $T^2 - T = 2\theta_{N-1}^2$. These are the two different recurrences in the input.
In particular, for $N \geq 2$,

$$V_1 - V_0 = \frac{2\mathcal{I}_{\star,0} + 2\theta_1\mathcal{I}_{\star,1} + 2\mathcal{I}_{0,1}}{T^2};$$

for $1 \leq k \leq N-2$,

$$V_{k+1} - V_k = \frac{2\theta_{k+1}\mathcal{I}_{\star,k+1} + 2\theta_k^2\mathcal{I}_{k,k+1}}{T^2};$$

and, when $N \geq 2$,

$$V_N - V_{N-1} = \frac{T\mathcal{I}_{\star,N} + 2\theta_{N-1}^2\mathcal{I}_{N-1,N}}{T^2}.$$

For $N = 1$, $T = 2$ and the direct identity is

$$V_1 = \frac{2\mathcal{I}_{\star,0} + 2\mathcal{I}_{\star,1} + 2\mathcal{I}_{0,1}}{4}.$$

Every multiplier is nonnegative: $\theta_0 = 1$, the ordinary recurrence preserves $\theta_j \geq 1$, and the terminal recurrence gives $T \geq 2$. All divisions are valid since $L > 0$ and $T > 0$. Thus all displayed increments are nonpositive.
Finally, the terminal formula gives

$$f(x_N) - f_\star = V_N + \frac{L}{2T^2}\|a\|^2 - \frac{L}{2T^2}\left\|z_N - x_\star - \frac{T}{L}g_N\right\|^2 \leq \frac{LR^2}{2T^2}.$$

This uses the nonnegative terminal square and the initial radius bound, and includes $R = 0$. The notebook below verifies each identity in all symbolic scalar components. The proof is local and valid for arbitrary horizon; a separate global analytic multiplier-table verification is not claimed.

E.5 BPPM

E.5.1 Input prompt

```
# Common assumptions

All cases use real finite-dimensional Euclidean spaces, positive `L`
where present, nonnegative `R`, and integer horizons `N >= 1`.
Proximal updates and stated evaluations are assumed well-defined.
Supply these common assumptions and only the selected case subsection
to its worker.


### `bppm`


- Problem: Bregman proximal point minimization. `f` is closed, proper,
convex, and differentiable relative to the Bregman kernel `h`; `h` is
closed, proper, convex, and differentiable.
- Define `D_h(u,v) = h(u) - h(v) - <grad h(v), u-v>`.
- Parameters: fixed `alpha > 0` and `R >= 0`.
- Initial condition: `x_star` minimizes `f`, and `D_h(x_star, x_0) <=
R`. Here the radius parameter is `R`, not `R^2`.
- Metric: `f(x_N) - f(x_star)`.
- For `k = 0, ..., N-1`, `x_{k+1} = prox^h_{alpha*f}(x_k)`, meaning an
exact minimizer of `alpha*f(x) + D_h(x, x_k)`.
- PEPFlow interface: `f.bregman_prox(x_k, alpha, h)`.
- Conjectured rate provided in the input: `R/(alpha*N)`.
```

E.5.2 Generated output (repetition 1)

Theorem
Under the stated Bregman proximal point assumptions, let $\alpha > 0$, $R \geq 0$, $N \geq 1$ be an integer, and $D_h(x_\star, x_0) \leq R$. No strong convexity of h or Lipschitz gradient assumption is needed. Define

$$V_0 = 0, \qquad V_k = \frac{\alpha k\big(f(x_k) - f(x_\star)\big) + D_h(x_\star, x_k) - D_h(x_\star, x_0)}{\alpha N}, \quad 1 \leq k \leq N.$$

Then $V_{k+1} \leq V_k$ and

$$0 \leq f(x_N) - f(x_\star) \leq \frac{R}{\alpha N}.$$

The normalization is the supplied Bregman radius R, not R^2. The separate definition of V_0 does not require evaluating $f(x_0)$.
Proof outline
Write $I_q(u, v) = q(v) - q(u) + \langle \nabla q(v), u - v \rangle \leq 0$ for convex $q \in \{f, h\}$. Proximal optimality gives $\nabla h(x_{k+1}) = \nabla h(x_k) - \alpha \nabla f(x_{k+1})$.
Use $V_0 = 0$ and $V_k = k(f(x_k) - f(x_\star))/N + [D_h(x_\star, x_k) - D_h(x_\star, x_0)]/(\alpha N)$ for $k \geq 1$. The initial step is

$$V_1 - V_0 = \frac{1}{N} I_f(x_\star, x_1) + \frac{1}{\alpha N} I_h(x_1, x_0) \leq 0.$$

For $1 \leq k < N$, direct expansion and the proximal relation give the complete identity

$$\begin{aligned} V_{k+1} - V_k = {} & \frac{1}{N} I_f(x_\star, x_{k+1}) + \frac{k}{N} I_f(x_k, x_{k+1}) \\ & + \frac{k+1}{\alpha N} I_h(x_{k+1}, x_k) + \frac{k}{\alpha N} I_h(x_k, x_{k+1}) \leq 0. \end{aligned}$$

All multipliers are nonnegative, so $V_N \leq V_0 = 0$. There are no additional squares or residual families in this exact local certificate. Finally,

$$f(x_N) - f(x_\star) - \frac{R}{\alpha N} = V_N + \frac{D_h(x_\star, x_0) - R}{\alpha N} + \frac{I_h(x_\star, x_N)}{\alpha N} \leq 0.$$

Here the last two terms are nonpositive by the initial condition and convexity, respectively. The lower bound follows because $x_\star$ minimizes f. When $N = 1$, the initial step and boundary identity suffice; the proof also covers $R = 0$.

E.6 FEG

E.6.1 INPUT PROMPT

````
# Common assumptions

All cases use real finite-dimensional Euclidean spaces, positive `L`
where present, nonnegative `R`, and integer horizons `N >= 1`.
Proximal updates and stated evaluations are assumed well-defined.
Supply these common assumptions and only the selected case subsection
to its worker.


### `feg`


- Problem: monotone operators. `A` is monotone and `L`-Lipschitz.
- Initial condition: `A(x_star) = 0` and `||x_0 - x_star||^2 <= R^2`.
- Metric: `||A(x_N)||^2`.
- For `k = 0, ..., N-1`:

```text
x_{k+1/2} = x_k + (x_0-x_k)/(k+1) - k*A(x_k)/((k+1)*L)
x_{k+1} = x_k + (x_0-x_k)/(k+1) - A(x_{k+1/2})/L
```

- At `k = 0`, tag `x_0` also as `x_{1/2}`; they are the same point. Do
not create an independent intermediate iterate.
````

```
- Conjectured rate: unknown.
```

E.6.2 GENERATED OUTPUT (REPETITION 1)

Theorem
Let A be monotone and L-Lipschitz in a real finite-dimensional Euclidean space, with $L > 0$, $A(x_\star) = 0$, and $\|x_0 - x_\star\| \le R$ for $R \ge 0$. For integer $N \ge 1$, use

$$x_{k+1/2} = x_k + \frac{x_0 - x_k}{k+1} - \frac{k}{(k+1)L} A(x_k),$$
$$x_{k+1} = x_k + \frac{x_0 - x_k}{k+1} - \frac{1}{L} A(x_{k+1/2}),$$

for $0 \le k < N$, with $x_{1/2} = x_0$. Define the normalized potential

$$V_0 = 0, \qquad V_k = \frac{2k^2}{N^2}\|A(x_k)\|^2 + \frac{4kL}{N^2}\langle A(x_k), x_k - x_0\rangle, \quad 1 \le k \le N.$$

Then $V_{k+1} \le V_k$, and

$$\boxed{\|A(x_N)\|^2 \le \frac{4L^2R^2}{N^2}.}$$

The potential need not be nonnegative. Its terminal identity, rather than positivity, supplies the residual bound. The result covers $N = 1$ and $R = 0$.
Proof outline
Set $g_k = A(x_k)$, $d = x_0 - x_\star$, and define nonpositive residuals

$$\mathrm{Mon}(u, v) = -\langle A(u) - A(v), u - v\rangle,$$
$$\mathrm{Lip}(u, v) = \|A(u) - A(v)\|^2 - L^2\|u - v\|^2.$$

For $V_k = 2k^2\|g_k\|^2/N^2 + 4kL\langle g_k, x_k - x_0\rangle/N^2$ and $V_0 = 0$, the shared initial half-step gives

$$V_1 - V_0 = \frac{2}{N^2}\,\mathrm{Lip}(x_1, x_0) \le 0.$$

For every $1 \le k < N$, substitution of both updates gives the exact identity

$$V_{k+1} - V_k = \frac{4Lk(k+1)}{N^2}\,\mathrm{Mon}(x_k, x_{k+1}) + \frac{2(k+1)^2}{N^2}\,\mathrm{Lip}(x_{k+1}, x_{k+1/2}) \le 0.$$

There are no other local residual or square terms. All multipliers are positive on the stated domain, so $V_N \le V_0 = 0$. The terminal completion of the square is

$$\|g_N\|^2 - \frac{4L^2R^2}{N^2} = V_N + \frac{4L^2}{N^2}(\|d\|^2 - R^2) + \frac{4L}{N}\,\mathrm{Mon}(x_N, x_\star) - \left\|g_N - \frac{2L}{N}d\right\|^2.$$

Every term on the right is nonpositive: use the initial radius bound, monotonicity with $A(x_\star) = 0$, and nonnegativity of a squared norm. This proves the claimed rate. For $N = 1$ the interior telescoping sum is empty; the base and terminal identities still apply. No division by R or $\|g_N\|$ is needed.

E.7 DUAL-FEG

E.7.1 INPUT PROMPT

```
# Common assumptions

All cases use real finite-dimensional Euclidean spaces, positive `L`
where present, nonnegative `R`, and integer horizons `N >= 1`.
Proximal updates and stated evaluations are assumed well-defined.
```

````
Supply these common assumptions and only the selected case subsection
to its worker.


### `dual_feg`


- Problem: monotone operators. `A` is monotone and `L`-Lipschitz.
- Initial condition: `A(x_star) = 0`, `||x_0 - x_star||^2 <= R^2`, and
`z_0 = 0`.
- Parameters: fixed horizon `N >= 1` and `alpha = 1/L`.
- Performance metric: unknown.
- For `k = 0, ..., N-1`:

```text
x_{k+1/2} = x_k - alpha*z_k - alpha*A(x_k)
x_{k+1} = x_{k+1/2} - ((N-k-1)/(N-k))*alpha*(A(x_{k+1/2})-A(x_k))
z_{k+1} = ((N-k-1)/(N-k))*z_k - A(x_{k+1/2})/(N-k)
```

- Conjectured rate: unknown.
````

E.7.2 GENERATED OUTPUT (REPETITION 1)

Theorem
For every integer $N \geq 1$, $L > 0$ and $R \geq 0$, the stated Dual FEG iteration for a monotone L-Lipschitz operator with $A(x_\star) = 0$, $\|x_0 - x_\star\|^2 \leq R^2$ and $z_0 = 0$ satisfies

$$\boxed{\|A(x_N)\|^2 \leq \frac{4L^2R^2}{N^2}.}$$

Set $G = A(x_N)$, $a = x_0 - x_N$, and $b_k = x_k - x_N$. A certificate is $V_0 = 0$ and, for $1 \leq k \leq N-1$,

$$V_k = -\frac{4L}{N}\langle a, G\rangle - 2\|z_k\|^2 + \frac{4L}{N-k}\langle z_k, b_k\rangle$$
$$- 4\langle z_k, G\rangle + \frac{4L}{N-k}\langle b_k, G\rangle.$$

At the terminal index use the separate expression

$$V_N = 2\|G\|^2 - \frac{4L}{N}\langle a, G\rangle.$$

The sequence is nonincreasing. The interior expression also gives $V_0 = 0$ when evaluated at $k = 0$, $z_0 = 0$; it is never evaluated at $k = N$.
Proof outline
Let $y_k = x_{k+1/2}$ and define the nonpositive residuals

$$\mathrm{Lip}(u, v) = \|A(u) - A(v)\|^2 - L^2\|u - v\|^2,$$
$$\mathrm{Mon}(u, v) = -\langle A(u) - A(v), u - v\rangle.$$

With $G = A(x_N)$ and $a = x_0 - x_N$, write the interior potential as

$$V_k = -\frac{4L}{N}\langle a, G\rangle - 2\|z_k\|^2 - 4\langle z_k, G\rangle$$
$$+ \frac{4L}{N-k}\langle z_k + G, x_k - x_N\rangle.$$

Its initial value is zero. Substitution of the algorithm gives, for $0 \leq k \leq N-2$,

$$V_{k+1} - V_k = \frac{2}{(N-k)^2}\,\mathrm{Lip}(x_k, y_k) + \frac{4L}{(N-k)(N-k-1)}\,\mathrm{Mon}(x_N, y_k) \leq 0.$$

The actual final update is $x_N = x_{N-1} - (z_{N-1} + A(x_{N-1}))/L$. With the separate terminal potential $V_N = 2\|G\|^2 - (4L/N)\langle a, G\rangle$, it gives

$$V_N - V_{N-1} = 2\,\mathrm{Lip}(x_{N-1}, x_N) \leq 0.$$

For $N = 1$ there are no ordinary steps; the same last-step identity with $z_0 = 0$ starts directly from $V_0 = 0$. All multipliers are positive on their stated domains, so $V_N \le 0$ for every $N \ge 1$.
Finally, set $d = x_0 - x_\star$. The exact normalized identity is

$$\|G\|^2 - \frac{4L^2R^2}{N^2} = V_N + \frac{4L}{N}\,\mathrm{Mon}(x_N, x_\star) - \left\|\frac{2L}{N}d - G\right\|^2 + \frac{4L^2}{N^2}\big(\|d\|^2 - R^2\big).$$

Every term on the right is nonpositive: use $V_N \le 0$, monotonicity and $A(x_\star) = 0$, the square's nonnegativity, and the given initial-distance bound. This proves the theorem, including $R = 0$. The notebook below derives the coefficients and checks every identity exactly in Gram, function-value and constant coordinates.

F PROOFS FOR THE CASE STUDIES

This appendix records the complete Lyapunov certificates behind Theorems 5.1–5.4 and 5.5. We used `GPT-6 Astra` with `Ultra` reasoning effort throughout these case studies. The corresponding input prompt is recorded at the start of each proof subsection.

Throughout this appendix, cited code cells refer to exact algebraic verifications in the companion notebook specified for the corresponding proof. Cell numbers are the 1-based positions among code cells, excluding Markdown and raw cells. After restarting the kernel and executing each code cell once from top to bottom, these numbers agree with Jupyter's `In[n]` labels.

Table 4: Theorem correspondences and companion notebooks.

Appendix F	Main text	Companion notebook
Theorem F.1	Theorem 5.4	`fgm_rational_tight_example_lyap.ipynb`
Theorem F.3	Theorem 5.1	`fgm_conjecture4_example_lyap.ipynb`
Theorem F.5	Theorem 5.2	`fgm_conjecture5_example_lyap.ipynb`
Theorem F.7	Theorem 5.3	`ogm_conjecture4_example_lyap.ipynb`
Theorem F.9	Theorem 5.5	`fista_rational_tight_example_lyap.ipynb`

We use the Hilbert space $\mathcal{H}$ and the function class $\mathcal{F}_L$ defined in Appendix A. Thus $f : \mathcal{H} \to \mathbb{R}$, and the iterates x_k, y_k, z_k, the minimizer $x_\star$, and the gradients $\nabla f(x_k)$ belong to $\mathcal{H}$. The parameters satisfy $L, R, \theta_k \in \mathbb{R}$. For each trajectory, the potentials V_k and square remainders S_k defined below take values in $\mathbb{R}$. For $p, q \in \mathcal{H}$, we use the smooth convex interpolation residual $\mathcal{I}_f : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$

$$\mathcal{I}_f(p,q) = f(q) - f(p) + \langle \nabla f(q), p - q\rangle + \frac{1}{2L}\|\nabla f(p) - \nabla f(q)\|^2 \le 0. \tag{11}$$

For FGM and OGM, set $\theta_{-1} = 0$. Their parameter recurrence gives the following identity for $k \ge 0$ in FGM and $0 \le k < N$ in OGM:[4]

$$\theta_k^2 - \theta_k = \theta_{k-1}^2. \tag{12}$$

In the proofs for FGM, define the scalars $C_k \in \mathbb{R}$ by

$$C_0 = 0, \qquad C_k = \frac{1}{\theta_{k-1}^2}\sum_{j=0}^{k-1}\theta_j^3, \qquad k \ge 1. \tag{13}$$

The definition of C_k gives

$$\theta_k^2 C_{k+1} = \theta_{k-1}^2 C_k + \theta_k^3, \qquad k \ge 0.$$

Combining this with (12) yields the recurrence used in the proofs below, with $\delta_k \in \mathbb{R}$:

$$C_{k+1} = \theta_k + \left(1 - \frac{1}{\theta_k}\right)C_k, \qquad \delta_k = C_{k+1} - C_k. \tag{14}$$

[4] In standard OGM, $\theta_N = (1 + \sqrt{1 + 8\theta_{N-1}^2})/2$ instead. The y_N conjecture proved here uses only $\theta_0, \ldots, \theta_{N-1}$.

The correspondence between these FGM coefficients and the original conjecture notation is verified in Appendix F.6.

The expanded certificate theorem in each subsection proves an upper bound. Tightness follows by combining these bounds with the matching lower bounds established by Taylor et al. (2017c, Section 4.2) for FGM, Kim & Fessler (2017, Proposition 4.1) for OGM, and Taylor et al. (2017a, Table 1) for FGM with rational coefficients.

The proofs of Theorems 5.1–5.3 follow the same structure as that of Theorem 5.4. We therefore present Theorem 5.4 first: its rational coefficient sequence makes the calculations comparatively simple and gives the clearest introduction to the common argument.

F.1 Proof of Theorem 5.4 (Table 1 conjecture in Taylor et al. (2017a))

F.1.1 Input prompt

```
/pep-implement

Function: f is convex and L-smooth
Parameters: L, R
Initial condition: ||x_0 - x_star|| <= R, where grad f(x_star) = 0
Performance metric: f(y_N) - f(x_star)
Algorithm: Fast gradient method with rational coefficients and fixed
    step size 1/L

Parameter sequence: theta_k = (k+2)/2 for k >= 0
Initialization: x_0 = y_0 = z_0
For k = 0,...,N-1:
y_{k+1} = x_k - (1/L) * grad f(x_k)
z_{k+1} = z_k - (theta_k/L) * grad f(x_k)
x_{k+1} = (1 - 1/theta_{k+1}) * y_{k+1} + (1/theta_{k+1}) * z_{k+1}

Conjectured rate: f(y_N) - f(x_star) <= 2 * L * R^2 / ((N+2) * (N+3))
```

Then, we repeatedly entered:

```
proceed with the next step
```

Proof outline for Theorem 5.4. Set $\tau_N = 2/[(N+2)(N+3)] \in \mathbb{R}$. For $1 \le k < N$, define $\boldsymbol{w}_k \in \mathcal{H}^2$ by

$$\boldsymbol{w}_k = \begin{bmatrix} z_k - x_\star \\ x_k - x_\star \end{bmatrix}.$$

Set $V_0 = 0 \in \mathbb{R}$. For $N \ge 2$, define V_1 by

$$\begin{aligned} V_1 = a_1\big(f(x_1) - f_\star\big) - \tau_N L \left\|x_0 - x_\star\right\|^2 + \frac{a_1}{2L}\left\|\nabla f(x_1)\right\|^2 \\ - b_1 \left\langle \nabla f(x_1), x_1 - x_\star \right\rangle + \frac{L\tau_N(1-4\tau_N)}{(1-2\tau_N)^2}\left\|x_1 - x_\star\right\|^2 \end{aligned} \tag{15}$$

For $2 \le k < N$, with $\mathbf{d}_k \in \mathbb{R}^2$ specified below, define the interior potential V_k by

$$\begin{aligned} V_k = a_k\big(f(x_k) - f(x_\star)\big) - \tau_N L \left\|x_0 - x_\star\right\|^2 + \frac{a_k}{2L}\left\|\nabla f(x_k)\right\|^2 \\ - 2\left\langle \nabla f(x_k), \mathbf{d}_k^\top \boldsymbol{w}_k \right\rangle + L\boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k. \end{aligned} \tag{16}$$

At the reported point, set

$$V_N = f(y_N) - f(x_\star) - \tau_N L \left\|x_0 - x_\star\right\|^2 + \frac{1}{2L}\left\|\nabla f(y_N) - L b_N (y_N - x_\star)\right\|^2. \tag{17}$$

We will show that each consecutive difference, or decrement, of V_k is a positive-weighted sum of interpolation residuals minus at most two nonnegative squares. For $N \geq 2$, we show

$$V_{k+1} - V_k = \begin{cases} b_0\mathcal{I}_f(x_\star, x_0) + a_0\mathcal{I}_f(x_0, x_1) + b_1\mathcal{I}_f(x_\star, x_1) - S_0, & k = 0, \\ a_k\mathcal{I}_f(x_k, x_{k+1}) + b_{k+1}\mathcal{I}_f(x_\star, x_{k+1}) - S_k, & 1 \leq k < N-1, \\ a_{N-1}\mathcal{I}_f(x_{N-1}, y_N) + b_N\mathcal{I}_f(x_\star, y_N) - S_{N-1}, & k = N-1, \end{cases} \tag{18}$$

for some positive numbers a_k, b_k and sum-of-squares $S_k \in \mathbb{R}$ with $S_k \geq 0$.[5] Together with the coefficient signs in Lemma F.2, these identities imply $V_N \leq V_0 = 0$, and dropping the nonnegative square term in V_N yields (7). Below, we state a separate theorem which fully characterizes the Lyapunov analysis and prove it in detail. □

Theorem F.1 (Theorem 5.4 with full details of the Lyapunov function)**.** Let $f \in \mathcal{F}_L$, let $x_\star \in \mathcal{H}$ minimize f, and write $f_\star = f(x_\star) \in \mathbb{R}$. Suppose that $\|x_0 - x_\star\| \leq R$. For a fixed integer horizon $N \geq 1$, set

$$\tau_N = \frac{2}{(N+2)(N+3)} \in \mathbb{R}.$$

For $0 \leq k < N$, define the real scalars $D_k, a_k, b_k \in \mathbb{R}$ by

$$D_k = 2N^2 + 10N + 4 - k^2 - 7k,$$

$$a_k = \frac{(k+1)\{2N(N+1)(k+4) - k(k+2)(k+3)\}}{N(N+1)D_k}, \qquad a_{-1} = 0, \qquad a_N = 1. \tag{19}$$

$$b_k = a_k - a_{k-1}, \qquad 0 \leq k \leq N. \tag{20}$$

For $1 \leq k < N$, define the state $\boldsymbol{w}_k \in \mathcal{H}^2$, the matrix $M_k \in \mathbb{R}^{2\times 2}$, and the coefficient vectors $\mathbf{c}_k, \mathbf{d}_k \in \mathbb{R}^2$ by

$$\boldsymbol{w}_k = \begin{bmatrix} z_k - x_\star \\ x_k - x_\star \end{bmatrix}, \qquad M_k = \begin{pmatrix} 1 & 0 \\ 2/(k+3) & (k+1)/(k+3) \end{pmatrix},$$

$$\mathbf{c}_k = \begin{bmatrix} (k+2)/2 \\ (2k+3)/(k+3) \end{bmatrix}, \qquad \mathbf{d}_k = \frac{1}{2}\begin{bmatrix} 2a_{k-1}/k \\ a_k - (k+2)a_{k-1}/k \end{bmatrix}.$$

For $2 \leq k < N$, define $\boldsymbol{\xi}_k, \boldsymbol{\zeta}_k \in \mathbb{R}^2$ by

$$\boldsymbol{\xi}_k = \begin{bmatrix} 1 - \tau_N k(k+3)/2 \\ 1 - \tau_N k(k+7)/4 \end{bmatrix},$$

$$\boldsymbol{\zeta}_k = \tau_N \begin{bmatrix} 2a_{k-1}/k + 1 - (k+1)(k+2)/[N(N+1)] \\ (k+1)(k+2)/[N(N+1)] - (k+2)a_{k-1}/k \end{bmatrix}.$$

The same expressions are used at adjacent indices when they occur below. For $2 \leq k < N$, define an auxiliary matrix $\overline{\mathbf{Q}}_k \in \mathbb{R}^{2\times 2}$ satisfying $\overline{\mathbf{Q}}_k \boldsymbol{\xi}_k = \boldsymbol{\zeta}_k$ by

$$\overline{\mathbf{Q}}_k = \frac{(\boldsymbol{\zeta}_k)_1}{(\boldsymbol{\xi}_k)_2}\begin{pmatrix} 0 & 1 \\ 1 & \dfrac{(\boldsymbol{\zeta}_k)_2}{(\boldsymbol{\zeta}_k)_1} - \dfrac{(\boldsymbol{\xi}_k)_1}{(\boldsymbol{\xi}_k)_2} \end{pmatrix}.$$

With $\boldsymbol{e}_1 = [1, 0]^\top \in \mathbb{R}^2$, define $\boldsymbol{n}_k \in \mathbb{R}^2$ and $\mathbf{Q}_k \in \mathbb{R}^{2\times 2}$ by

$$\boldsymbol{n}_k = \begin{bmatrix} 1 \\ -(\boldsymbol{\xi}_k)_1/(\boldsymbol{\xi}_k)_2 \end{bmatrix},$$

$$\mathbf{Q}_k = \overline{\mathbf{Q}}_k + \frac{(\mathbf{d}_{k-1})_1 - \boldsymbol{e}_1^\top M_{k-1}^\top \overline{\mathbf{Q}}_k \mathbf{c}_{k-1}}{(\boldsymbol{e}_1^\top M_{k-1}^\top \boldsymbol{n}_k)(\boldsymbol{n}_k^\top \mathbf{c}_{k-1})} \boldsymbol{n}_k \boldsymbol{n}_k^\top. \tag{21}$$

Define the Lyapunov values $V_k \in \mathbb{R}$ as in (15), (16), and (17), *i.e.*, set $V_0 = 0$. For $N \geq 2$, define V_1 by

$$V_1 = a_1\big(f(x_1) - f(x_\star)\big) - \tau_N L\|x_0 - x_\star\|^2 + \frac{a_1}{2L}\|\nabla f(x_1)\|^2$$

$$- b_1\langle \nabla f(x_1), x_1 - x_\star\rangle + \frac{L\tau_N(1 - 4\tau_N)}{(1 - 2\tau_N)^2}\|x_1 - x_\star\|^2$$

[5] For $N = 1$, the first line holds with x_1 replaced by y_1 and that is the terminal identity.

For $2 \le k < N$, define V_k by

$$V_k = a_k\big(f(x_k) - f(x_\star)\big) - \tau_N L \|x_0 - x_\star\|^2 + \frac{a_k}{2L}\|\nabla f(x_k)\|^2 - 2\left\langle \nabla f(x_k), \mathbf{d}_k^\top \boldsymbol{w}_k \right\rangle + L \boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k,$$

At the reported point, set

$$V_N = f(y_N) - f(x_\star) - \tau_N L \|x_0 - x_\star\|^2 + \frac{1}{2L}\|\nabla f(y_N) - L b_N (y_N - x_\star)\|^2.$$

For $2 \le k < N$, define the vectors $s_k, t_k \in \mathcal{H}$ by

$$s_k = \frac{\nabla f(x_k)}{L} - \frac{2\tau_N}{(\boldsymbol{\xi}_k)_2}(x_k - x_\star), \qquad t_k = \boldsymbol{n}_k^\top \boldsymbol{w}_k.$$

The boundary square terms are defined below, with $\alpha_1 \in \mathbb{R}$:

$$\begin{aligned} S_0 &= \frac{L\tau_N}{(1-2\tau_N)^2}\left\|\frac{\nabla f(x_0)}{L} - 2\tau_N(x_0 - x_\star)\right\|^2, \\ S_1 &= L\alpha_1\left\|\frac{\nabla f(x_1)}{L} - \frac{2\tau_N}{1-2\tau_N}(x_1 - x_\star)\right\|^2, \qquad \alpha_1 = \frac{(N^2+5N+2)(5N^2+5N-6)}{N(N+1)(N^2+5N-2)^2}, \end{aligned} \tag{22}$$

where the second line is used when $N \ge 3$. For $2 \le k \le N-2$, define $\alpha_k, \beta_k \in \mathbb{R}$ and S_k by

$$\begin{aligned} \alpha_k &= a_k - \mathbf{c}_k^\top \mathbf{Q}_{k+1} \mathbf{c}_k = \frac{a_k(D_k+8)}{2D_k}, \\ \beta_k &= (\mathbf{Q}_k - M_k^\top \mathbf{Q}_{k+1} M_k)_{11}, \qquad S_k = L\alpha_k \|s_k\|^2 + L\beta_k \|t_k\|^2. \end{aligned} \tag{23}$$

For $N \ge 3$, define the terminal coefficients $\alpha_{N-1}, \beta_{N-1} \in \mathbb{R}$ and square S_{N-1} by

$$\begin{aligned} \alpha_{N-1} &= 1 - b_N - \frac{b_N^2}{2}, \qquad \beta_{N-1} = (\mathbf{Q}_{N-1})_{11} - \frac{a_{N-2}^2}{(N-1)^2\alpha_{N-1}}, \\ S_{N-1} &= L\alpha_{N-1}\left\|s_{N-1} - \frac{a_{N-2}}{(N-1)\alpha_{N-1}} t_{N-1}\right\|^2 + L\beta_{N-1}\|t_{N-1}\|^2, \end{aligned} \tag{24}$$

while for $N = 2$, we instead set

$$S_1 = \frac{11L}{18}\left\|\frac{\nabla f(x_1)}{L} - \frac{x_1 - x_\star}{4}\right\|^2. \tag{25}$$

The square sums satisfy $S_k \ge 0$. For $N \ge 2$, the following decrement identities hold:[6]

$$V_{k+1} - V_k = \begin{cases} b_0 \mathcal{I}_f(x_\star, x_0) + a_0 \mathcal{I}_f(x_0, x_1) + b_1 \mathcal{I}_f(x_\star, x_1) - S_0, & k = 0, \\ a_k \mathcal{I}_f(x_k, x_{k+1}) + b_{k+1}\mathcal{I}_f(x_\star, x_{k+1}) - S_k, & 1 \le k < N-1, \\ a_{N-1}\mathcal{I}_f(x_{N-1}, y_N) + b_N \mathcal{I}_f(x_\star, y_N) - S_{N-1}, & k = N-1. \end{cases}$$

The coefficient signs in Lemma F.2 and the nonpositive interpolation residuals imply $V_N \le \cdots \le V_0 = 0$. Dropping the nonnegative square in V_N and applying the radius assumption gives

$$f(y_N) - f_\star \le \tau_N L \|x_0 - x_\star\|^2 \le \frac{2LR^2}{(N+2)(N+3)}.$$

Proof. The companion notebook is `fgm_rational_tight_example_lyap.ipynb`, in `examples_peppy/fgm_rational_tight/`. The notebook verifies the same certificate using $\mathbf{Q}_k$ and the dimensionless constant τ_N, with different auxiliary notation. The exact verification section starts at Cell 31 and can be run independently of the numerical discovery blocks. Its symbolic identities hold for arbitrary parameters on the stated domains. Cell 36 provides an additional check of the complete potential decrease identities at $N = 1, 2, 3, 4, 6, 8$.

[6] For $N = 1$, use the first case with x_1 replaced by y_1; this is the single base–terminal step.

For $2 \le k \le N-2$, subtracting two consecutive instances of (16) cancels the common initial-radius term and gives

$$\begin{aligned} V_{k+1} - V_k = {} & a_{k+1}\big(f(x_{k+1}) - f(x_\star)\big) - a_k\big(f(x_k) - f(x_\star)\big) \\ & + \frac{a_{k+1}}{2L}\left\|\nabla f(x_{k+1})\right\|^2 - 2\left\langle \nabla f(x_{k+1}), \mathbf{d}_{k+1}^\top \boldsymbol{w}_{k+1}\right\rangle + L\boldsymbol{w}_{k+1}^\top \mathbf{Q}_{k+1}\boldsymbol{w}_{k+1} \\ & - \frac{a_k}{2L}\left\|\nabla f(x_k)\right\|^2 + 2\left\langle \nabla f(x_k), \mathbf{d}_k^\top \boldsymbol{w}_k\right\rangle - L\boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k. \end{aligned} \tag{26}$$

Our goal is to identify this expression with the interior case of (18). We first factor the quadratic remainder, then match the weighted interpolation residuals; the boundary cases are treated separately.

Step 1: Expressing the remainder as a sum of squares. For $2 \le k \le N-2$, Steps 1 and 2 prove the second line of (18), namely,

$$V_{k+1} - V_k = a_k \mathcal{I}_f(x_k, x_{k+1}) + b_{k+1}\mathcal{I}_f(x_\star, x_{k+1}) - S_k,$$

with S_k as defined in (23). We first factor the quadratic remainder into the two squares defining S_k; Step 2 identifies this remainder in the Lyapunov difference. The FGM-rational updates give (notebook Cell 32)

$$\boldsymbol{w}_{k+1} = M_k \boldsymbol{w}_k - \mathbf{c}_k \frac{\nabla f(x_k)}{L}. \tag{27}$$

Define the symmetric matrix $\mathbf{S}_k \in \mathbb{R}^{3\times 3}$ in the coordinates $(\boldsymbol{w}_k, \nabla f(x_k)/L)$ by

$$\mathbf{S}_k = \begin{pmatrix} \mathbf{Q}_k - M_k^\top \mathbf{Q}_{k+1} M_k & -\mathbf{d}_k + M_k^\top \mathbf{Q}_{k+1}\mathbf{c}_k \\ \left(-\mathbf{d}_k + M_k^\top \mathbf{Q}_{k+1}\mathbf{c}_k\right)^\top & a_k - \mathbf{c}_k^\top \mathbf{Q}_{k+1}\mathbf{c}_k \end{pmatrix}. \tag{28}$$

Quadratic forms in these coordinates use inner products of the component vectors. The definition of $\mathbf{Q}_k$ gives the following two identities, verified in notebook Cell 33:

$$\mathbf{Q}_k \boldsymbol{\xi}_k = \boldsymbol{\zeta}_k, \qquad \left(-\mathbf{d}_{k-1} + M_{k-1}^\top \mathbf{Q}_k \mathbf{c}_{k-1}\right)_1 = 0.$$

The same cell verifies all three transport identities by direct substitution:

$$\begin{aligned} M_k \boldsymbol{\xi}_k - 2\tau_N \mathbf{c}_k &= \boldsymbol{\xi}_{k+1}, \\ \boldsymbol{\zeta}_k - M_k^\top \boldsymbol{\zeta}_{k+1} - 2\tau_N \mathbf{d}_k &= 0, \\ -\mathbf{d}_k^\top \boldsymbol{\xi}_k + \mathbf{c}_k^\top \boldsymbol{\zeta}_{k+1} + 2\tau_N a_k &= 0. \end{aligned} \tag{29}$$

Using $\mathbf{Q}_j \boldsymbol{\xi}_j = \boldsymbol{\zeta}_j$ at $j = k, k+1$ together with (29), we find that $\mathbf{S}_k$ annihilates $[(\boldsymbol{\xi}_k)_1, (\boldsymbol{\xi}_k)_2, 2\tau_N]^\top$. The cross-term identity at index $k+1$ and the diagonal entries identify the following coefficients of the quadratic form of $\mathbf{S}_k$ in $(\boldsymbol{w}_k, \nabla f(x_k)/L)$:

$$\begin{aligned} \left\|(\boldsymbol{w}_k)_1\right\|^2 : &\quad (\mathbf{S}_k)_{11} = \beta_k, \\ \frac{2}{L}\left\langle \nabla f(x_k), (\boldsymbol{w}_k)_1\right\rangle : &\quad (\mathbf{S}_k)_{13} = 0, \\ \frac{1}{L^2}\left\|\nabla f(x_k)\right\|^2 : &\quad (\mathbf{S}_k)_{33} = \alpha_k. \end{aligned}$$

Together with the kernel relation, these coefficients determine the matrix as

$$\mathbf{S}_k = \alpha_k \begin{bmatrix} 0 \\ -2\tau_N/(\boldsymbol{\xi}_k)_2 \\ 1 \end{bmatrix}\begin{bmatrix} 0 \\ -2\tau_N/(\boldsymbol{\xi}_k)_2 \\ 1 \end{bmatrix}^\top + \beta_k \begin{bmatrix} 1 \\ -(\boldsymbol{\xi}_k)_1/(\boldsymbol{\xi}_k)_2 \\ 0 \end{bmatrix}\begin{bmatrix} 1 \\ -(\boldsymbol{\xi}_k)_1/(\boldsymbol{\xi}_k)_2 \\ 0 \end{bmatrix}^\top.$$

Evaluating the associated quadratic form and using the definitions of s_k, t_k gives

$$L\begin{bmatrix} \boldsymbol{w}_k \\ \nabla f(x_k)/L \end{bmatrix}^\top \mathbf{S}_k \begin{bmatrix} \boldsymbol{w}_k \\ \nabla f(x_k)/L \end{bmatrix} = L\alpha_k \left\|s_k\right\|^2 + L\beta_k \left\|t_k\right\|^2 = S_k,$$

The $(3,3)$ entry also yields the displayed formula for α_k in (23). Notebook Cell 33 verifies the kernel, the zero $(1,3)$ entry, every entry of this matrix factorization, and the displayed rational formula for α_k.

Step 2: Matching the remainder to the Lyapunov difference. It remains to show that the quadratic form evaluated in Step 1 is the difference between the weighted interpolation residuals and $V_{k+1} - V_k$. To match (26), expand the weighted residuals. Since $b_{k+1} = a_{k+1} - a_k$ and $\nabla f(x_\star) = 0$, their function-value terms agree with the potential difference:

$$\begin{aligned} &a_k \mathcal{I}_f(x_k, x_{k+1}) + b_{k+1}\mathcal{I}_f(x_\star, x_{k+1}) \\ &\quad = a_{k+1}\big(f(x_{k+1}) - f(x_\star)\big) - a_k\big(f(x_k) - f(x_\star)\big) \\ &\qquad + \frac{a_k}{2L}\left\|\nabla f(x_k)\right\|^2 + \frac{a_{k+1}}{2L}\left\|\nabla f(x_{k+1})\right\|^2 \\ &\qquad + \left\langle \nabla f(x_{k+1}), a_k\left(x_k - x_{k+1} - \frac{\nabla f(x_k)}{L}\right) + b_{k+1}(x_\star - x_{k+1})\right\rangle. \end{aligned} \tag{30}$$

The remaining inner-product coefficient follows from $y_{k+1} = x_k - \nabla f(x_k)/L$ and the mixing update:

$$\begin{aligned} &a_k\left(x_k - x_{k+1} - \frac{\nabla f(x_k)}{L}\right) + b_{k+1}(x_\star - x_{k+1}) \\ &\quad = a_k(y_{k+1} - x_\star) - a_{k+1}(x_{k+1} - x_\star) \\ &\quad = -\frac{2a_k}{k+1}(z_{k+1} - x_\star) + \left(\frac{(k+3)a_k}{k+1} - a_{k+1}\right)(x_{k+1} - x_\star) = -2\mathbf{d}_{k+1}^\top \boldsymbol{w}_{k+1}. \end{aligned} \tag{31}$$

Subtracting (26) now cancels all function-value terms and all terms involving $\nabla f(x_{k+1})$. The remaining expression is

$$L\big(\boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k - \boldsymbol{w}_{k+1}^\top \mathbf{Q}_{k+1}\boldsymbol{w}_{k+1}\big) + \frac{a_k}{L}\left\|\nabla f(x_k)\right\|^2 - 2\left\langle \nabla f(x_k), \mathbf{d}_k^\top \boldsymbol{w}_k\right\rangle.$$

Substituting (27) into the remaining quadratic terms gives

$$\begin{aligned} &a_k \mathcal{I}_f(x_k, x_{k+1}) + b_{k+1}\mathcal{I}_f(x_\star, x_{k+1}) - (V_{k+1} - V_k) \\ &\quad = L\begin{bmatrix} \boldsymbol{w}_k \\ \nabla f(x_k)/L \end{bmatrix}^\top \mathbf{S}_k \begin{bmatrix} \boldsymbol{w}_k \\ \nabla f(x_k)/L \end{bmatrix} = S_k, \end{aligned}$$

where the last equality is the factorization proved in Step 1. Rearranging proves the second line of (18) for $2 \le k \le N-2$. The case $k = 1$ uses the separate potential (15) and is treated below. The signs needed to deduce $V_{k+1} \le V_k$ are established in Lemma F.2. Notebook Cell 37 verifies this cancellation in every quadratic, function-value, and constant coefficient, including the specialization to the first interior step.

Boundary indices. We now verify the remaining cases of (18): the base step, the first interior step $k = 1$ for $N \ge 3$, and the terminal step. Using the boundary potentials (15) and (17), we identify their quadratic remainders with the stated S_0, S_1, S_{N-1}. At $k = 1$, for $N \ge 3$, $\boldsymbol{w}_1 = [x_1 - x_\star, x_1 - x_\star]^\top$. Substitution of $\mathbf{Q}_2$ leaves the rank-one quadratic

$$L\alpha_1\left\|\frac{\nabla f(x_1)}{L} - \frac{2\tau_N}{1-2\tau_N}(x_1 - x_\star)\right\|^2,$$

which is S_1 in (22). Notebook Cell 35 verifies this rank-one factorization and the stated formula for α_1.

At $k = N-1$, for $N \ge 3$, use $y_N - x_\star = [0,1]\boldsymbol{w}_{N-1} - \nabla f(x_{N-1})/L$ and the scalar $b_N^2/2$ from the terminal square. The resulting block has the same kernel as (28), with $(1,3)$ entry $-\frac{a_{N-2}}{N-1}$, $(3,3)$ entry α_{N-1}, and $(1,1)$ entry $(\mathbf{Q}_{N-1})_{11}$. Completing the square gives (24). Notebook Cell 35 verifies the terminal block, its kernel, and the completed-square factorization; Cell 39 verifies the full terminal decrement identity, including the function values.

For the base step, substituting $a_0 = 2\tau_N/(1-2\tau_N)$ and $x_1 = x_0 - \nabla f(x_0)/L$ gives S_0 in (22) (notebook Cells 35 and 38).[7]

[7] When $N = 1$, $a_0 = b_0 = b_1 = 1/2$, and the same calculation uses the terminal definition of V_1 (Cell 40). When $N = 2$, $\tau_N = 1/10$, $a_0 = 1/4$, $a_1 = 2/3$, and $b_2 = 1/3$; the remaining quadratic is (25) (Cell 41).

Positivity of the coefficients. Lemma F.2 proves the coefficient signs and the bounds needed for the defining denominators. Hence every $S_k \geq 0$, and the nonpositive interpolation residuals in (18) give $V_{k+1} \leq V_k$.

Conclusion. Thus $V_N \leq \cdots \leq V_0 = 0$. Dropping the nonnegative square in (17) and using $\|x_0 - x_\star\| \leq R$ proves the claimed bound.

□

Lemma F.2. The coefficients are positive on their respective index ranges:[8]

$$a_k > 0, \qquad b_k > 0, \qquad \alpha_k > 0, \qquad \beta_k > 0.$$

The remaining boundary-square coefficients in (22) and (25) are also positive.

Proof. For $0 \leq k < N$, $D_k \geq D_{N-1} = N^2 + 5N + 10 > 0$. For $N \geq 3$, use the polynomials $p_4, p_9 : \mathbb{R} \to \mathbb{R}$ defined by

$$\begin{aligned} p_4(N) &= N^4 + 10N^3 + 37N^2 + 60N - 12, \\ p_9(N) &= N^9 + 26N^8 + 280N^7 + 1696N^6 + 6325N^5 + 13538N^4 \\ &\quad + 11402N^3 - 4844N^2 - 2904N + 5456. \end{aligned}$$

For the following expansions, define $\ell := N - 3 \in \mathbb{R}_{\geq 0}$:

$$\begin{aligned} p_4(3+\ell) &= \ell^4 + 22\ell^3 + 181\ell^2 + 660\ell + 852, \\ p_9(3+\ell) &= \ell^9 + 53\ell^8 + 1228\ell^7 + 16396\ell^6 + 139291\ell^5 + 780011\ell^4 \\ &\quad + 2867792\ell^3 + 6635500\ell^2 + 8715168\ell + 4933568. \end{aligned}$$

Both polynomials are positive for $\ell \geq 0$.

- **Positivity of a_k.** The coefficient formula (19) gives the initial value and the successive differences (for $1 \leq k < N$):

$$D_0 a_0 = 8, \qquad D_k a_k - D_{k-1} a_{k-1} = \frac{4(k+2)(N-k)(N+k+1)}{N(N+1)} > 0.$$

  Since $D_k > 0$ and $D_{k-1} - D_k = 2(k+3) > 0$, these identities imply that $a_k > 0$ for $0 \leq k < N$ and that a_k is strictly increasing through $k = N-1$. The remaining value is $a_N = 1$. For $2 \leq k < N$, $(k+1)(k+2) \leq N(N+1)$, so

$$(\boldsymbol{\zeta}_k)_1 = \tau_N \left(\frac{2a_{k-1}}{k} + 1 - \frac{(k+1)(k+2)}{N(N+1)} \right) \geq \frac{2\tau_N a_{k-1}}{k} > 0.$$

  The remaining denominators in (21) are nonzero because of the following factorizations, also verified in notebook Cell 34:

$$\begin{aligned} (\boldsymbol{\xi}_k)_2 &= \frac{D_k + 8}{2(N+2)(N+3)} > 0, \\ (\boldsymbol{e}_1^\top M_{k-1}^\top \boldsymbol{n}_k)(\boldsymbol{n}_k^\top \mathbf{c}_{k-1}) &= \frac{k^2(k-1)D_{k-2}D_{k-1}}{2(k+2)^2(D_k+8)^2} > 0. \end{aligned}$$

- **Positivity of b_k.** For $1 \leq k < N$, strict increase gives $b_k = a_k - a_{k-1} > 0$. At the endpoints,

$$b_0 = a_0 > 0, \qquad b_N = 1 - a_{N-1} = \frac{8}{N^2 + 5N + 10} > 0.$$

  Notebook Cell 34 verifies the identities for D_k, a_k, b_k used in these two items.

[8] The unused value a_{-1} is zero.

- **Positivity of α_k.** For $N \geq 3$, the numerator factors in α_1 in (22) are positive. For $2 \leq k \leq N-2$, (23) gives
$$\alpha_k = \frac{a_k(D_k+8)}{2D_k} > 0.$$
At the terminal index,
$$\alpha_{N-1} = \frac{p_4(N)}{(N^2+5N+10)^2} > 0.$$
Notebook Cells 33 and 35 verify the interior and boundary formulas.

- **Positivity of β_k.** For $2 \leq k \leq N-2$, set $m = k-2 \in \mathbb{R}$ and $\ell = N-k-2 \in \mathbb{R}$. Substitution into (21) gives
$$\beta_k = \frac{2(D_k+8)P_{m,\ell}}{Nk^2(N+1)(k-1)D_{k-2}^2 D_{k-1}^2},$$
where $P_{m,\ell} \in \mathbb{R}$ is given by
$$\begin{aligned} P_{m,\ell} = {} & 15m^7 + m^6(140\ell+753) + m^5(446\ell^2+5214\ell+14835) \\ & + m^4(624\ell^3+12008\ell^2+74836\ell+150455) \\ & + m^3(532\ell^4+13176\ell^3+126946\ell^2+545318\ell+861038) \\ & + m^2(240\ell^5+7988\ell^4+101880\ell^3+658392\ell^2+2157668\ell+2814280) \\ & + m\big(40\ell^6+2072\ell^5+37696\ell^4+336856\ell^3 \\ & \qquad + 1663624\ell^2+4416768\ell+4903840\big) \\ & + 112\ell^6+3920\ell^5+54576\ell^4+395056\ell^3 \\ & + 1620512\ell^2+3649344\ell+3529344. \end{aligned}$$
Every coefficient is positive and $m, \ell \geq 0$, so $\beta_k > 0$ on the interior range. Notebook Cell 33 verifies the rational expression for β_k against (21), checks the displayed polynomial $P_{m,\ell}$ coefficient by coefficient, and checks its positive coefficients and constant term.

  For $N \geq 3$, the terminal coefficient satisfies
$$\beta_{N-1} = \frac{(N^2+5N+18)^2 p_9(N)}{(N-2)(N-1)^2(N^2+7N+14)^2(N^2+9N+16)^2 p_4(N)} > 0.$$
Notebook Cell 35 verifies both terminal rational expressions and both shifted-polynomial expansions above, including their coefficient signs.

Since $0 < \tau_N \leq 1/6$, the coefficient of S_0 in (22) is positive.[9] □

F.2 Proof of Theorem 5.1 (Conjecture 4 in Taylor et al. (2017c))

For this case, we used the following initial prompt.

```
/pep-implement

Function: f is convex and L-smooth
Parameters: L, R
Initial condition: ||x_0 - x_star|| <= R, where grad f(x_star) = 0
Performance metric: f(y_N) - f(x_star)
Algorithm: Fast gradient method with fixed step size 1/L

Parameter sequence: theta_0 = 1 and theta_{k+1} = (1 + sqrt(1 + 4 *
    theta_k^2)) / 2 for k >= 0
Initialization: x_0 = y_0 = z_0
For k = 0,...,N-1:
```

[9] When $N = 2$, the coefficient $11L/18$ in (25) is positive as well.

```
y_{k+1} = x_k - (1/L) * grad f(x_k)
z_{k+1} = z_k - (theta_k/L) * grad f(x_k)
x_{k+1} = (1 - 1/theta_{k+1}) * y_{k+1} + (1/theta_{k+1}) * z_{k+1}

Conjectured rate: f(y_N) - f(x_star) <= L * R^2 / (2 * (2 * C_N + 1)),
    where C_N = 1 + sum_{j=0}^{N-2} h_{N-1,j} and x_i = x_0 - (1/L) *
    sum_{j=0}^{i-1} h_{i,j} * grad f(x_j)
```

Then, we repeatedly entered:

```
proceed with the next step
```

The initial proof of Conjecture 4 required a lengthy and intricate argument to establish positivity of the coefficients. We therefore asked the agent to revise it using the two-square decomposition in the FGM-rational proof (Appendix F.1), with the additional prompt below. This led to formulas for the square coefficients and a simpler positivity argument, as presented in the proof below.

```
Could you explore whether allowing two squared-norm terms per
iteration, as in fgm_rational_tight_example_lyap.ipynb, can give
explicit formulas for the square coefficients and simplify their
positivity proof for FGM Conjecture 4?
```

Proof outline for Theorem 5.1. Set $D_k = 2C_N + 1 - C_k \in \mathbb{R}$ and $\tau_N = 1/(2D_0) \in \mathbb{R}$. Define the Lyapunov function by $V_0 = 0$ and

$$V_k = a_k\big(f(x_k) - f(x_\star)\big) - \tau_N L \left\|x_0 - x_\star\right\|^2 - \frac{a_k}{2L}\left\|\nabla f(x_k)\right\|^2 + L\boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k \quad (1 \le k < N). \tag{32}$$

At the last iterate, set

$$V_N = f(y_N) - f(x_\star) - \tau_N L \left\|x_0 - x_\star\right\|^2. \tag{33}$$

For $N \ge 2$, the exact decrement identities are[10]

$$V_{k+1} - V_k = \begin{cases} b_0\mathcal{I}_f(x_\star, x_0) + a_0\mathcal{I}_f(x_0, x_1) + b_1\mathcal{I}_f(x_\star, x_1) - S_0, & k = 0, \\ a_k\mathcal{I}_f(x_k, x_{k+1}) + b_{k+1}\mathcal{I}_f(x_\star, x_{k+1}) - S_k, & 1 \le k < N-1, \\ a_{N-1}\mathcal{I}_f(x_{N-1}, y_N) + b_N\mathcal{I}_f(x_\star, y_N) - S_{N-1}, & k = N-1. \end{cases} \tag{34}$$

Each S_k is the sum of two squares with positive coefficients. Theorem F.3 below specifies the Lyapunov function coefficients and S_k. Its proof verifies the identities (34), and Lemma F.4 establishes the coefficient signs and hence $S_k \ge 0$. Together with the fact that the interpolation inequalities $\mathcal{I}_f$ are nonpositive, (34) therefore gives $V_N \le \cdots \le V_0 = 0$ and consequently shows (4). □

Theorem F.3 (Theorem 5.1 with full details of the Lyapunov function)**.** Let $f \in \mathcal{F}_L$, let $x_\star \in \mathcal{H}$ minimize f, and suppose $\|x_0 - x_\star\| \le R$, where $R \ge 0$. Fix an integer $N \ge 1$. Run FGM from $x_0 = y_0 = z_0$. Define

$$D_k = 2C_N + 1 - C_k \in \mathbb{R} \quad (0 \le k \le N), \qquad \tau_N = \frac{1}{2(2C_N+1)} = \frac{1}{2D_0} \in \mathbb{R}.$$

Set $a_{-1} = 0$ and $a_N = 1$. For $0 \le k < N$, define $a_k \in \mathbb{R}$, and for $0 \le k \le N$, define $b_k \in \mathbb{R}$, by

$$\begin{aligned} a_k &= \frac{\sum_{i=0}^{k} \theta_i(1 - \theta_{i-1}^2/\theta_{N-1}^2)}{D_{k+1}} = \frac{\theta_k^2(\theta_{N-1}^2 - \theta_k^2 + C_{k+1})}{\theta_{N-1}^2 D_{k+1}}, \\ b_k &= a_k - a_{k-1}. \end{aligned} \tag{35}$$

[10]For $N = 1$, only the first line is used, with x_1 replaced by y_1, $a_0 = b_0 = b_1 = 1/2$, and S_0 as specified in the footnote to (43), and $V_1 = V_N$ is given by (33). For $N = 2$, only the first and last lines are used.

For $0 \le k < N$, define the auxiliary scalar $\sigma_k \in \mathbb{R}$ by

$$\sigma_k = \frac{a_k}{\theta_k^2} - \frac{a_{k-1}}{\theta_{k-1}^2}. \tag{36}$$

The second term in σ_0 is defined to be zero. In particular, $a_0 = b_0 = 1/(D_0 - 1)$ and $b_N = 1/(C_N + 1)$. For $1 \le k < N$, define the state $\boldsymbol{w}_k \in \mathcal{H}^2$ and the matrix $\mathbf{Q}_k \in \mathbb{R}^{2\times 2}$ by

$$\boldsymbol{w}_k = \begin{bmatrix} z_{k+1} - x_\star \\ y_{k+1} - z_{k+1} \end{bmatrix}, \qquad \mathbf{Q}_k = \frac{1}{2}\begin{pmatrix} \dfrac{1-a_k}{D_0 - \theta_k^2} & 0 \\ 0 & \dfrac{\theta_k^2/\theta_{N-1}^2 - a_k}{\theta_k^2 - C_{k+1}} \end{pmatrix}. \tag{37}$$

Define the Lyapunov values $V_k \in \mathbb{R}$ as in (32) and (33), *i.e.*, set $V_0 = 0$ and

$$V_k = a_k\big(f(x_k) - f(x_\star)\big) - \tau_N L \left\|x_0 - x_\star\right\|^2 - \frac{a_k}{2L}\left\|\nabla f(x_k)\right\|^2 + L\boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k \quad (1 \le k < N).$$

At the reported point, set

$$V_N = f(y_N) - f(x_\star) - \tau_N L \left\|x_0 - x_\star\right\|^2.$$

For $1 \le k < N-1$, define the vectors $s_k, t_k \in \mathcal{H}$ by

$$\begin{aligned} s_k &= z_{k+1} - x_\star - (D_0 - \theta_k^2)\frac{\nabla f(x_{k+1})}{L}, \\ t_k &= \frac{\nabla f(x_{k+1})}{L} - \frac{y_{k+1} - z_{k+1}}{\theta_k^2 - C_{k+1}}. \end{aligned} \tag{38}$$

For $N \ge 2$, define $s_{N-1}, t_{N-1} \in \mathcal{H}$ by the same formulas with $k = N-1$ and $\nabla f(x_{k+1})$ replaced by $\nabla f(y_N)$.

For $0 \le k < N-1$, define $\alpha_k \in \mathbb{R}$ by

$$\alpha_k = \frac{(D_0 b_{k+1} - \theta_{k+1})(D_0 - \theta_{k+1}^2) + (D_0 a_{k+1} - \theta_{k+1}^2)\theta_{k+1}}{2D_0(D_0 - \theta_{k+1}^2)(D_0 - \theta_k^2)}. \tag{39}$$

For $1 \le k < N-1$, define $\beta_k \in \mathbb{R}$ by

$$\beta_k = \frac{\theta_k^2(\theta_k^2 - C_{k+1})}{2}\left[\sigma_{k+1} + \frac{\theta_{k+1} - 1}{\theta_{k+1}^2 - C_{k+2}}\left(\frac{1}{\theta_{N-1}^2} - \frac{a_{k+1}}{\theta_{k+1}^2}\right)\right]. \tag{40}$$

At the terminal index define $\alpha_{N-1}, \beta_{N-1} \in \mathbb{R}$ by

$$\alpha_{N-1} = \frac{b_N}{2(D_0 - \theta_{N-1}^2)}, \qquad \beta_{N-1} = \frac{b_N(\theta_{N-1}^2 - C_N)}{2}. \tag{41}$$

For $1 \le k < N$, set

$$S_k = L\alpha_k \left\|s_k\right\|^2 + L\beta_k \left\|t_k\right\|^2. \tag{42}$$

The initial term is, for $N \ge 2$,[11]

$$S_0 = \frac{LD_0}{2(D_0 - 1)^2}\left\|\frac{\nabla f(x_0)}{L} - \frac{x_0 - x_\star}{D_0}\right\|^2 + L\alpha_0\left\|z_1 - x_\star - (D_0 - 1)\frac{\nabla f(x_1)}{L}\right\|^2. \tag{43}$$

The square sums satisfy $S_k \ge 0$. For $N \ge 2$, the following decrement identities hold:[12]

$$V_{k+1} - V_k = \begin{cases} b_0\mathcal{I}_f(x_\star, x_0) + a_0\mathcal{I}_f(x_0, x_1) + b_1\mathcal{I}_f(x_\star, x_1) - S_0, & k = 0, \\ a_k\mathcal{I}_f(x_k, x_{k+1}) + b_{k+1}\mathcal{I}_f(x_\star, x_{k+1}) - S_k, & 1 \le k < N-1, \\ a_{N-1}\mathcal{I}_f(x_{N-1}, y_N) + b_N\mathcal{I}_f(x_\star, y_N) - S_{N-1}, & k = N-1. \end{cases}$$

[11] When $N = 1$, only (43) is used: replace x_1 by y_1 and set $D_0 = 3$ and $\alpha_0 = 1/8$. No expression containing $(\theta_0^2 - C_1)^{-1}$ is evaluated.

[12] For $N = 1$, use the first case with x_1 replaced by y_1, $a_0 = b_0 = b_1 = 1/2$, and the convention for S_0 above. For $N = 2$, the middle range is empty.

The coefficient signs in Lemma F.4 and the nonpositive interpolation residuals imply $V_N \le \cdots \le V_0 = 0$. The terminal definition and the radius assumption therefore give

$$f(y_N) - f(x_\star) \le \frac{LR^2}{2(2C_N+1)} = \frac{LR^2\theta_{N-1}^2}{2\big(2\sum_{k=0}^{N-1}\theta_k^3 + \theta_{N-1}^2\big)}. \tag{44}$$

Proof. The companion notebook is `examples_peppy/fgm_conjecture4/fgm_conjecture4_example_lyap.ipynb`. The notebook also documents the smaller prerequisite set for running just the proof checks.

For $1 \le k < N-1$, subtracting two consecutive instances of (32) cancels the common initial-radius term and gives

$$\begin{aligned} V_{k+1} - V_k = {} & a_{k+1}\big(f(x_{k+1}) - f(x_\star)\big) - a_k\big(f(x_k) - f(x_\star)\big) \\ & - \frac{a_{k+1}}{2L}\|\nabla f(x_{k+1})\|^2 + \frac{a_k}{2L}\|\nabla f(x_k)\|^2 \\ & + L\boldsymbol{w}_{k+1}^\top \mathbf{Q}_{k+1}\boldsymbol{w}_{k+1} - L\boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k. \end{aligned} \tag{45}$$

Our goal is to identify this expression with the interior case of (34). We first factor the quadratic remainder, then match the weighted interpolation residuals; the boundary cases are treated separately.

Step 1: factorization of the interior slack. For $1 \le k < N-1$, Steps 1 and 2 prove the second line of (34), namely,

$$V_{k+1} - V_k = a_k \mathcal{I}_f(x_k, x_{k+1}) + b_{k+1}\mathcal{I}_f(x_\star, x_{k+1}) - S_k,$$

with S_k as defined in (42). We first factor a quadratic remainder into the two squares defining S_k; Step 2 identifies this remainder in the Lyapunov difference. For this index, the algorithm gives

$$\begin{aligned} \boldsymbol{w}_{k+1} &= \begin{pmatrix} 1 & 0 \\ 0 & \theta_k^2/\theta_{k+1}^2 \end{pmatrix}\boldsymbol{w}_k - \begin{bmatrix} \theta_{k+1} \\ 1-\theta_{k+1} \end{bmatrix}\frac{\nabla f(x_{k+1})}{L}, \\ x_{k+1} - x_\star &= z_{k+1} - x_\star + \frac{\theta_k^2}{\theta_{k+1}^2}(y_{k+1} - z_{k+1}). \end{aligned} \tag{46}$$

Define the symmetric matrix $\mathbf{S}_k \in \mathbb{R}^{3\times 3}$ by

$$\begin{aligned} \mathbf{S}_k = {} & \begin{pmatrix} \mathbf{Q}_k & -\frac{1}{2}\begin{bmatrix} b_{k+1} \\ \theta_k^2\sigma_{k+1} \end{bmatrix} \\ -\frac{1}{2}\begin{bmatrix} b_{k+1} \\ \theta_k^2\sigma_{k+1} \end{bmatrix}^\top & a_{k+1} \end{pmatrix} \\ & - \begin{pmatrix} 1 & 0 & -\theta_{k+1} \\ 0 & \theta_k^2/\theta_{k+1}^2 & \theta_{k+1}-1 \end{pmatrix}^\top \mathbf{Q}_{k+1} \begin{pmatrix} 1 & 0 & -\theta_{k+1} \\ 0 & \theta_k^2/\theta_{k+1}^2 & \theta_{k+1}-1 \end{pmatrix}. \end{aligned} \tag{47}$$

By (46), the last matrix is the transition matrix. Substituting (37), (39), and (40) gives

$$\mathbf{S}_k = \begin{bmatrix} 1 & 0 \\ 0 & -1/(\theta_k^2 - C_{k+1}) \\ -(D_0 - \theta_k^2) & 1 \end{bmatrix} \begin{pmatrix} \alpha_k & 0 \\ 0 & \beta_k \end{pmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1/(\theta_k^2 - C_{k+1}) \\ -(D_0 - \theta_k^2) & 1 \end{bmatrix}^\top.$$

Evaluating the associated quadratic form and using (38) gives

$$L\begin{bmatrix} \boldsymbol{w}_k \\ \nabla f(x_{k+1})/L \end{bmatrix}^\top \mathbf{S}_k \begin{bmatrix} \boldsymbol{w}_k \\ \nabla f(x_{k+1})/L \end{bmatrix} = L\alpha_k \|s_k\|^2 + L\beta_k \|t_k\|^2 = S_k. \tag{48}$$

This proves the claim. Code Cell 12 checks this factorization coefficient by coefficient. The two pure position terms give the coefficient identities

$$\begin{aligned} \|z_{k+1} - x_\star\|^2 : \quad & (\mathbf{Q}_k)_{11} - (\mathbf{Q}_{k+1})_{11} = \alpha_k, \\ \|y_{k+1} - z_{k+1}\|^2 : \quad & (\mathbf{Q}_k)_{22} - \frac{\theta_k^4}{\theta_{k+1}^4}(\mathbf{Q}_{k+1})_{22} = \frac{\beta_k}{(\theta_k^2 - C_{k+1})^2}. \end{aligned}$$

The interior coefficient calculation after Step 2 checks the two gradient cross terms and the squared gradient norm. The position cross term $\langle z_{k+1} - x_\star, y_{k+1} - z_{k+1} \rangle$ is absent from both sides of (48).

Step 2: verification of the interior decrement identity. It remains to show that the left-hand side of (48) is the difference between the weighted interpolation residuals and $V_{k+1} - V_k$. To match (45), expand the weighted residuals. Since $b_{k+1} = a_{k+1} - a_k$ and $\nabla f(x_\star) = 0$, their function-value terms agree with the potential difference:

$$\begin{aligned} & a_k \mathcal{I}_f(x_k, x_{k+1}) + b_{k+1} \mathcal{I}_f(x_\star, x_{k+1}) \\ & \quad = a_{k+1}\big(f(x_{k+1}) - f(x_\star)\big) - a_k\big(f(x_k) - f(x_\star)\big) \\ & \qquad + \frac{a_k}{2L} \|\nabla f(x_k)\|^2 + \frac{a_{k+1}}{2L} \|\nabla f(x_{k+1})\|^2 \\ & \qquad + \left\langle \nabla f(x_{k+1}), a_k \left(x_k - x_{k+1} - \frac{\nabla f(x_k)}{L} \right) + b_{k+1}(x_\star - x_{k+1}) \right\rangle . \end{aligned} \tag{49}$$

The remaining inner-product coefficient follows from $y_{k+1} = x_k - \nabla f(x_k)/L$ and the mixing update:

$$a_k(y_{k+1} - x_{k+1}) + b_{k+1}(x_\star - x_{k+1}) = -b_{k+1}(z_{k+1} - x_\star) - \theta_k^2 \sigma_{k+1}(y_{k+1} - z_{k+1}). \tag{50}$$

Here $x_{k+1} - x_\star = z_{k+1} - x_\star + \theta_k^2(y_{k+1} - z_{k+1})/\theta_{k+1}^2$, and the definition of σ_{k+1} gives the stated equality.

Subtract the difference of the potentials in (32). The function-value terms and the terms involving $\|\nabla f(x_k)\|^2$ cancel, while the common initial-radius term cancels between the two potentials. The two contributions involving $\|\nabla f(x_{k+1})\|^2$ add to $\frac{a_{k+1}}{L} \|\nabla f(x_{k+1})\|^2$. The remaining position and cross terms are exactly those in (47), so

$$\begin{aligned} & a_k \mathcal{I}_f(x_k, x_{k+1}) + b_{k+1} \mathcal{I}_f(x_\star, x_{k+1}) - (V_{k+1} - V_k) \\ & \quad = L \begin{bmatrix} \boldsymbol{w}_k \\ \nabla f(x_{k+1})/L \end{bmatrix}^\top \mathbf{S}_k \begin{bmatrix} \boldsymbol{w}_k \\ \nabla f(x_{k+1})/L \end{bmatrix} = S_k, \end{aligned} \tag{51}$$

where the last equality is the factorization proved in Step 1. Rearranging proves the second line of (34) for $1 \le k < N - 1$. The base and terminal cases are treated below, followed by the coefficient signs needed to deduce $V_{k+1} \le V_k$. Code Cell 12 checks this interpolation cancellation and the complete decrement, including every Gram, function-value, and constant coefficient.

Terminal index. For $N \ge 2$, this calculation proves the last line of (34) using the endpoint (33) (Code Cell 14). The final gradient step gives $x_{N-1} - x_\star = y_N - x_\star + \nabla f(x_{N-1})/L$. Because $a_{N-1} = 1 - b_N$, the remaining normalized quadratic is

$$\boldsymbol{w}_{N-1}^\top \mathbf{Q}_{N-1} \boldsymbol{w}_{N-1} + \frac{1}{2L^2} \|\nabla f(y_N)\|^2 - \frac{b_N}{L} \langle \nabla f(y_N), y_N - x_\star \rangle .$$

At this index, the position coefficients are

$$\begin{aligned} \|z_N - x_\star\|^2 : \quad & (\mathbf{Q}_{N-1})_{11} = \frac{b_N}{2(D_0 - \theta_{N-1}^2)}, \\ \|y_N - z_N\|^2 : \quad & (\mathbf{Q}_{N-1})_{22} = \frac{b_N}{2(\theta_{N-1}^2 - C_N)}. \end{aligned}$$

The gradient coefficient in the two squares satisfies

$$\frac{1}{L^2} \|\nabla f(y_N)\|^2 : \quad (D_0 - \theta_{N-1}^2)^2 \alpha_{N-1} + \beta_{N-1} = \frac{b_N(C_N + 1)}{2} = \frac{1}{2}.$$

Completing the two squares therefore gives $\alpha_{N-1} \|s_{N-1}\|^2 + \beta_{N-1} \|t_{N-1}\|^2 = S_{N-1}/L$. This proves the terminal line of (34), without any additional terminal square in V_N.

Base index. This calculation settles the first line of (34) for $N \ge 2$ (Code Cell 13).[13] For $N \ge 2$, $x_1 = y_1 = z_1 = x_0 - \nabla f(x_0)/L$ and $\theta_1^2 - \theta_1 = 1$. Substitution of these relations and $a_0 = b_0 = 1/(D_0 - 1)$ into (32) yields

$$b_0 \mathcal{I}_f(x_\star, x_0) + a_0 \mathcal{I}_f(x_0, x_1) + b_1 \mathcal{I}_f(x_\star, x_1) - V_1 = S_0.$$

[13] Code Cell 15 verifies the single base–terminal case for $N = 1$.

Both sides are quadratic in $x_0 - x_\star$, $\nabla f(x_0)/L$, and $\nabla f(x_1)/L$ after their function terms cancel. For $N = 1$, the same expansion uses $a_0 = b_0 = b_1 = 1/2$ and gives directly

$$\begin{aligned}&\tfrac{1}{2}\mathcal{I}_f(x_\star, x_0) + \tfrac{1}{2}\mathcal{I}_f(x_0, y_1) + \tfrac{1}{2}\mathcal{I}_f(x_\star, y_1) - \left(f(y_1) - f(x_\star) - L\left\|x_0 - x_\star\right\|^2/6\right)\\ &\quad= \frac{3L}{8}\left\|\frac{\nabla f(x_0)}{L} - \frac{x_0 - x_\star}{3}\right\|^2 + \frac{L}{8}\left\|y_1 - x_\star - 2\frac{\nabla f(y_1)}{L}\right\|^2.\end{aligned}$$

This verifies the decrement identity directly for $N = 1$.[14]

Positivity of the coefficients. Lemma F.4 proves every required denominator and coefficient sign by scalar identities, checked in Code Cell 10. The denominator bounds make the formulas well-defined; the positive square coefficients give $S_k \geq 0$, and the positive interpolation weights make (34) nonpositive.

Conclusion. Every interpolation residual in (34) is nonpositive and every S_k is nonnegative. Thus $V_N \leq \cdots \leq V_0 = 0$. The terminal definition in (33) and $\|x_0 - x_\star\| \leq R$ give (44). □

Interior coefficient calculation. This verifies the vector cancellation in Step 2 and all coefficients of (48) (Code Cell 12). Use $x_k - \nabla f(x_k)/L = y_{k+1}$ and the mixing step to obtain

$$\begin{aligned}&a_k\left(x_k - x_{k+1} - \frac{\nabla f(x_k)}{L}\right) + b_{k+1}(x_\star - x_{k+1})\\ &\quad= -b_{k+1}(z_{k+1} - x_\star) + \left(a_k - \frac{\theta_k^2}{\theta_{k+1}^2}a_{k+1}\right)(y_{k+1} - z_{k+1})\\ &\quad= -b_{k+1}(z_{k+1} - x_\star) - \theta_k^2\sigma_{k+1}(y_{k+1} - z_{k+1}).\end{aligned}$$

Together with the two position coefficients displayed after (48), that quadratic identity is determined by the following three coefficient comparisons:

$$\begin{aligned}\frac{1}{L}\langle\nabla f(x_{k+1}), z_{k+1} - x_\star\rangle:&\quad 2\theta_{k+1}(\mathbf{Q}_{k+1})_{11} - b_{k+1} = -2(D_0 - \theta_k^2)\alpha_k,\\ \frac{1}{L}\langle\nabla f(x_{k+1}), y_{k+1} - z_{k+1}\rangle:&\quad -\frac{2\theta_k^2(\theta_{k+1} - 1)}{\theta_{k+1}^2}(\mathbf{Q}_{k+1})_{22} - \theta_k^2\sigma_{k+1} = -\frac{2\beta_k}{\theta_k^2 - C_{k+1}},\\ \frac{1}{L^2}\left\|\nabla f(x_{k+1})\right\|^2:&\quad a_{k+1} - \theta_{k+1}^2(\mathbf{Q}_{k+1})_{11} - (\theta_{k+1} - 1)^2(\mathbf{Q}_{k+1})_{22}\\ &\qquad= (D_0 - \theta_k^2)^2\alpha_k + \beta_k.\end{aligned}$$

They follow by substituting the displayed rational coefficients and using $\theta_{k+1}^2 - \theta_{k+1} = \theta_k^2$ and (14). Together with the zero $\langle z_{k+1} - x_\star, y_{k+1} - z_{k+1}\rangle$ coefficient, these five coefficients cover all quadratic terms in $z_{k+1} - x_\star$, $y_{k+1} - z_{k+1}$, and $\nabla f(x_{k+1})/L$, so the verification is independent of N and dimension.

Lemma F.4. The interpolation and square coefficients are positive on their respective index ranges:[15]

$$a_k > 0, \qquad b_k > 0, \qquad \alpha_k > 0, \qquad \beta_k > 0.$$

Proof. Define $\rho_k \in \mathbb{R}$ for this scalar argument by

$$\rho_k = \frac{\sum_{i=0}^{k-1}\theta_i^2}{\theta_k^2}, \qquad \rho_0 = 0.$$

The theta relation gives $\theta_k = \theta_k^2 - \theta_{k-1}^2$ and, by factoring, $\theta_k^4 - \theta_{k-1}^4 = \theta_k(\theta_k^2 + \theta_{k-1}^2) = 2\theta_k^3 - \theta_k^2$. Summing these identities gives

$$C_{k+1} = \frac{\theta_k^2 + \rho_k + 1}{2}, \qquad \rho_k + 1 = 1 + \left(1 - \frac{1}{\theta_k}\right)(\rho_{k-1} + 1). \tag{52}$$

[14] When $N = 2$, the base and terminal calculations already cover every transition.

[15] The unused values a_{-1} and, when $N = 1$, β_0 are zero.

The recurrence is understood for $k \geq 1$. Starting from $\rho_0 + 1 = 1$ and using $\theta_k > \theta_{k-1} + 1/2$, induction yields

$$\rho_k + 1 \leq \frac{2\theta_k + 1}{3}, \qquad \rho_{k-1} + 1 < \frac{2\theta_k}{3}, \qquad \rho_k - \rho_{k-1} > \frac{1}{3} \quad (k \geq 1).$$

Indeed the second inequality follows from the preceding-index upper bound, and inserting it into (52) proves the next upper bound and the increment estimate. Hence C_k increases, $D_k \geq C_N + 1 > 0$, and $D_0 - \theta_k^2 \geq \rho_{N-1} + 2 > 0$ for $k < N$. Moreover $\theta_k^2 - C_{k+1} > 0$ for $k \geq 1$, by (52) and $\rho_k \leq 2(\theta_k - 1)/3$.

- **Positivity of a_k.** The same telescoping identities give

$$\sum_{i=0}^{k} \theta_i \theta_{i-1}^2 = \theta_k^2(\theta_k^2 - C_{k+1}), \qquad \sum_{i=0}^{k} \theta_i = \theta_k^2.$$

  They prove the two formulas for a_k in (35). For $0 \leq k < N$, every summand in the first formula is positive because $\theta_i > 0$ and $\theta_{i-1} < \theta_{N-1}$ for $0 \leq i \leq k$. Together with $D_{k+1} > 0$, this gives $a_k > 0$. At the boundary indices, $a_N = 1$ and $a_{-1} = 0$ by definition. Code Cell 10 checks the telescoping increments and coefficient identities.

  For the remaining coefficient signs, we first show that the auxiliary difference σ_k is positive. At index zero, $\sigma_0 = a_0 > 0$. For $1 \leq k < N$, using $\theta_{N-1} \geq \theta_k$, substitution of (35) into (36) gives the positive expansion

$$\begin{aligned} &\theta_k \theta_{N-1}^2 D_k D_{k+1} \sigma_k \\ &\quad = \frac{\theta_k}{18}\left[(3\theta_k - 4)(\theta_k - 1) + 2(3\theta_k + 1)(\theta_{N-1} - \theta_k) + 6(\theta_{N-1} - \theta_k)^2\right] \\ &\qquad + \left(\frac{2\theta_k}{3} - (\rho_{k-1} + 1)\right)\left[\theta_{N-1}^2 - \theta_k^2 + \frac{\theta_k}{2} + \frac{\theta_{N-1}}{3} + \frac{2}{3}\right] \\ &\qquad + C_k\left(\frac{2\theta_{N-1} + 1}{3} - (\rho_{N-1} + 1)\right). \end{aligned} \tag{53}$$

  By the bounds on ρ_{k-1} and ρ_{N-1} above, every summand is nonnegative, and the first is strictly positive because $\theta_k \geq \theta_1 > 4/3$. The factor $\theta_k \theta_{N-1}^2 D_k D_{k+1}$ is positive, so $\sigma_k > 0$. Code Cell 10 checks (53) on the stated scalar relations.

- **Positivity of b_k.** For $1 \leq k < N$, the theta relation and (36) give

$$b_k = \frac{a_k}{\theta_k} + \theta_{k-1}^2 \sigma_k > 0.$$

  At the boundary indices, $b_0 = a_0 > 0$ and $b_N = 1/(C_N + 1) > 0$. Since $a_N = 1$ and $b_k = a_k - a_{k-1} > 0$, we also have $1 - a_k > 0$ for $0 \leq k < N$.

- **Positivity of α_k.** The inequalities $\sigma_k > 0$ imply that a_k/θ_k^2 increases from $a_0 = 1/(D_0 - 1) > 1/D_0$. Together with the formula for b_k above, this gives

$$D_0 b_k - \theta_k > 0, \qquad D_0 a_k - \theta_k^2 > 0 \quad (0 \leq k < N),$$

  where the signs at $k = 0$ follow from $b_0 = a_0$ and $\theta_0 = 1$. All numerator factors in (39) are therefore positive, as are its denominator factors, so $\alpha_k > 0$ for $0 \leq k < N - 1$. The terminal formula (41) gives $\alpha_{N-1} > 0$ for $N \geq 2$.[16]

- **Positivity of β_k.** For $0 \leq k < N$, the exact tail identity is

$$\frac{1}{\theta_{N-1}^2} - \frac{a_k}{\theta_k^2} = \frac{\rho_{N-1} + 1 - \rho_k}{\theta_{N-1}^2 D_{k+1}} > 0. \tag{54}$$

  Its numerator is at least one because $\rho_k \leq \rho_{N-1}$. Code Cell 10 checks (54) on the same relations. Thus (40) is a sum of positive terms multiplied by a positive factor, proving $\beta_k > 0$ for $1 \leq k < N - 1$. For $N \geq 2$, the terminal formula (41) is positive because $b_N > 0$ and $\theta_{N-1}^2 - C_N > 0$.[17]

□

[16] For $N = 1$, $\alpha_0 = 1/8$, and the other coefficient in S_0 is $3/8$.

[17] For $N = 1$, the unused value β_0 is zero.

F.3 Proof of Theorem 5.2 (Conjecture 5 in Taylor et al. (2017c))

F.3.1 Input prompt

```
/pep-implement

Function: f is convex and L-smooth
Parameters: L, R
Initial condition: ||x_0 - x_star|| <= R, where grad f(x_star) = 0
Performance metric: f(x_N) - f(x_star)
Algorithm: Fast gradient method with fixed step size 1/L

Parameter sequence: theta_0 = 1 and theta_{k+1} = (1 + sqrt(1 + 4 *
    theta_k^2)) / 2 for k >= 0
Initialization: x_0 = y_0 = z_0
For k = 0,...,N-1:
y_{k+1} = x_k - (1/L) * grad f(x_k)
z_{k+1} = z_k - (theta_k/L) * grad f(x_k)
x_{k+1} = (1 - 1/theta_{k+1}) * y_{k+1} + (1/theta_{k+1}) * z_{k+1}

Conjectured rate: f(x_N) - f(x_star) <= L * R^2 / (2 * (2 * D_N + 1)),
    where D_N = sum_{j=0}^{N-1} h_{N,j} and x_i = x_0 - (1/L) * sum_{j
    =0}^{i-1} h_{i,j} * grad f(x_j)
```

Then, we repeatedly entered:

```
proceed with the next step
```

The initial proof of Conjecture 5 was algebraically involved. We therefore asked the agent to revise it using the two-square decomposition in the FGM-rational proof (Appendix F.1), with the additional prompt below. This led to the simpler two-square proof presented below.

```
Could you explore whether allowing two squared-norm terms per
iteration in the SOS decomposition, as in
fgm_rational_tight_example_lyap.ipynb, can simplify the proof of FGM
Conjecture 5?
```

Proof outline for Theorem 5.2. Write $f_\star = f(x_\star) \in \mathbb{R}$, and set $D_j = 2C_{N+1} - 1 - C_j \in \mathbb{R}$ and $\tau_N = 1/(2D_0) \in \mathbb{R}$. Define the initial and interior potentials by $V_0 = 0$ and

$$\begin{aligned} V_k = a_{k-1}(f(x_k) - f_\star) - \tau_N L \|x_0 - x_\star\|^2 - \frac{a_{k-1}}{2L} \|\nabla f(x_k)\|^2 \\ + b_k \langle \nabla f(x_k), y_{k+1} - x_\star \rangle + L \boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k, \qquad 1 \le k < N. \end{aligned} \tag{55}$$

At the output point, use the terminal potential

$$V_N = a_{N-1}(f(x_N) - f_\star) - \tau_N L \|x_0 - x_\star\|^2 + b_N \left(\langle \nabla f(x_N), x_N - x_\star \rangle - \frac{1}{2L} \|\nabla f(x_N)\|^2 \right). \tag{56}$$

For $N \ge 2$, the exact decrement identities are[18]

$$V_{k+1} - V_k = \begin{cases} a_0 \big[\mathcal{I}_f(x_\star, x_0) + \mathcal{I}_f(x_0, x_1) \big] - S_0, & k = 0, \\ b_k \mathcal{I}_f(x_\star, x_k) + a_k \mathcal{I}_f(x_k, x_{k+1}) - S_k, & 1 \le k \le N-2, \\ b_{N-1} \mathcal{I}_f(x_\star, x_{N-1}) + a_{N-1} \mathcal{I}_f(x_{N-1}, x_N) - S_{N-1}, & k = N-1. \end{cases} \tag{57}$$

[18] For $N = 2$, the middle range is empty. For $N = 1$, the first line alone holds with $a_0 = 1/2$, V_1 given by (56), and S_0 given by (66); no interior matrix is used.

The full theorem and proof below verify the identities, and Lemma F.6 proves positivity of every used interpolation and square coefficient. Thus $S_k \geq 0$ and $\mathcal{I}_f \leq 0$ imply $V_N \leq \cdots \leq V_0 = 0$. The endpoint identity is

$$f(x_N) - f_\star - \tau_N L R^2 = V_N + b_N \mathcal{I}_f(x_\star, x_N) + \tau_N L\big(\|x_0 - x_\star\|^2 - R^2\big). \tag{58}$$

Since $b_N, \tau_N > 0$, the stationary-point residual and the radius assumption make the last two terms nonpositive. This gives (5). □

Theorem F.5 (Theorem 5.2 with full details of the Lyapunov function)**.** Let $f \in \mathcal{F}_L$, let $x_\star \in \mathcal{H}$ minimize f, write $f_\star = f(x_\star) \in \mathbb{R}$, and suppose $\|x_0 - x_\star\| \leq R$, where $R \geq 0$. For a fixed integer $N \geq 1$, run FGM from $x_0 = y_0 = z_0$. For $0 \leq k \leq N$, define

$$D_k = 2C_{N+1} - 1 - C_k \in \mathbb{R}, \qquad \tau_N = \frac{1}{2(2C_{N+1} - 1)} = \frac{1}{2D_0} \in \mathbb{R}.$$

Here D_k denotes a gap variable, not a fixed-step row sum. For $0 \leq k < N$, define $h_{N,k}, m_k \in \mathbb{R}$ by

$$h_{N,k} = \theta_k \left(1 - \frac{\theta_{k-1}^2}{\theta_N^2}\right), \qquad m_k = \frac{\theta_k - 1}{\theta_{k+1}}. \tag{59}$$

Set $a_{-1} = 0$ and $a_N = 1$. Define the chain weights $a_k \in \mathbb{R}$ for $0 \leq k < N$ and the star weights $b_k \in \mathbb{R}$ for $0 \leq k \leq N$ by

$$a_k = \frac{\sum_{j=0}^{k} h_{N,j}}{D_{k+1}}, \qquad b_k = a_k - a_{k-1}. \tag{60}$$

In particular, $a_0 = b_0 = 1/D_1$.

For $1 \leq k < N$, define the state $\boldsymbol{w}_k \in \mathcal{H}^2$ and the symmetric matrix $\mathbf{Q}_k \in \mathbb{R}^{2\times 2}$ by

$$\boldsymbol{w}_k = \begin{bmatrix} y_k - x_\star \\ y_{k+1} - x_\star \end{bmatrix}, \qquad \mathbf{Q}_k = \frac{1}{2}\begin{pmatrix} \dfrac{1 - h_{N,k} + D_{k+1}(a_{k-1} + a_k/D_k)}{D_k} & -a_{k-1} - \dfrac{a_k}{D_k} \\ -a_{k-1} - \dfrac{a_k}{D_k} & a_k \end{pmatrix}. \tag{61}$$

Define the Lyapunov values $V_k \in \mathbb{R}$ as in (55) and (56), *i.e.*, set $V_0 = 0$ and

$$\begin{aligned} V_k = {} & a_{k-1}(f(x_k) - f(x_\star)) - \tau_N L \|x_0 - x_\star\|^2 - \frac{a_{k-1}}{2L}\|\nabla f(x_k)\|^2 \\ & + b_k \langle \nabla f(x_k), y_{k+1} - x_\star \rangle + L \boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k, \qquad 1 \leq k < N. \end{aligned}$$

At the output point, set

$$\begin{aligned} V_N = {} & a_{N-1}(f(x_N) - f(x_\star)) - \tau_N L \|x_0 - x_\star\|^2 \\ & + b_N \left(\langle \nabla f(x_N), x_N - x_\star \rangle - \frac{1}{2L}\|\nabla f(x_N)\|^2\right). \end{aligned}$$

Here $\boldsymbol{w}^\top \mathbf{Q} \boldsymbol{w}$ denotes the corresponding scalar sum of inner products. The gradient at x_N is a proof witness and does not change the algorithm.[19]

For $1 \leq k \leq N-2$, define the square coefficients $\alpha_k, \beta_k \in \mathbb{R}$ by

$$\begin{aligned} \alpha_k &= \frac{a_{k+1}}{2}, \\ \beta_k &= \frac{1}{2}\big[a_k D_{k+1}^2 + (a_k - 1)D_{k+1} - (h_{N,k} - 1)(D_k + D_{k+1}) - a_{k+1} m_k^2 D_k^2\big]. \end{aligned} \tag{62}$$

With the vectors $s_k, t_k \in \mathcal{H}$ defined by

$$s_k = \frac{\nabla f(x_{k+1})}{L} - \frac{y_{k+1} - x_\star}{D_{k+1}}, \qquad t_k = \frac{y_k - x_\star}{D_k} - \frac{y_{k+1} - x_\star}{D_{k+1}}, \tag{63}$$

[19] For $N = 1$, only V_0 and the terminal definition are used.

set

$$S_k = L\alpha_k \|s_k\|^2 + L\beta_k \|t_k\|^2 .$$

For $N \geq 2$, define the terminal coefficients $\alpha_{N-1}, \beta_{N-1} \in \mathbb{R}$, directions $s_{N-1}, t_{N-1} \in \mathcal{H}$, and square sum S_{N-1} by

$$\begin{aligned} \alpha_{N-1} &= \frac{1}{2}, \qquad \beta_{N-1} = \frac{a_{N-1} - (m_{N-1} + b_N)^2}{2}, \\ s_{N-1} &= \frac{\nabla f(x_N)}{L} - (x_N - x_\star) + a_{N-1}(y_N - x_\star), \\ t_{N-1} &= y_N - x_\star - \frac{D_N}{D_{N-1}}(y_{N-1} - x_\star), \\ S_{N-1} &= L\alpha_{N-1} \|s_{N-1}\|^2 + L\beta_{N-1} \|t_{N-1}\|^2 . \end{aligned} \tag{64}$$

For $N \geq 2$, the initial square sum is

$$S_0 = \frac{L}{2D_0} \left\| x_0 - x_\star - \frac{D_0}{D_1}(x_1 - x_\star) \right\|^2 + \frac{La_1}{2} \left\| \frac{\nabla f(x_1)}{L} - \frac{x_1 - x_\star}{D_1} \right\|^2 . \tag{65}$$

For $N = 1$, define the initial square sum separately by

$$S_0 = \frac{3L}{8} \left\| \frac{\nabla f(x_0)}{L} - \frac{x_0 - x_\star}{3} \right\|^2 + \frac{L}{2} \left\| \frac{\nabla f(x_1)}{L} - \frac{x_1 - x_\star}{2} \right\|^2 . \tag{66}$$

The square sums satisfy $S_k \geq 0$. For $N \geq 2$, the following decrement identities hold:[20]

$$V_{k+1} - V_k = \begin{cases} a_0 \big[\mathcal{I}_f(x_\star, x_0) + \mathcal{I}_f(x_0, x_1)\big] - S_0, & k = 0, \\ b_k \mathcal{I}_f(x_\star, x_k) + a_k \mathcal{I}_f(x_k, x_{k+1}) - S_k, & 1 \leq k \leq N-2, \\ b_{N-1} \mathcal{I}_f(x_\star, x_{N-1}) + a_{N-1} \mathcal{I}_f(x_{N-1}, x_N) - S_{N-1}, & k = N-1. \end{cases}$$

The coefficient signs in Lemma F.6 and $\mathcal{I}_f \leq 0$ imply $V_N \leq \cdots \leq V_0 = 0$. Since $a_{N-1} + b_N = 1$, the terminal definition gives

$$f(x_N) - f(x_\star) = V_N + b_N \mathcal{I}_f(x_\star, x_N) + \tau_N L \|x_0 - x_\star\|^2 \leq \tau_N L R^2.$$

Consequently,

$$f(x_N) - f_\star \leq \frac{LR^2}{2(2C_{N+1} - 1)} = \frac{LR^2 \theta_N^2}{2\big(2\sum_{j=0}^{N} \theta_j^3 - \theta_N^2\big)}. \tag{67}$$

Proof. The companion notebook is `examples_peppy/fgm_conjecture5/fgm_conjecture5_example_lyap.ipynb`. The notebook also documents the prerequisites for running its exact proof checks separately.

For $1 \leq k \leq N-2$, subtracting two consecutive instances of (55) cancels the common initial-radius term and gives

$$\begin{aligned} V_{k+1} - V_k = {} & a_k\big(f(x_{k+1}) - f(x_\star)\big) - a_{k-1}\big(f(x_k) - f(x_\star)\big) \\ & - \frac{a_k}{2L} \|\nabla f(x_{k+1})\|^2 + \frac{a_{k-1}}{2L} \|\nabla f(x_k)\|^2 \\ & + b_{k+1} \langle \nabla f(x_{k+1}), y_{k+2} - x_\star \rangle - b_k \langle \nabla f(x_k), y_{k+1} - x_\star \rangle \\ & + L \boldsymbol{w}_{k+1}^\top \mathbf{Q}_{k+1} \boldsymbol{w}_{k+1} - L \boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k. \end{aligned} \tag{68}$$

Our goal is to identify this expression with the interior case of (57). We first factor the quadratic remainder, then match the weighted interpolation residuals; the boundary cases are treated separately.

Step 1: factorization of the interior slack. For $1 \leq k \leq N-2$, Steps 1 and 2 prove the second line of (57), namely,

$$V_{k+1} - V_k = b_k \mathcal{I}_f(x_\star, x_k) + a_k \mathcal{I}_f(x_k, x_{k+1}) - S_k,$$

[20] For $N = 1$, the first case holds with $a_0 = 1/2$, the terminal definition of V_1, and (66). For $N = 2$, the middle range is empty.

with S_k defined by (62)–(63). We first factor a quadratic remainder into its two squares; Step 2 identifies that remainder by subtracting the potential difference from the interpolation residuals. Fix $1 \le k \le N-2$. Define the symmetric matrix $\mathbf{S}_k \in \mathbb{R}^{3\times 3}$ by

$$\mathbf{S}_k = \begin{pmatrix} \mathbf{Q}_k & \frac{1}{2}\begin{bmatrix} m_k a_{k+1} \\ -a_k m_k - b_{k+1}(1+m_k) \end{bmatrix} \\ \frac{1}{2}\begin{bmatrix} m_k a_{k+1} \\ -a_k m_k - b_{k+1}(1+m_k) \end{bmatrix}^\top & a_{k+1} \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 \\ -m_k & 1+m_k & -1 \end{pmatrix}^\top \mathbf{Q}_{k+1} \begin{pmatrix} 0 & 1 & 0 \\ -m_k & 1+m_k & -1 \end{pmatrix}. \tag{69}$$

Eliminating $z_k = \theta_k x_k - (\theta_k - 1) y_k$ from the updates gives

$$x_{k+1} = (1+m_k) y_{k+1} - m_k y_k, \qquad 0 \le k < N. \tag{70}$$

Together with $y_{k+2} = x_{k+1} - \nabla f(x_{k+1})/L$, this gives

$$\boldsymbol{w}_{k+1} = \begin{pmatrix} 0 & 1 & 0 \\ -m_k & 1+m_k & -1 \end{pmatrix} \begin{bmatrix} \boldsymbol{w}_k \\ \nabla f(x_{k+1})/L \end{bmatrix}.$$

Unrolling this momentum form to x_N gives the final-row relations

$$h_{N,k} = 1 + m_k h_{N,k+1} \quad (0 \le k < N-1), \qquad h_{N,N-1} = 1 + m_{N-1}. \tag{71}$$

The formula in (59) satisfies these relations by (12), so it gives the final row of the fixed-step representation in the common FGM lemma. Substituting (14) and (12) gives $\delta_{k+1} = 1 + m_k \delta_k$ for $0 \le k < N$. Thus the definitions of D_k and a_k give

$$\begin{gathered} D_k - D_{k+1} = \delta_k, \qquad D_{k+2} = D_{k+1} - 1 - m_k \delta_k, \\ a_k D_{k+1} - a_{k-1} D_k = h_{N,k}. \end{gathered} \tag{72}$$

The second identity is used for $0 \le k < N-1$; the others hold for $0 \le k < N$. Code Cell 29 checks these scalar identities. These relations suffice for the following coefficient comparison; the denominator signs are proved independently in Lemma F.6. The position matrices satisfy

$$\begin{aligned} &\boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k - \begin{bmatrix} y_{k+1} - x_\star \\ x_{k+1} - x_\star \end{bmatrix}^\top \mathbf{Q}_{k+1} \begin{bmatrix} y_{k+1} - x_\star \\ x_{k+1} - x_\star \end{bmatrix} \\ &\quad = \frac{a_{k+1}}{2D_{k+1}^2} \left\| y_{k+1} - x_\star \right\|^2 + \beta_k \left\| \frac{y_k - x_\star}{D_k} - \frac{y_{k+1} - x_\star}{D_{k+1}} \right\|^2. \end{aligned} \tag{73}$$

Using (70) to expand $x_{k+1} - x_\star = (1+m_k)(y_{k+1} - x_\star) - m_k(y_k - x_\star)$, the three position coefficients satisfy

$$\begin{aligned} \|y_k - x_\star\|^2 : &\quad (\mathbf{Q}_k)_{11} - m_k^2 (\mathbf{Q}_{k+1})_{22} = \frac{\beta_k}{D_k^2}, \\ 2\langle y_k - x_\star, y_{k+1} - x_\star \rangle : &\quad (\mathbf{Q}_k)_{12} + m_k (\mathbf{Q}_{k+1})_{12} + m_k(1+m_k)(\mathbf{Q}_{k+1})_{22} = -\frac{\beta_k}{D_k D_{k+1}}, \\ \|y_{k+1} - x_\star\|^2 : &\quad (\mathbf{Q}_k)_{22} - (\mathbf{Q}_{k+1})_{11} - 2(1+m_k)(\mathbf{Q}_{k+1})_{12} \\ &\qquad - (1+m_k)^2 (\mathbf{Q}_{k+1})_{22} = \frac{a_{k+1} + 2\beta_k}{2D_{k+1}^2}. \end{aligned}$$

Substitution of (71) and (72) verifies all three identities; Code Cell 30 checks the coefficients of (73).

Using $a_{k+1} = a_k + b_{k+1}$ and $2(\mathbf{Q}_{k+1})_{22} = a_{k+1}$, the gradient coefficients are

$$\begin{aligned} \frac{1}{L^2} \|\nabla f(x_{k+1})\|^2 : &\quad a_{k+1} - (\mathbf{Q}_{k+1})_{22} = \frac{a_{k+1}}{2}, \\ \frac{1}{L} \langle \nabla f(x_{k+1}), y_k - x_\star \rangle : &\quad m_k \big(a_{k+1} - 2(\mathbf{Q}_{k+1})_{22} \big) = 0, \\ \frac{1}{L} \langle \nabla f(x_{k+1}), y_{k+1} - x_\star \rangle : &\quad a_k + 2(\mathbf{Q}_{k+1})_{12} = -\frac{a_{k+1}}{D_{k+1}}. \end{aligned}$$

Together with (73), these coefficient identities give

$$\mathbf{S}_k = \alpha_k \begin{bmatrix} 0 \\ -1/D_{k+1} \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ -1/D_{k+1} \\ 1 \end{bmatrix}^\top + \beta_k \begin{bmatrix} 1/D_k \\ -1/D_{k+1} \\ 0 \end{bmatrix} \begin{bmatrix} 1/D_k \\ -1/D_{k+1} \\ 0 \end{bmatrix}^\top .$$

Evaluating the associated quadratic form and using (63) gives

$$L \begin{bmatrix} \boldsymbol{w}_k \\ \nabla f(x_{k+1})/L \end{bmatrix}^\top \mathbf{S}_k \begin{bmatrix} \boldsymbol{w}_k \\ \nabla f(x_{k+1})/L \end{bmatrix} = L\alpha_k \|s_k\|^2 + L\beta_k \|t_k\|^2 = S_k, \tag{74}$$

as claimed. Code Cell 30 checks all six quadratic coefficients in $y_k - x_\star$, $y_{k+1} - x_\star$, and $\nabla f(x_{k+1})$.

Step 2: verification of the interior decrement identity. It remains to show that the left-hand side of (74) is the difference between $b_k \mathcal{I}_f(x_\star, x_k) + a_k \mathcal{I}_f(x_k, x_{k+1})$ and $V_{k+1} - V_k$. To match (68), first compare the function values. The identity $b_k = a_k - a_{k-1}$ gives the coefficients a_k and $-a_{k-1}$ of $f(x_{k+1}) - f(x_\star)$ and $f(x_k) - f(x_\star)$. For the gradient terms, the update and (70) give

$$x_k - x_\star = y_{k+1} - x_\star + \frac{\nabla f(x_k)}{L}, \qquad x_k - x_{k+1} - \frac{\nabla f(x_k)}{L} = m_k(y_k - y_{k+1}).$$

Expanding the residuals using these relations yields

$$\begin{aligned} b_k \mathcal{I}_f(x_\star, x_k) + a_k \mathcal{I}_f(x_k, x_{k+1}) &= a_k\big(f(x_{k+1}) - f(x_\star)\big) - a_{k-1}\big(f(x_k) - f(x_\star)\big) \\ &+ \frac{a_{k-1}}{2L} \|\nabla f(x_k)\|^2 + \frac{a_k}{2L} \|\nabla f(x_{k+1})\|^2 \\ &- b_k \langle \nabla f(x_k), y_{k+1} - x_\star \rangle + a_k m_k \langle \nabla f(x_{k+1}), y_k - y_{k+1} \rangle . \end{aligned} \tag{75}$$

Subtracting (68) cancels the function values and the terms involving $\nabla f(x_k)$. The two next-gradient norm terms add to $(a_k/L)\|\nabla f(x_{k+1})\|^2$; the remaining position and inner-product terms give the left-hand side of (74). Thus

$$\begin{aligned} &b_k \mathcal{I}_f(x_\star, x_k) + a_k \mathcal{I}_f(x_k, x_{k+1}) - (V_{k+1} - V_k) \\ &\quad = L \begin{bmatrix} \boldsymbol{w}_k \\ \nabla f(x_{k+1})/L \end{bmatrix}^\top \mathbf{S}_k \begin{bmatrix} \boldsymbol{w}_k \\ \nabla f(x_{k+1})/L \end{bmatrix} = S_k, \end{aligned} \tag{76}$$

where the last equality is the factorization proved in Step 1. Rearranging proves the second line of (57) for $1 \le k \le N-2$. Code Cell 36 checks this bridge and the complete decrement in every quadratic, function-value, and constant coefficient. The base and terminal cases are treated next; the coefficient signs needed for monotonicity are established after them.

Terminal index. Assume $N \ge 2$. This verifies the last line of (57) using (56); Code Cell 37 checks both the position factorization and the full decrement. Let $m = m_{N-1} \in \mathbb{R}$. The matrix gives

$$\begin{aligned} \boldsymbol{w}_{N-1}^\top \mathbf{Q}_{N-1} \boldsymbol{w}_{N-1} = \frac{1}{2} &\|m(y_{N-1} - x_\star) - (m + b_N)(y_N - x_\star)\|^2 \\ &+ \beta_{N-1} \|t_{N-1}\|^2 . \end{aligned} \tag{77}$$

The common FGM lemma gives $\sum_{k=0}^{N-1} h_{N,k} = C_{N+1} - 1$, so (60) yields $b_N = \delta_N / D_N$. Together with $h_{N,N-1} = 1 + m_{N-1}$ and $\delta_N = 1 + m_{N-1}\delta_{N-1}$, this gives the identities

$$\mathbf{Q}_{N-1} \begin{bmatrix} D_{N-1} \\ D_N \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 - h_{N,N-1} \\ h_{N,N-1} - a_{N-1} \end{bmatrix}, \qquad mD_{N-1} - (m + b_N)D_N = -1.$$

Both sides of (77) have this action on the indicated vector and the same $\|y_N - x_\star\|^2$ coefficient, $a_{N-1}/2$. Since $D_{N-1} > 0$, these facts determine the three matrix entries uniquely. Using $x_N - x_\star = (1+m)(y_N - x_\star) - m(y_{N-1} - x_\star)$ and $b_N = 1 - a_{N-1}$, expanding the two final interpolation residuals converts the first square in (77) into $\|s_{N-1}\|^2$ and gives (64). This proves the final decrement.

Base index. This verifies the first line of (57) for $N \ge 2$ (Code Cell 38).[21] For $N \ge 2$, we have $x_1 = y_1 = z_1$, $D_0 = D_1 + 1$, $D_2 = D_1 - 1$, $a_0 = 1/D_1$, and $h_{N,1} = a_1 D_2 - 1$. Direct expansion

[21] Together with the terminal calculation, this also settles every transition when $N = 2$.

gives

$$\begin{aligned}&\mathcal{I}_f(x_\star, x_0) + \mathcal{I}_f(x_0, x_1)\\ &\quad = f(x_1) - f_\star + L\|x_1 - x_\star\|^2 - L\langle x_0 - x_\star, x_1 - x_\star\rangle + \frac{1}{2L}\|\nabla f(x_1)\|^2.\end{aligned}$$

Substituting (61) in V_1 and subtracting gives precisely S_0 in (65), proving the base identity.

Positivity of the coefficients. Lemma F.6 proves all required coefficient signs and denominator bounds directly from the scalar definitions. The positive square coefficients give $S_k \geq 0$, and the positive interpolation weights make (57) nonpositive.

Conclusion. For $N \geq 2$, the decrements in (57) are nonpositive, so $V_N \leq V_0 = 0$. Using (58), checked in Code Cell 39, proves (67).

Small horizons. For $N = 1$, $D_0 = 3$, $a_0 = b_0 = b_1 = 1/2$, and $x_1 = x_0 - \nabla f(x_0)/L$.[22] The terminal definition directly satisfies

$$\begin{aligned}V_1 = \tfrac{1}{2}\big[\mathcal{I}_f(x_\star, x_0) + \mathcal{I}_f(x_0, x_1)\big]\\ - \frac{3L}{8}\left\|\frac{\nabla f(x_0)}{L} - \frac{x_0 - x_\star}{3}\right\|^2 - \frac{L}{2}\left\|\frac{\nabla f(x_1)}{L} - \frac{x_1 - x_\star}{2}\right\|^2.\end{aligned}$$

Together with (58), this proves $LR^2/6$ without an interior matrix. No step divides by R.

□

Lemma F.6. The interpolation and square coefficients are positive on their respective index ranges:[23]

$$a_k > 0, \qquad b_k > 0, \qquad \alpha_k > 0, \qquad \beta_k > 0.$$

The initial square coefficients in (65) and (66) are also positive.

Proof. For $0 \leq k \leq N$, define the scalar $\rho_k \in \mathbb{R}$ by

$$\rho_k = \frac{1}{\theta_k^2}\sum_{i=0}^{k-1}\theta_i^2, \qquad \rho_0 = 0.$$

Its recurrence follows directly from this definition. Telescoping $\theta_k = \theta_k^2 - \theta_{k-1}^2$ and $2\theta_k^3 = \theta_k^4 - \theta_{k-1}^4 + \theta_k^2$ gives

$$\begin{aligned}\rho_k &= \left(1 - \frac{1}{\theta_k}\right)(\rho_{k-1} + 1) \quad (k \geq 1),\\ C_{k+1} &= \frac{\theta_k^2 + \rho_k + 1}{2},\\ \sum_{i=0}^{k} h_{N,i} &= \frac{\theta_k^2(2\theta_N^2 - \theta_k^2 + \rho_k + 1)}{2\theta_N^2} \quad (k < N).\end{aligned} \tag{78}$$

The last identity uses the final-row formula (59). The theta equation gives $1/2 < \theta_k - \theta_{k-1} < 1$ for $k \geq 1$. Starting from equality at $k = 0$, induction in the ρ recurrence yields

$$\frac{\theta_k - 1}{2} \leq \rho_k \leq \frac{2(\theta_k - 1)}{3}, \qquad 0 \leq k \leq N. \tag{79}$$

Indeed, the lower and upper induction steps reduce, respectively, to the positive quantities

$$\frac{(\theta_k - 1)(\theta_{k-1} + 1 - \theta_k)}{2\theta_k} > 0, \qquad \frac{(\theta_k - 1)(2\theta_k - 2\theta_{k-1} - 1)}{3\theta_k} > 0.$$

Moreover, $\rho_{k-1} + 1 \leq (2\theta_{k-1} + 1)/3 \leq \theta_{k-1} < \theta_k$, so

$$\rho_k - \rho_{k-1} = 1 - \frac{\rho_{k-1} + 1}{\theta_k} > 0 \qquad (k \geq 1).$$

[22] Code Cells 40–41 verify the single base–terminal decrement and endpoint normalization using (56) and (66) directly.

[23] The unused value a_{-1} is zero; no α_k, β_k is needed at index zero.

In particular $\rho_k > 0$ for $k \geq 1$. Subtracting adjacent moment identities gives

$$\delta_0 = 1, \qquad \delta_k = \frac{1}{2}\left(\theta_k + 1 - \frac{\rho_k}{\theta_k - 1}\right) > 0 \quad (1 \leq k \leq N), \\ D_k = D_{k+1} + \delta_k \quad (0 \leq k < N). \tag{80}$$

Here positivity follows from (79). The moment formula gives $C_{N+1} > 1$, so $D_N = C_{N+1} - 1 + \delta_N > 0$ and hence $D_k \geq D_N > 0$ for $0 \leq k \leq N$. Finally, (59) gives $h_{N,k} > 0$ for $0 \leq k < N$, because $\theta_{k-1}^2 < \theta_N^2$. Code Cell 29 checks the moment identities; the preceding induction proves the bounds used below.

- **Positivity of a_k.** For $0 \leq k < N$, all $h_{N,j}$ in the numerator of (60) are positive, and $D_{k+1} > 0$. Thus $a_k > 0$; also $a_N = 1$.
- **Positivity of b_k.** Direct subtraction in (60) gives

  $$b_k = \frac{h_{N,k} + a_{k-1}\delta_k}{D_{k+1}} > 0 \quad (1 \leq k < N), \qquad b_0 = a_0 > 0.$$

  The row-sum identity in Lemma F.11 gives $a_{N-1} = (C_{N+1}-1)/D_N$, so $b_N = 1 - a_{N-1} = \delta_N/D_N > 0$. Code Cell 29 checks these coefficient identities.
- **Positivity of α_k.** For $1 \leq k \leq N-2$, (62) gives $\alpha_k = a_{k+1}/2 > 0$. For $N \geq 2$, the terminal value is $\alpha_{N-1} = 1/2$. The initial square coefficients $L/(2D_0)$ and $La_1/2$ in (65) are also positive.[24]
- **Positivity of β_k.**

  Terminal coefficient. For $N \geq 2$, the preceding formulas give

  $$b_N = \frac{\delta_N}{D_N}, \qquad a_{N-1} = \frac{C_{N+1} - 1}{D_N}.$$

  The moment identities and (80) give $D_N = \rho_N + \theta_N \delta_N$. Using $m_{N-1} = (\theta_{N-1} - 1)/\theta_N$ and $\theta_{N-1}^2 = \theta_N^2 - \theta_N$, expansion yields

  $$a_{N-1} - (m_{N-1} + b_N)^2 = \frac{\rho_N\big((2\theta_{N-1} + \theta_N)D_N - \rho_N\big)}{\theta_N^2 D_N^2} > 0. \tag{81}$$

  Indeed, $\rho_N, D_N, \delta_N > 0$, and

  $$(2\theta_{N-1} + \theta_N)D_N - \rho_N = (2\theta_{N-1} + \theta_N - 1)D_N + \theta_N\delta_N > 0.$$

  Thus (64) gives $\beta_{N-1} > 0$. Code Cell 32 verifies (81) and its positive-factor identities.[25]

  Interior coefficient. The original proof of interior $\beta_k > 0$ involved lengthy polynomial-table calculations. We therefore gave the agent the following additional prompt to obtain the simpler analytic regrouping proof below.

  > In fgm_conjecture5_example_lyap.ipynb, find a simpler analytic proof that $\beta_k > 0$ for all $N \geq 3$ and $1 \leq k \leq N-2$, starting from its current direct definition and regrouping terms using D_k, D_{k+1}, and $\delta_k := D_k - D_{k+1}$. Preserve the existing coefficient definitions and normalization, avoid the large polynomial tables, justify any sequence bounds and denominator signs used, and verify the resulting identities exactly with SymPy. Return a concise proof and executable verification code, clearly identifying any remaining gap.

[24] For $N = 1$, the coefficients in (66) are $3L/8$ and $L/2$, both positive.

[25] The factorization also holds for $N = 1$, where the scalar difference is $1/4$, although its square coefficient is not used in the one-step formula.

Fix $N \geq 3$ and $1 \leq k \leq N-2$. We first strengthen the sequence bounds used above:

$$\begin{gathered}\frac{1}{2} < \theta_j - \theta_{j-1} < \frac{2}{3}, \qquad \frac{3}{5}(\theta_j - 1) < \rho_j < \frac{2}{3}(\theta_j - 1), \\ \rho_j - \rho_{j-1} > \frac{1}{3}, \qquad 1 \leq j \leq N.\end{gathered} \tag{82}$$

For $1 \leq j \leq N$, the recurrence gives $\theta_j^2 - \theta_j - \theta_{j-1}^2 = 0$ with $\theta_j > 0$. Since $\theta_{j-1} \geq 1$,

$$\begin{aligned}\left(\theta_{j-1} + \frac{1}{2}\right)^2 - \left(\theta_{j-1} + \frac{1}{2}\right) - \theta_{j-1}^2 &= -\frac{1}{4} < 0, \\ \left(\theta_{j-1} + \frac{2}{3}\right)^2 - \left(\theta_{j-1} + \frac{2}{3}\right) - \theta_{j-1}^2 &= \frac{\theta_{j-1}}{3} - \frac{2}{9} > 0.\end{aligned}$$

The positive root θ_j therefore lies between $\theta_{j-1} + 1/2$ and $\theta_{j-1} + 2/3$, proving the first bounds. Starting from $\theta_0 = 1$ and $\rho_0 = 0$, the recurrence in (78) transports the non-strict bounds at index $j-1$ to

$$\begin{aligned}\rho_j - \frac{3}{5}(\theta_j - 1) &\geq \frac{\theta_j - 1}{5\theta_j}\big(2 - 3(\theta_j - \theta_{j-1})\big) > 0, \\ \frac{2}{3}(\theta_j - 1) - \rho_j &\geq \frac{\theta_j - 1}{3\theta_j}\big(2(\theta_j - \theta_{j-1}) - 1\big) > 0.\end{aligned}$$

Both margins are positive since $\theta_j > 1$ and $1/2 < \theta_j - \theta_{j-1} < 2/3$. This proves the strict bounds by induction, and

$$\rho_j - \rho_{j-1} = 1 - \frac{\rho_{j-1} + 1}{\theta_j} \geq \frac{1}{3} + \frac{2(\theta_j - \theta_{j-1}) - 1}{3\theta_j} > \frac{1}{3}.$$

For the fixed k, we have $\theta_k > 1$, $\theta_{k+1}^2 - \theta_{k+1} = \theta_k^2$, and $m_k = (\theta_k - 1)/\theta_{k+1} \in (0, 1)$. Using $\delta_k = D_k - D_{k+1} = C_{k+1} - C_k$, the moment and gap identities give

$$\begin{aligned}\delta_k &= \frac{1}{2}\left(\theta_k + 1 - \frac{\rho_k}{\theta_k - 1}\right) > 0, \qquad \delta_{k+1} = 1 + m_k\delta_k > 1, \\ C_{k+1} &= \theta_k^2 - (\theta_k - 1)\delta_k, \qquad D_k = D_{k+1} + \delta_k, \\ D_{k+1} &= \theta_N^2 + \rho_N - C_{k+1}, \qquad D_{k+2} = D_{k+1} - \delta_{k+1}.\end{aligned} \tag{83}$$

Since the C_j are strictly increasing and $k + 2 \leq N$,

$$D_{k+2} = 2C_{N+1} - 1 - C_{k+2} \geq C_{k+2} - 1 = C_{k+1} + \delta_{k+1} - 1 > C_{k+1} > 0.$$

In particular, $D_k, D_{k+1}, D_{k+2}, \theta_N^2, \theta_{k+1}, \theta_k - 1, D_0$ are positive. The moment identities and (59)–(60) yield

$$\begin{aligned}a_k D_{k+1} &= \theta_k^2\left(1 - \frac{(\theta_k - 1)\delta_k}{\theta_N^2}\right), \\ a_k D_{k+1} + (1 - h_{N,k})\delta_k &= C_{k+1}, \\ m_k^2 a_{k+1} D_{k+2} &= (\theta_k - 1)^2\left(1 - \frac{(\theta_{k+1} - 1)\delta_{k+1}}{\theta_N^2}\right).\end{aligned} \tag{84}$$

Code Cell 31 verifies these three identities. The direct definition (62) becomes

$$2\beta_k = a_k D_{k+1}^2 + (a_k - 1)D_{k+1} - (h_{N,k} - 1)(D_k + D_{k+1}) - a_{k+1}m_k^2 D_k^2.$$

Define the real scalars $\chi_k, \varphi_k, \psi_k \in \mathbb{R}$ by

$$\begin{aligned}\chi_k &= (m_k + 2)(\rho_k + 1) - \delta_k - (\theta_k - 1), \\ \varphi_k &= \frac{\theta_k - 1}{\theta_N^2}\big[\chi_k D_{k+2} - (\theta_k - 1)(1 + \rho_{k+1} - \rho_N)(2\delta_k + \delta_{k+1})\big], \\ \psi_k &= D_{k+2}\big(C_{k+1} - m_k^2 a_{k+1}\delta_k^2\big).\end{aligned}$$

Substituting (83)–(84) and using the recurrences for θ_k and ρ_k gives

$$2\beta_k D_{k+2} = D_{k+1}\varphi_k + \psi_k. \tag{85}$$

Code Cell 31 checks this regrouping from the direct definition.

Since $(\theta_{k+1}-1)\delta_{k+1} = \theta_{k+1}^2 - C_{k+2} \in (0, \theta_{k+1}^2)$ and $\theta_N^2 \geq \theta_{k+1}^2$, we have $0 < m_k^2 a_{k+1} D_{k+2} < (\theta_k - 1)^2$. Also $C_{k+1} - (\theta_k - 1)\delta_k = \rho_k + 1 > 0$ and $C_{k+1} + (\theta_k - 1)\delta_k = \theta_k^2$. Hence

$$\psi_k > C_{k+1}^2 - (\theta_k - 1)^2 \delta_k^2 = \theta_k^2(\rho_k + 1) > 0.$$

For φ_k, the sequence bounds imply

$$\delta_k < \frac{\theta_k}{2} + \frac{1}{5}, \qquad C_{k+1} > \frac{\theta_k^2}{2} + \frac{3\theta_k}{10} + \frac{1}{5}.$$

Using $\rho_k > 3(\theta_k - 1)/5$ and the bound on δ_k gives

$$\begin{aligned}\chi_k &> m_k\left(\frac{3\theta_k}{5} + \frac{2}{5}\right) - \frac{3\theta_k}{10} + \frac{8}{5} \\ &= \frac{3\theta_k}{10} + 1 + \frac{3(\theta_k - 1)}{5\theta_{k+1}}\left(\theta_k + \frac{2}{3} - \theta_{k+1}\right) > \frac{3\theta_k}{10} + 1 > 0,\end{aligned}$$

because $\theta_{k+1} - \theta_k < 2/3$. The rearrangement in this bound is checked in Code Cell 31. Since $N \geq k+2$, we have $\rho_N - \rho_{k+1} > 1/3$, so $1 + \rho_{k+1} - \rho_N < 2/3$; this quantity need not be positive. Using $\theta_k - 1, 2\delta_k + \delta_{k+1} > 0$ and $2\delta_k + \delta_{k+1} < 3\delta_k + 1 < 3\theta_k/2 + 8/5$, we obtain

$$\begin{aligned}(\theta_k - 1)(1 + \rho_{k+1} - \rho_N)(2\delta_k + \delta_{k+1}) &< \frac{2}{3}(\theta_k - 1)(2\delta_k + \delta_{k+1}) \\ &< (\theta_k - 1)\left(\theta_k + \frac{16}{15}\right).\end{aligned}$$

Together with $D_{k+2} > C_{k+1} > 0$ and the lower bounds above, this proves

$$\begin{aligned}&\chi_k D_{k+2} - (\theta_k - 1)(1 + \rho_{k+1} - \rho_N)(2\delta_k + \delta_{k+1}) \\ &\quad > \left(\frac{3\theta_k}{10} + 1\right)\left(\frac{\theta_k^2}{2} + \frac{3\theta_k}{10} + \frac{1}{5}\right) - (\theta_k - 1)\left(\theta_k + \frac{16}{15}\right) \\ &\quad = \frac{45\theta_k(\theta_k - 41/30)^2 + (79/20)\theta_k + 380}{300} > 0.\end{aligned} \tag{86}$$

Code Cell 31 verifies the final equality. Thus $\varphi_k, \psi_k > 0$, and (85), with $D_{k+1}, D_{k+2} > 0$, proves $\beta_k > 0$ throughout the interior range. The induction and inequalities above establish the signs for all such N, k.

□

F.4 Proof of Theorem 5.3 (Conjecture 4 in Taylor et al. (2017c))

F.4.1 Input prompt

```
/pep-implement

Function: f is convex and L-smooth
Parameters: L, R
Initial condition: ||x_0 - x_star|| <= R, where grad f(x_star) = 0
Performance metric: f(y_N) - f(x_star)
Algorithm: Optimized gradient method with fixed step size 1/L

Parameter sequence: theta_0 = 1 and theta_{k+1} = (1 + sqrt(1 + 4 *
    theta_k^2)) / 2 for k = 0,...,N-2
```

```
Initialization: x_0 = y_0 = z_0
For k = 0,...,N-2:
y_{k+1} = x_k - (1/L) * grad f(x_k)
z_{k+1} = z_k - (2 * theta_k/L) * grad f(x_k)
x_{k+1} = (1 - 1/theta_{k+1}) * y_{k+1} + (1/theta_{k+1}) * z_{k+1}

Final step:
y_N = x_{N-1} - (1/L) * grad f(x_{N-1})
z_N = z_{N-1} - (2 * theta_{N-1}/L) * grad f(x_{N-1})

Reported point: y_N

Conjectured rate: f(y_N) - f(x_star) <= L * R^2 / (2 * (2 * theta_{N
    -1}^2 + 1))
```

Then, we repeatedly entered:

```
proceed with the next step
```

Proof outline for Theorem 5.3. Write $f_\star = f(x_\star) \in \mathbb{R}$ and set $\tau_N = 1/[2(2\theta_{N-1}^2+1)] \in \mathbb{R}$. Define the potentials by $V_{-1} = 0$ and

$$V_k = a_{k+1}\big(f(x_k) - f_\star\big) - \tau_N L \|x_0 - x_\star\|^2 - \frac{a_{k+1}}{2L}\|\nabla f(x_k)\|^2 + L\boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k, \quad 0 \le k < N. \tag{87}$$

At the reported point, use

$$V_N = f(y_N) - f_\star - \tau_N L \|x_0 - x_\star\|^2. \tag{88}$$

The exact decrement identities are[26]

$$V_k - V_{k-1} = \begin{cases} a_1 \mathcal{I}_f(x_\star, x_0) - S_0, & k = 0, \\ b_{k+1}\mathcal{I}_f(x_\star, x_k) + a_k \mathcal{I}_f(x_{k-1}, x_k) - S_k, & 1 \le k < N, \\ b_{N+1}\mathcal{I}_f(x_\star, y_N) + a_N \mathcal{I}_f(x_{N-1}, y_N) - S_N, & k = N. \end{cases} \tag{89}$$

The full proof establishes these identities, and Lemma F.8 proves positivity of every used interpolation and square coefficient. Since $\mathcal{I}_f \le 0$ and $S_k \ge 0$, they imply $V_N \le \cdots \le V_{-1} = 0$. Finally,

$$f(y_N) - f_\star - \tau_N L R^2 = V_N + \tau_N L\big(\|x_0 - x_\star\|^2 - R^2\big) \le 0, \tag{90}$$

where the last term is nonpositive by the radius assumption. This proves (6). □

Theorem F.7 (Theorem 5.3 with full details of the Lyapunov function)**.** Let $f \in \mathcal{F}_L$, let $x_\star \in \mathcal{H}$ minimize f, and write $f_\star = f(x_\star) \in \mathbb{R}$. Suppose that $\|x_0 - x_\star\| \le R$, with $R \ge 0$. For a fixed integer horizon $N \ge 1$, run the OGM iteration in (OGM) and set

$$\tau_N = \frac{1}{2(2\theta_{N-1}^2 + 1)} \in \mathbb{R}.$$

Define the real scalar coefficients $a_k, b_k \in \mathbb{R}$ by

$$a_0 = 0, \qquad a_k = \frac{\theta_{k-1}^2(2\theta_{N-1}^2 - \theta_{k-1}^2)}{\theta_{N-1}^2(2\theta_{N-1}^2 - \theta_{k-1}^2 + 1)} \quad (1 \le k \le N), \qquad a_{N+1} = 1,$$
$$b_k = a_k - a_{k-1}, \qquad 1 \le k \le N+1. \tag{91}$$

In particular,

$$b_{N+1} = \frac{1}{\theta_{N-1}^2 + 1}.$$

[26] For $N = 1$, the middle range is empty; the base and terminal lines give the two transitions $V_{-1} \to V_0 \to V_1$, with the same boundary formulas.

For $0 \le k < N$, define the state

$$\boldsymbol{w}_k = \begin{bmatrix} z_{k+1} - x_\star \\ y_{k+1} - x_\star \end{bmatrix} \in \mathcal{H}^2 \tag{92}$$

and the symmetric matrix $\mathbf{Q}_k \in \mathbb{R}^{2\times 2}$ by

$$\mathbf{Q}_k = \frac{1}{4\theta_{N-1}^2(2\theta_{N-1}^2+1-\theta_k^2)^3} \begin{pmatrix} (2\theta_{N-1}^2-\theta_k^2)(2\theta_{N-1}^2+1-\theta_k^2)^2 & -\theta_k^2(2\theta_{N-1}^2+1-\theta_k^2) \\ -\theta_k^2(2\theta_{N-1}^2+1-\theta_k^2) & \theta_k^2(6\theta_{N-1}^2-4\theta_k^2+3) \end{pmatrix}. \tag{93}$$

Define the Lyapunov values $V_k \in \mathbb{R}$ as in (87) and (88), *i.e.*, set $V_{-1} = 0$ and

$$\begin{aligned} V_k = {} & a_{k+1}\big(f(x_k) - f(x_\star)\big) - \tau_N L \left\|x_0 - x_\star\right\|^2 - \frac{a_{k+1}}{2L}\left\|\nabla f(x_k)\right\|^2 \\ & + L\boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k, \quad 0 \le k < N. \end{aligned}$$

At the reported point, set

$$V_N = f(y_N) - f(x_\star) - \tau_N L \left\|x_0 - x_\star\right\|^2.$$

The quadratic form denotes the corresponding sum of inner products of the vector components of $\boldsymbol{w}_k$.

For $1 \le k < N$, define the scalars $P_k, \alpha_k, \beta_k, \rho_k \in \mathbb{R}$ by

$$\begin{aligned} P_k &= (8\theta_k - 6)(\theta_{N-1}^2 - \theta_k^2) + 4(\theta_k - 1)^3 + 10(\theta_k - 1)^2 + 12(\theta_k - 1) + 3, \\ \alpha_k &= \frac{\theta_k^2 P_k}{4\theta_{N-1}^2(2\theta_{N-1}^2 + 1 - \theta_k^2)^3}, \\ \beta_k &= \frac{3\theta_k(\theta_k - 1)}{4\theta_{N-1}^2(2\theta_{N-1}^2 + 1 - \theta_{k-1}^2)P_k}, \qquad \rho_k = \frac{2\theta_{N-1}^2 + 1 - 2\theta_{k-1}^2}{2\theta_{N-1}^2 + 1 - \theta_{k-1}^2}. \end{aligned} \tag{94}$$

Define $M_k \in \mathbb{R}^{2\times 2}$, $\mathbf{c}_k, \mathbf{d}_k \in \mathbb{R}^2$, and the square directions $s_k, t_k \in \mathcal{H}$, together with S_k, for $1 \le k < N$ by

$$\begin{aligned} M_k &= \begin{pmatrix} 1 & 0 \\ 1/\theta_k & (\theta_k - 1)/\theta_k \end{pmatrix}, \\ \mathbf{c}_k &= \begin{bmatrix} 2\theta_k \\ 1 \end{bmatrix}, \qquad \mathbf{d}_k = \frac{a_{k+1}}{2}\begin{bmatrix} 1/\theta_k \\ (\theta_k - 1)/\theta_k \end{bmatrix} - \frac{a_k}{2}\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \end{aligned} \tag{95}$$

$$\begin{aligned} s_k &= \frac{\nabla f(x_k)}{L} + \frac{(-\mathbf{d}_k + M_k^\top \mathbf{Q}_k \mathbf{c}_k)^\top \boldsymbol{w}_{k-1}}{\alpha_k}, \quad t_k = z_k - x_\star - \rho_k(y_k - x_\star), \\ S_k &= L\alpha_k \left\|s_k\right\|^2 + L\beta_k \left\|t_k\right\|^2. \end{aligned} \tag{96}$$

The boundary square terms are

$$\begin{aligned} S_0 &= \frac{L}{32\theta_{N-1}^8(2\theta_{N-1}^2+1)} \left\| x_0 - x_\star - \frac{2\theta_{N-1}^2+1}{L}\nabla f(x_0) \right\|^2, \\ S_N &= \frac{1}{2L}\left\| \nabla f(y_N) - \frac{L}{\theta_{N-1}^2+1}(y_N - x_\star) \right\|^2 + \frac{L}{4(\theta_{N-1}^2+1)} \left\| z_N - x_\star - \frac{y_N - x_\star}{\theta_{N-1}^2+1} \right\|^2. \end{aligned} \tag{97}$$

The square sums satisfy $S_k \ge 0$, and the following decrement identities hold:[27]

$$V_k - V_{k-1} = \begin{cases} a_1 \mathcal{I}_f(x_\star, x_0) - S_0, & k = 0, \\ b_{k+1}\mathcal{I}_f(x_\star, x_k) + a_k \mathcal{I}_f(x_{k-1}, x_k) - S_k, & 1 \le k < N, \\ b_{N+1}\mathcal{I}_f(x_\star, y_N) + a_N \mathcal{I}_f(x_{N-1}, y_N) - S_N, & k = N. \end{cases}$$

The coefficient signs in Lemma F.8 and the nonpositive interpolation residuals imply $V_N \le \cdots \le V_{-1} = 0$. The terminal definition and the radius assumption therefore give

$$f(y_N) - f_\star \le \tau_N L \left\|x_0 - x_\star\right\|^2 \le \frac{LR^2}{2(2\theta_{N-1}^2+1)}.$$

[27] When $N = 1$, the middle range is empty and the two boundary cases give $V_{-1} \to V_0 \to V_1$.

Proof. The companion notebook is `examples_peppy/ogm_conjecture4/ogm_conjecture4_example_lyap.ipynb`. The notebook's auxiliary vector $\theta_k x_k - (\theta_k - 1)y_k - x_\star$ equals $z_k - x_\star$ (Code Cell 25), so its momentum form uses the same proof state.

For $1 \le k < N$, subtracting two consecutive instances of (87) cancels the common initial-radius term and gives

$$\begin{aligned} V_k - V_{k-1} = a_{k+1}\big(f(x_k) - f(x_\star)\big) - a_k\big(f(x_{k-1}) - f(x_\star)\big) \\ - \frac{a_{k+1}}{2L}\left\|\nabla f(x_k)\right\|^2 + \frac{a_k}{2L}\left\|\nabla f(x_{k-1})\right\|^2 \\ + L\boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k - L\boldsymbol{w}_{k-1}^\top \mathbf{Q}_{k-1}\boldsymbol{w}_{k-1}. \end{aligned} \tag{98}$$

Our goal is to identify this expression with the interior case of (89). We first factor the quadratic remainder, then match the weighted interpolation residuals; the boundary cases are treated separately.

Step 1: factorization of the interior slack. For $1 \le k < N$, Steps 1 and 2 prove the second line of (89), namely,

$$V_k - V_{k-1} = b_{k+1}\mathcal{I}_f(x_\star, x_k) + a_k\mathcal{I}_f(x_{k-1}, x_k) - S_k.$$

Step 1 factors a quadratic remainder into the squares in (96); Step 2 identifies that remainder by subtracting the potential difference from the weighted interpolation residuals. For $1 \le k < N$, the OGM updates and the mixing identity for x_k give

$$\boldsymbol{w}_k = M_k \boldsymbol{w}_{k-1} - \mathbf{c}_k \frac{\nabla f(x_k)}{L}, \qquad x_k - x_\star = \frac{(\boldsymbol{w}_{k-1})_1 + (\theta_k - 1)(\boldsymbol{w}_{k-1})_2}{\theta_k}. \tag{99}$$

Define the symmetric matrix $\mathbf{S}_k \in \mathbb{R}^{3\times 3}$ by

$$\mathbf{S}_k = \begin{pmatrix} \mathbf{Q}_{k-1} - M_k^\top \mathbf{Q}_k M_k & -\mathbf{d}_k + M_k^\top \mathbf{Q}_k \mathbf{c}_k \\ (-\mathbf{d}_k + M_k^\top \mathbf{Q}_k \mathbf{c}_k)^\top & a_{k+1} - \mathbf{c}_k^\top \mathbf{Q}_k \mathbf{c}_k \end{pmatrix}. \tag{100}$$

The claim in this step is the quadratic-form identity (102). Direct substitution of (93) and (95), using $\theta_{k-1}^2 = \theta_k^2 - \theta_k$, gives

$$\begin{aligned} &\frac{1}{L^2}\left\|\nabla f(x_k)\right\|^2 : \quad a_{k+1} - \mathbf{c}_k^\top \mathbf{Q}_k \mathbf{c}_k = \alpha_k, \\ &\boldsymbol{w}_{k-1}^\top(\cdot)\boldsymbol{w}_{k-1} : \quad \mathbf{Q}_{k-1} - M_k^\top \mathbf{Q}_k M_k = \frac{(-\mathbf{d}_k + M_k^\top \mathbf{Q}_k \mathbf{c}_k)(-\mathbf{d}_k + M_k^\top \mathbf{Q}_k \mathbf{c}_k)^\top}{\alpha_k} \\ &\qquad\qquad + \beta_k \begin{bmatrix} 1 \\ -\rho_k \end{bmatrix}\begin{bmatrix} 1 \\ -\rho_k \end{bmatrix}^\top. \end{aligned} \tag{101}$$

These are rational identities in θ_k and θ_{N-1}. For the first identity, the numerator after collecting terms is $\theta_k^2 P_k$; expanding the entries of the second identity verifies its four scalar entries. Code Cell 42 verifies both identities against the literal formulas and every entry of the resulting 3-by-3 factorization. Evaluating that factorization and using the definitions of s_k, t_k in (96) gives

$$L\begin{bmatrix} \boldsymbol{w}_{k-1} \\ \nabla f(x_k)/L \end{bmatrix}^\top \mathbf{S}_k \begin{bmatrix} \boldsymbol{w}_{k-1} \\ \nabla f(x_k)/L \end{bmatrix} = L\alpha_k \left\|s_k\right\|^2 + L\beta_k \left\|t_k\right\|^2 = S_k, \tag{102}$$

as claimed.

Step 2: verification of the interior decrement identity. For $1 \le k < N$, it remains to identify the left-hand side of (102) with the weighted interpolation residuals minus $V_k - V_{k-1}$. To match (98), use $b_{k+1} = a_{k+1} - a_k$ and $\nabla f(x_\star) = 0$ to expand the residuals:

$$\begin{aligned} &b_{k+1}\mathcal{I}_f(x_\star, x_k) + a_k\mathcal{I}_f(x_{k-1}, x_k) \\ &\quad = a_{k+1}\big(f(x_k) - f(x_\star)\big) - a_k\big(f(x_{k-1}) - f(x_\star)\big) \\ &\qquad + \frac{a_k}{2L}\left\|\nabla f(x_{k-1})\right\|^2 + \frac{a_{k+1}}{2L}\left\|\nabla f(x_k)\right\|^2 \\ &\qquad + \left\langle \nabla f(x_k), a_k\left(x_{k-1} - x_k - \frac{\nabla f(x_{k-1})}{L}\right) + b_{k+1}(x_\star - x_k)\right\rangle. \end{aligned} \tag{103}$$

The function-value terms already agree. The remaining vector coefficient is determined by the gradient step and the mixing identity in (99):

$$\begin{aligned} & a_k\left(x_{k-1}-x_k-\frac{\nabla f(x_{k-1})}{L}\right)+b_{k+1}(x_\star-x_k) \\ & \quad = a_k(y_k-x_\star)-a_{k+1}(x_k-x_\star) \\ & \quad = a_k(\boldsymbol{w}_{k-1})_2-\frac{a_{k+1}}{\theta_k}\left[(\boldsymbol{w}_{k-1})_1+(\theta_k-1)(\boldsymbol{w}_{k-1})_2\right]=-2\mathbf{d}_k^\top \boldsymbol{w}_{k-1}. \end{aligned} \tag{104}$$

Subtracting (98) cancels the function values and the old-gradient norm. The current-gradient terms give $(a_{k+1}/L)\,\|\nabla f(x_k)\|^2-2\left\langle \nabla f(x_k),\mathbf{d}_k^\top \boldsymbol{w}_{k-1}\right\rangle$. Substituting (99) in the remaining terms gives

$$\begin{aligned} & b_{k+1}\mathcal{I}_f(x_\star,x_k)+a_k\mathcal{I}_f(x_{k-1},x_k)-(V_k-V_{k-1}) \\ & \quad = L\begin{bmatrix}\boldsymbol{w}_{k-1}\\ \nabla f(x_k)/L\end{bmatrix}^\top \mathbf{S}_k\begin{bmatrix}\boldsymbol{w}_{k-1}\\ \nabla f(x_k)/L\end{bmatrix}=S_k, \end{aligned} \tag{105}$$

where the last equality is the factorization proved in Step 1. Rearranging proves the second line of (89) for every $1\le k<N$, including $k=1$ whenever this range is nonempty. Code Cell 44 checks the bridge and full decrement in every Gram, function-value, and constant coordinate. The remaining base and terminal cases follow next; positivity is established in Lemma F.8.

Terminal index. This verifies the last line of (89), using (88) and the terminal square sum in (97), for every $N\ge 1$. The coefficient definitions give

$$a_N=\frac{\theta_{N-1}^2}{\theta_{N-1}^2+1},\qquad b_{N+1}=\frac{1}{\theta_{N-1}^2+1}.$$

Using $y_N=x_{N-1}-\nabla f(x_{N-1})/L$ cancels the old gradient terms in the terminal interpolation residuals. Completing the new gradient square in S_N leaves the exact position identity

$$\mathbf{Q}_{N-1}-\begin{pmatrix}0 & 0\\ 0 & 1/[2(\theta_{N-1}^2+1)^2]\end{pmatrix}=\frac{1}{4(\theta_{N-1}^2+1)}\begin{bmatrix}1\\ -1/(\theta_{N-1}^2+1)\end{bmatrix}\begin{bmatrix}1\\ -1/(\theta_{N-1}^2+1)\end{bmatrix}^\top. \tag{106}$$

Code Cell 42 verifies this position factorization, and Code Cell 46 checks the complete terminal decrement and endpoint normalization.

Base index. This verifies the first line of (89), with $V_{-1}=0$ and the initial square in (97).[28] Since

$$\boldsymbol{w}_0=\begin{pmatrix}1 & -2\\ 1 & -1\end{pmatrix}\begin{bmatrix}x_0-x_\star\\ \nabla f(x_0)/L\end{bmatrix},$$

direct substitution gives

$$\begin{aligned} & \begin{pmatrix}\tau_N & -a_1/2\\ -a_1/2 & a_1\end{pmatrix}-\begin{pmatrix}1 & 1\\ -2 & -1\end{pmatrix}\mathbf{Q}_0\begin{pmatrix}1 & -2\\ 1 & -1\end{pmatrix} \\ & \quad =\frac{1}{32\theta_{N-1}^8(2\theta_{N-1}^2+1)}\begin{bmatrix}1\\ -(2\theta_{N-1}^2+1)\end{bmatrix}\begin{bmatrix}1\\ -(2\theta_{N-1}^2+1)\end{bmatrix}^\top. \end{aligned} \tag{107}$$

Expanding $a_1\mathcal{I}_f(x_\star,x_0)-V_0$ produces the quadratic form on the left multiplied by L, and hence equals S_0. This proves the base line of (89). Code Cell 42 checks the matrix factorization, and Code Cell 45 checks the full base decrement.

Positivity of the coefficients. Lemma F.8 proves the required coefficient signs and denominator bounds. Its positive square coefficients give $S_k\ge 0$ at the interior and boundary indices.

[28] The same base calculation applies when $N=1$.

Conclusion. Each interpolation residual in (89) is nonpositive. The coefficient signs from Lemma F.8 and the square decompositions above therefore give

$$V_N \le V_{N-1} \le \cdots \le V_0 \le V_{-1} = 0.$$

Using (90), checked in Code Cell 46, yields[29]

$$f(y_N) - f_\star \le \tau_N L \left\|x_0 - x_\star\right\|^2 \le \frac{LR^2}{2(2\theta_{N-1}^2 + 1)}.$$

□

Lemma F.8. The coefficients are positive on their respective index ranges:[30]

$$a_k > 0, \qquad b_k > 0, \qquad \alpha_k > 0, \qquad \beta_k > 0.$$

The three boundary-square coefficients in (97) are also positive.

Proof. For $0 \le k < N$, $\theta_{N-1}^2 \ge \theta_k^2 \ge 1$, so

$$2\theta_{N-1}^2 + 1 - \theta_k^2 \ge \theta_{N-1}^2 + 1 > 0.$$

For $1 \le k < N$, $\theta_k > 1$. Every term in the formula for P_k in (94) is nonnegative, and its constant term is positive. Hence $P_k > 0$. Code Cell 42 verifies the derivative, denominator, and polynomial identities used below.

- **Positivity of a_k.** For $1 \le k \le N$, $2\theta_{N-1}^2 - \theta_{k-1}^2 \ge \theta_{N-1}^2 > 0$, so (91) gives $a_k > 0$. Also, $a_{N+1} = 1$.
- **Positivity of b_k.** Define $p : [0, \theta_{N-1}^2] \to \mathbb{R}$ by

  $$p(t) = \frac{t(2\theta_{N-1}^2 - t)}{\theta_{N-1}^2(2\theta_{N-1}^2 + 1 - t)}.$$

  Then

  $$p'(t) = \frac{1 - (2\theta_{N-1}^2 + 1)/(2\theta_{N-1}^2 + 1 - t)^2}{\theta_{N-1}^2} > 0.$$

  Indeed, $2\theta_{N-1}^2 + 1 - t \ge \theta_{N-1}^2 + 1$, while $(\theta_{N-1}^2 + 1)^2 - (2\theta_{N-1}^2 + 1) = \theta_{N-1}^4 > 0$. Since $p(0) = a_0 = 0$, $a_k = p(\theta_{k-1}^2)$ for $1 \le k \le N$, and θ_k strictly increases, $b_k = a_k - a_{k-1} > 0$ through $k = N$. At the terminal index, $b_{N+1} = 1/(\theta_{N-1}^2 + 1) > 0$.
- **Positivity of α_k.** For $1 \le k < N$, the numerator $\theta_k^2 P_k$ and denominator in (94) are positive, so $\alpha_k > 0$.
- **Positivity of β_k.** For $1 \le k < N$, $3\theta_k(\theta_k - 1) > 0$, and every denominator factor in (94) is positive. Thus $\beta_k > 0$.

Finally, $L > 0$ and $\theta_{N-1}^2 \ge 1$ make all three coefficients in (97) positive.[31] □

[29] When $N = 1$, there is no interior transition; the base and terminal identities above remain valid and give the same conclusion. Code Cells 45–46 also check these two transitions on the shared one-step trajectory, where $\theta_0 = 1$ and $\tau_1 = 1/6$.

[30] The unused value a_0 is zero.

[31] This also includes $N = 1$.

F.5 Proof of Theorem 5.5 (nonsmooth FPGM1 conjecture in Table 1 of Taylor et al. (2017a))

F.5.1 Input prompt

```
/pep-implement

Function: F = f + g, where f is convex and L-smooth and g is closed,
    proper, and convex
Parameters: L, R
Initial condition: ||x_0 - x_star|| <= R, where 0 in grad f(x_star) +
    partial g(x_star)
Performance metric: F(y_N) - F(x_star)
Algorithm: FISTA/FPGM1 with fixed step size 1/L and inertial parameter
    k/(k+3)

Initialization: y_0 = x_0
For k >= 0:
y_{k+1} = prox_{g/L}(x_k - (1/L) * grad f(x_k))
x_{k+1} = y_{k+1} + (k/(k+3)) * (y_{k+1} - y_k)

Conjectured rate: F(y_N) - F(x_star) <= 2 * L * R^2 / (N^2 + 5*N + 2)
```

Then, we repeatedly entered:

```
proceed with the next step
```

F.5.2 Proof outline

Define $V_k \in \mathbb{R}$ by $V_0 = 0$ and

$$V_N = F(y_N) - F(x_\star) - \tau_N L \left\|x_0 - x_\star\right\|^2, \tag{108}$$

and, for $1 \le k < N$,

$$\begin{aligned} V_k = a_k\big[f(x_k) - f(x_\star)\big] + \widetilde{a}_k\big[g(y_k) - g(x_\star)\big] - \tau_N L \left\|x_0 - x_\star\right\|^2 \\ - 2\left\langle \nabla f(x_k), \mathbf{d}_k^\top \boldsymbol{w}_k \right\rangle + L \boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k. \end{aligned} \tag{109}$$

Thus the interior checkpoint uses the smooth oracle at x_k and the proximal point y_k. No value of $g(y_0)$ is needed. To include the terminal step, define $\widehat{x}_{k+1} \in \mathcal{H}$ by

$$\widehat{x}_{k+1} = \begin{cases} x_{k+1}, & 0 \le k < N-1, \\ y_N, & k = N-1. \end{cases}$$

For notational convenience, define $\mathcal{T}_k \in \mathbb{R}$ using the nonpositive interpolation residuals defined below:

$$\begin{aligned} \mathcal{T}_k = a_k \mathcal{I}_f(x_k, \widehat{x}_{k+1}) + b_{k+1} \mathcal{I}_f(x_\star, \widehat{x}_{k+1}) \\ + \widetilde{a}_k \mathcal{I}_g(y_k, y_{k+1}) + \widetilde{b}_{k+1} \mathcal{I}_g(x_\star, y_{k+1}), \qquad 0 \le k < N, \end{aligned} \tag{110}$$

where the term with $\widetilde{a}_0 = 0$ is omitted, and define $S_k \in \mathbb{R}$ by

$$S_k = \frac{L a_k}{2} \left\|s_k\right\|^2 + \frac{b_{k+1}}{2L} \left\|\nabla f(\widehat{x}_{k+1}) - \nabla f(x_\star)\right\|^2 + L\alpha_k \left\|Y_{k+1} - Y_k\right\|^2. \tag{111}$$

For $N \ge 2$, the proof establishes the exact decrements

$$V_{k+1} - V_k = \begin{cases} \mathcal{T}_0 - S_0 + b_0 \mathcal{I}_f(x_\star, x_0) - \dfrac{b_0}{2L} \left\|\nabla f(x_0) - \nabla f(x_\star)\right\|^2, & k = 0, \\ \mathcal{T}_k - S_k, & 1 \le k < N-1, \\ \mathcal{T}_{N-1} - S_{N-1}, & k = N-1. \end{cases} \tag{112}$$

For $N = 1$, the initial and terminal conventions apply to the same step:

$$V_1 - V_0 = \mathcal{T}_0 - S_0 + b_0 \mathcal{I}_f(x_\star, x_0) - \frac{b_0}{2L} \|\nabla f(x_0) - \nabla f(x_\star)\|^2, \qquad \widehat{x}_1 = y_1. \tag{113}$$

Lemma F.10 gives nonnegative interpolation weights and $S_k \geq 0$, so these identities imply $V_N \leq \cdots \leq V_0 = 0$. The endpoint identity

$$F(y_N) - F(x_\star) - \tau_N L R^2 = V_N + \tau_N L\big(\|x_0 - x_\star\|^2 - R^2\big) \leq 0 \tag{114}$$

then gives the claimed bound, including $R = 0$.

F.5.3 Full proof

Theorem F.9 (Theorem 5.5 with full details of the Lyapunov function)**.** Let $L > 0$, let $f \in \mathcal{F}_L$, and let $g : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ be proper, closed, and convex, and set $F = f + g : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$. Fix an integer $N \geq 1$ and $R \geq 0$, and suppose that $0 \in \nabla f(x_\star) + \partial g(x_\star)$ and $\|x_0 - x_\star\| \leq R$. Let the iterates be generated by FISTA-rational. For this fixed horizon, set

$$\tau_N = \frac{2}{N^2 + 5N + 2} \in \mathbb{R},$$

and use the coefficients $a_k, b_k \in \mathbb{R}$ from (19)–(20). Thus, as in Theorem F.1, a_k weights the smooth function value and $b_k = a_k - a_{k-1}$. For the nonsmooth function value and $1 \leq k \leq N$, define the scalars $D_k, \widetilde{a}_k, \widetilde{b}_k \in \mathbb{R}$ by

$$D_0 = \frac{N^2 + 5N + 2}{4}, \qquad D_k = \frac{2N^2 + 10N + 2 - k^2 - 5k}{8},$$
$$\widetilde{a}_0 = 0, \qquad \widetilde{a}_k = \frac{D_k + 1}{D_k} a_{k-1}, \qquad \widetilde{b}_k = \widetilde{a}_k - \widetilde{a}_{k-1}.$$

In particular, $D_N = 1/(4\tau_N)$ and $a_N = \widetilde{a}_N = 1$. For $0 \leq k < N$, define $\delta_k \in \mathbb{R}$ by

$$\delta_k = \frac{(k+2)(N-k)(N+k+1)}{2N(N+1)}.$$

For $0 \leq k < N - 1$, define $\alpha_k \in \mathbb{R}$ by the first line below; the second line defines the terminal weight $\alpha_{N-1} \in \mathbb{R}$:

$$\begin{aligned} \alpha_k &= \frac{a_k (D_{k+1} + 1)^2}{2} - D_N - \frac{k D_k \delta_{k+1}}{k+3} - \frac{a_{k+1} k^2 D_k^2}{2(k+3)^2}, \\ \alpha_{N-1} &= \frac{D_N (D_N - 1)}{2} = \frac{(N-1)(N+6)(N^2+5N+2)}{128}. \end{aligned} \tag{115}$$

These definitions give $\alpha_0 = 0$.[32]

For $0 \leq k \leq N$, define the normalized position $Y_k \in \mathcal{H}$ by

$$Y_k = \frac{y_k - x_\star}{D_k}.$$

For $1 \leq k < N$, define the state, coefficient vector, and matrix $\mathbf{Q}_k \in \mathbb{R}^{2\times 2}$ by

$$\boldsymbol{w}_k = \begin{bmatrix} Y_k \\ x_k - x_\star \end{bmatrix} \in \mathcal{H}^2, \qquad \mathbf{d}_k = \frac{1}{2} \begin{bmatrix} -\widetilde{a}_k D_k \\ a_k \end{bmatrix} \in \mathbb{R}^2,$$
$$\mathbf{Q}_k = \begin{bmatrix} D_N - \widetilde{a}_k D_k + \dfrac{(\widetilde{a}_k D_k)^2 - \delta_k^2}{2a_k} & -\dfrac{\widetilde{a}_k D_k}{2} \\ -\dfrac{\widetilde{a}_k D_k}{2} & \dfrac{a_k}{2} \end{bmatrix}.$$

Define the Lyapunov values $V_k \in \mathbb{R}$ as in (108) and (109), *i.e.*, set $V_0 = 0$ and

$$V_N = F(y_N) - F(x_\star) - \tau_N L \|x_0 - x_\star\|^2,$$

[32] This includes the terminal definition when $N = 1$.

and, for $1 \le k < N$,

$$V_k = a_k\big[f(x_k) - f(x_\star)\big] + \widetilde{a}_k\big[g(y_k) - g(x_\star)\big] - \tau_N L\,\|x_0 - x_\star\|^2 - 2\,\big\langle \nabla f(x_k), \mathbf{d}_k^\top \boldsymbol{w}_k\big\rangle + L\boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k.$$

Use the smooth interpolation residual $\mathcal{I}_f$ from (11). Proximal optimality gives $L(x_k - y_{k+1}) - \nabla f(x_k) \in \partial g(y_{k+1})$, so define the real-valued residual $\mathcal{I}_g(\cdot, y_{k+1}) : \operatorname{dom} g \to \mathbb{R}$ by

$$\mathcal{I}_g(p, y_{k+1}) = g(y_{k+1}) - g(p) + \langle L(x_k - y_{k+1}) - \nabla f(x_k), p - y_{k+1}\rangle \le 0, \qquad p \in \operatorname{dom} g.$$

The subgradient is attached to the indexed occurrence of y_{k+1}, even when iterates coincide. For $0 \le k < N-1$, define $s_k \in \mathcal{H}$ by the first line below, and set $s_{N-1} \in \mathcal{H}$ by the second:

$$s_k = \frac{\nabla f(x_k) - \nabla f(x_{k+1})}{L} + Y_{k+1} - (x_k - y_{k+1}),$$
$$s_{N-1} = \frac{\nabla f(x_{N-1}) - \nabla f(y_N)}{L} + Y_N - (x_{N-1} - y_N).$$

For the terminal step, use

$$\widehat{x}_{k+1} = \begin{cases} x_{k+1}, & 0 \le k < N-1, \\ y_N, & k = N-1. \end{cases}$$

For notational convenience, define $\mathcal{T}_k, S_k \in \mathbb{R}$, for $0 \le k < N$, by

$$\mathcal{T}_k = a_k \mathcal{I}_f(x_k, \widehat{x}_{k+1}) + b_{k+1}\mathcal{I}_f(x_\star, \widehat{x}_{k+1}) + \widetilde{a}_k \mathcal{I}_g(y_k, y_{k+1}) + \widetilde{b}_{k+1}\mathcal{I}_g(x_\star, y_{k+1}), \qquad 0 \le k < N,$$

where the term with $\widetilde{a}_0 = 0$ is omitted, and

$$S_k = \frac{La_k}{2}\,\|s_k\|^2 + \frac{b_{k+1}}{2L}\,\|\nabla f(\widehat{x}_{k+1}) - \nabla f(x_\star)\|^2 + L\alpha_k\,\|Y_{k+1} - Y_k\|^2.$$

Then $S_k \ge 0$, and, for $N \ge 2$, the following decrement identities hold:[33]

$$V_{k+1} - V_k = \begin{cases} \mathcal{T}_0 - S_0 + b_0\mathcal{I}_f(x_\star, x_0) - \dfrac{b_0}{2L}\,\|\nabla f(x_0) - \nabla f(x_\star)\|^2, & k = 0, \\ \mathcal{T}_k - S_k, & 1 \le k < N-1, \\ \mathcal{T}_{N-1} - S_{N-1}, & k = N-1. \end{cases}$$

The coefficient signs in Lemma F.10 and the nonpositive interpolation residuals imply $V_N \le \cdots \le V_0 = 0$. The terminal definition and the radius assumption therefore give

$$F(y_N) - F(x_\star) \le \tau_N L\,\|x_0 - x_\star\|^2 \le \frac{2LR^2}{N^2 + 5N + 2}.$$

Proof. The companion notebook is `examples_peppy/fista_rational_tight/fista_rational_tight_example_lyap.ipynb`. The notebook retains its historical potential at the proximal checkpoints separately. Code Cell 18 verifies the interpolation redistribution relating it to the potential used here; the checks below use (108)–(109) literally.

Coefficient identities. The weight identities below hold for $0 \le k < N$, and the scale recurrence holds for $1 \le k < N$. Code Cell 18 verifies each identity by exact substitution in the literal coefficient formulas:

$$(D_{k+1} + 1)a_k = \widetilde{a}_{k+1}D_{k+1}, \qquad \widetilde{a}_{k+1}D_{k+1} - \widetilde{a}_k D_k = \delta_k, \tag{116}$$

$$\frac{2k+1}{k+2}D_k - \frac{k-1}{k+2}D_{k-1} = D_{k+1} + 1. \tag{117}$$

At startup, $D_0 = D_1 + 1 = 2D_N$. For $1 \le k < N-1$, we also have

$$\widetilde{a}_k D_k + \delta_k(D_{k+1} + 1) - 2D_N = \frac{kD_k\delta_{k+1}}{k+3}. \tag{118}$$

[33] For $N = 1$, the first case holds with $\widehat{x}_1 = y_1$, so the initial and terminal conventions apply to the same step.

The FISTA recurrence and (117) therefore give, for $1 \le k < N$,

$$Y_{k+1} - (x_k - y_{k+1}) = (D_{k+1}+1)\,(Y_{k+1} - Y_k) - \frac{(k-1)D_{k-1}}{k+2}\,(Y_k - Y_{k-1})\,. \tag{119}$$

For $1 \le k < N-1$, subtracting two consecutive instances of (109) cancels the common initial-radius term and gives

$$\begin{aligned} V_{k+1} - V_k = {} & a_{k+1}\big(f(x_{k+1}) - f(x_\star)\big) - a_k\big(f(x_k) - f(x_\star)\big) \\ & + \widetilde{a}_{k+1}\big(g(y_{k+1}) - g(x_\star)\big) - \widetilde{a}_k\big(g(y_k) - g(x_\star)\big) \\ & - 2\left\langle \nabla f(x_{k+1}), \mathbf{d}_{k+1}^\top \boldsymbol{w}_{k+1}\right\rangle + 2\left\langle \nabla f(x_k), \mathbf{d}_k^\top \boldsymbol{w}_k\right\rangle \\ & + L\boldsymbol{w}_{k+1}^\top \mathbf{Q}_{k+1}\boldsymbol{w}_{k+1} - L\boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k. \end{aligned} \tag{120}$$

Our goal is to identify this expression with the interior case of (112). We first factor the quadratic remainder, then match the weighted interpolation residuals; the boundary cases are treated separately. In particular, neither $\nabla f(x_\star)$ nor the selected subgradient of g at $x_\star$ is assumed to vanish separately.

Step 1: factorization of the interior remainder. For $1 \le k < N-1$, Steps 1 and 2 prove the second line of (112), namely,

$$\begin{aligned} V_{k+1} - V_k = {} & a_k \mathcal{I}_f(x_k, x_{k+1}) + b_{k+1}\mathcal{I}_f(x_\star, x_{k+1}) \\ & + \widetilde{a}_k \mathcal{I}_g(y_k, y_{k+1}) + \widetilde{b}_{k+1}\mathcal{I}_g(x_\star, y_{k+1}) - S_k. \end{aligned}$$

Step 1 factors the quadratic form associated with a symmetric matrix; Step 2 identifies it by expanding the interpolation residuals and the potential difference. Define $\boldsymbol{r}_k \in \mathcal{H}^5$ and $M_k \in \mathbb{R}^{2\times 2}$ by

$$\boldsymbol{r}_k = \begin{bmatrix} Y_k \\ Y_{k+1} \\ x_k - y_{k+1} \\ (\nabla f(x_k) - \nabla f(x_{k+1}))/L \\ (\nabla f(x_{k+1}) - \nabla f(x_\star))/L \end{bmatrix}, \qquad M_k = \begin{pmatrix} 0 & 1 \\ -kD_k/(k+3) & (2k+3)D_{k+1}/(k+3) \end{pmatrix}.$$

The FISTA update gives

$$\boldsymbol{w}_{k+1} = M_k \begin{bmatrix} Y_k \\ Y_{k+1} \end{bmatrix}.$$

Define the symmetric matrix $\mathbf{S}_k \in \mathbb{R}^{5\times 5}$ by

$$\begin{aligned} \mathbf{S}_k = {} & \begin{pmatrix} \operatorname{diag}(1, D_{k+1})\mathbf{Q}_k \operatorname{diag}(1, D_{k+1}) - M_k^\top \mathbf{Q}_{k+1} M_k & 0_{2\times 3} \\ 0_{3\times 2} & 0_{3\times 3} \end{pmatrix} \\ & + \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_k & a_k & 0 \\ 0 & -a_k & a_k & -a_k & 0 \\ 0 & a_k & -a_k & a_k & 0 \\ 0 & 0 & 0 & 0 & b_{k+1} \end{pmatrix}. \end{aligned} \tag{121}$$

The identities $(\mathbf{Q}_k)_{12} = -\widetilde{a}_k D_k/2$, $(\mathbf{Q}_k)_{22} = a_k/2$, and $\widetilde{a}_{k+1}D_{k+1} = a_k(D_{k+1}+1)$ give the displayed cross terms. The claim of Step 1 is $L\boldsymbol{r}_k^\top \mathbf{S}_k \boldsymbol{r}_k = S_k$. For $1 \le k < N$, the definitions imply

$$\begin{aligned} 2\mathbf{d}_k^\top \boldsymbol{w}_k &= a_k(x_k - x_\star) - \widetilde{a}_k(y_k - x_\star) = \delta_k Y_k + \frac{a_k(k-1)D_{k-1}}{k+2}\,(Y_k - Y_{k-1})\,, \\ \boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k &= (D_N - \widetilde{a}_k D_k)\,\|Y_k\|^2 + \frac{\delta_k(k-1)D_{k-1}}{k+2}\,\langle Y_k, Y_k - Y_{k-1}\rangle \\ & \quad + \frac{a_k(k-1)^2 D_{k-1}^2}{2(k+2)^2}\,\|Y_k - Y_{k-1}\|^2\,. \end{aligned} \tag{122}$$

For $1 \le k < N-1$, subtracting consecutive instances of (122) yields

$$\begin{aligned} & \boldsymbol{w}_{k+1}^\top \mathbf{Q}_{k+1}\boldsymbol{w}_{k+1} - \boldsymbol{w}_k^\top \mathbf{Q}_k \boldsymbol{w}_k \\ & = \langle x_k - y_{k+1}, \widetilde{a}_k(y_k - x_\star) - \widetilde{a}_{k+1}(y_{k+1} - x_\star)\rangle \\ & \quad - \frac{a_k}{2}\,\|Y_{k+1} - (x_k - y_{k+1})\|^2 - \alpha_k\,\|Y_{k+1} - Y_k\|^2\,. \end{aligned} \tag{123}$$

Indeed, expand both sides using (119) and the weight identities. The remaining coefficient comparisons reduce to

$$\begin{aligned}\langle Y_{k+1}, Y_{k+1}-Y_k\rangle: &\quad \widetilde{a}_k D_k + \delta_k(D_{k+1}+1) - 2D_N = \frac{kD_k\delta_{k+1}}{k+3},\\ \|Y_{k+1}-Y_k\|^2: &\quad \frac{a_k(D_{k+1}+1)^2}{2} - D_N - \frac{kD_k\delta_{k+1}}{k+3} - \frac{a_{k+1}k^2D_k^2}{2(k+3)^2} = \alpha_k.\end{aligned}$$

These are (118) and (115), respectively.

Equation (123) shows that the position terms in $\boldsymbol{r}_k^\top \mathbf{S}_k \boldsymbol{r}_k$ equal

$$\frac{a_k}{2}\left\|Y_{k+1}-(x_k-y_{k+1})\right\|^2 + \alpha_k\left\|Y_{k+1}-Y_k\right\|^2.$$

The resulting matrix factorization is

$$\mathbf{S}_k = \frac{a_k}{2}\begin{bmatrix}0\\1\\-1\\1\\0\end{bmatrix}\begin{bmatrix}0\\1\\-1\\1\\0\end{bmatrix}^\top + \frac{b_{k+1}}{2}\begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix}\begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix}^\top + \alpha_k\begin{bmatrix}-1\\1\\0\\0\\0\end{bmatrix}\begin{bmatrix}-1\\1\\0\\0\\0\end{bmatrix}^\top.$$

Evaluating at $\boldsymbol{r}_k$ gives

$$\begin{aligned}L\boldsymbol{r}_k^\top\mathbf{S}_k\boldsymbol{r}_k = \frac{La_k}{2}&\left\|\frac{\nabla f(x_k)-\nabla f(x_{k+1})}{L} + Y_{k+1} - (x_k-y_{k+1})\right\|^2\\ &+ \frac{b_{k+1}}{2L}\left\|\nabla f(x_{k+1})-\nabla f(x_\star)\right\|^2 + L\alpha_k\left\|Y_{k+1}-Y_k\right\|^2 = S_k,\end{aligned} \tag{124}$$

by the definition of s_k. Code Cell 18 verifies the position identities and this full quadratic factorization coefficient by coefficient.

Step 2: verification of the interior decrement identity. For $1 \le k < N-1$, it remains to prove $\mathcal{T}_k - (V_{k+1}-V_k) = L\boldsymbol{r}_k^\top\mathbf{S}_k\boldsymbol{r}_k$. To match (120), compare the smooth and nonsmooth function values separately. The identities $b_{k+1} = a_{k+1}-a_k$ and $\widetilde{b}_{k+1} = \widetilde{a}_{k+1}-\widetilde{a}_k$ give exactly the first two lines of that difference. Expanding the residuals and using the selected subgradient $L(x_k-y_{k+1})-\nabla f(x_k)$ gives

$$\begin{aligned}\mathcal{T}_k = {}& a_{k+1}\big(f(x_{k+1})-f(x_\star)\big) - a_k\big(f(x_k)-f(x_\star)\big)\\ &+ \widetilde{a}_{k+1}\big(g(y_{k+1})-g(x_\star)\big) - \widetilde{a}_k\big(g(y_k)-g(x_\star)\big)\\ &+ \frac{a_k}{2L}\left\|\nabla f(x_k)-\nabla f(x_{k+1})\right\|^2 + \frac{b_{k+1}}{2L}\left\|\nabla f(x_{k+1})-\nabla f(x_\star)\right\|^2\\ &+ \langle\nabla f(x_{k+1}), a_k(x_k-x_\star) - a_{k+1}(x_{k+1}-x_\star)\rangle\\ &+ \langle L(x_k-y_{k+1})-\nabla f(x_k), \widetilde{a}_k(y_k-x_\star) - \widetilde{a}_{k+1}(y_{k+1}-x_\star)\rangle.\end{aligned} \tag{125}$$

The function-value terms therefore cancel upon subtracting (120). Using $2\mathbf{d}_k^\top\boldsymbol{w}_k = a_k(x_k-x_\star) - \widetilde{a}_k(y_k-x_\star)$ at indices k and $k+1$, the remaining terms linear in the gradients equal

$$\begin{aligned}&\langle\nabla f(x_k)-\nabla f(x_{k+1}), \widetilde{a}_{k+1}(y_{k+1}-x_\star) - a_k(x_k-x_\star)\rangle\\ &\quad = a_k\langle\nabla f(x_k)-\nabla f(x_{k+1}), Y_{k+1}-(x_k-y_{k+1})\rangle,\end{aligned}$$

where the equality uses $\widetilde{a}_{k+1}D_{k+1} = a_k(D_{k+1}+1)$. The two smooth interpolation norm terms give the two pure-gradient terms in $L\boldsymbol{r}_k^\top\mathbf{S}_k\boldsymbol{r}_k$; the remaining terms give its position terms. Thus

$$\begin{aligned}&a_k\mathcal{I}_f(x_k,x_{k+1}) + b_{k+1}\mathcal{I}_f(x_\star,x_{k+1}) + \widetilde{a}_k\mathcal{I}_g(y_k,y_{k+1})\\ &\quad + \widetilde{b}_{k+1}\mathcal{I}_g(x_\star,y_{k+1}) - (V_{k+1}-V_k) = L\boldsymbol{r}_k^\top\mathbf{S}_k\boldsymbol{r}_k = S_k.\end{aligned} \tag{126}$$

The final equality is Step 1. Rearranging proves the second line of (112) on the full range $1 \le k < N-1$. Code Cell 20 checks the bridge and the complete decrement in every independent Gram, function-value, and constant coefficient. The remaining initial and terminal cases are verified next, with the signs needed for monotonicity established in Lemma F.10.

Initial and terminal increments. These calculations establish the first and last lines of (112).[34] Code Cell 21 checks the initial decrement and Code Cell 22 checks the terminal decrement. For $N \geq 2$, $x_1 = y_1$ and $\widetilde{a}_1 D_1 = 1$, so

$$\boldsymbol{w}_1^\top \mathbf{Q}_1 \boldsymbol{w}_1 = (D_N - 1) \|Y_1\|^2, \qquad 2\mathbf{d}_1^\top \boldsymbol{w}_1 = \delta_1 Y_1.$$

The initial position identity is

$$\begin{aligned} &- \widetilde{a}_1 \langle x_0 - y_1, y_1 - x_\star \rangle - \frac{a_0}{2} \|Y_1 - (x_0 - y_1)\|^2 \\ &\quad = (D_N - 1) \|Y_1\|^2 - \tau_N \|x_0 - x_\star\|^2 . \end{aligned}$$

Together with $b_0 = a_0$, this gives the stated initial increment by the same expansion.

For $N \geq 2$, direct substitution at the terminal step in (122) gives

$$\begin{aligned} -\boldsymbol{w}_{N-1}^\top \mathbf{Q}_{N-1} \boldsymbol{w}_{N-1} = \langle x_{N-1} - y_N, \widetilde{a}_{N-1}(y_{N-1} - x_\star) - (y_N - x_\star) \rangle \\ - \frac{a_{N-1}}{2} \|Y_N - (x_{N-1} - y_N)\|^2 \\ - \alpha_{N-1} \|Y_N - Y_{N-1}\|^2 . \end{aligned}$$

In $\mathcal{T}_{N-1} - S_{N-1}$, all terms involving $\nabla f(y_N)$ cancel, using $a_{N-1}(D_N + 1) = D_N$ and $a_N = \widetilde{a}_N = 1$. Thus the terminal increment follows as well. For $N = 1$, the values

$$D_1 = 1, \quad \tau_1 = \frac{1}{4}, \quad a_0 = b_0 = b_1 = \frac{1}{2}, \quad a_1 = \widetilde{a}_1 = \widetilde{b}_1 = 1, \quad \alpha_0 = 0$$

give both boundary conventions in a single expansion. Code Cell 21 checks the literal one-step potential and (113) directly, omitting $g(y_0)$.

Positivity of the coefficients. Lemma F.10 proves the coefficient signs, including the zero boundary values, and the denominator bounds.

Every interpolation weight and every square weight in (112) is therefore nonnegative. Since $\mathcal{I}_f, \mathcal{I}_g \leq 0$, (112) and (113) give $V_N \leq V_{N-1} \leq \cdots \leq V_0 = 0$. The endpoint normalization (114), checked in Code Cell 23, proves the claimed rate.[35] □

Lemma F.10. The coefficients are positive on their respective index ranges:[36]

$$a_k > 0, \qquad b_k > 0, \qquad \widetilde{a}_k > 0, \qquad \widetilde{b}_k > 0, \qquad \alpha_k > 0.$$

Proof. For $1 \leq k \leq N$,

$$D_k = D_N + \frac{(N-k)(N+k+5)}{8} > 0, \qquad D_0 = 2D_N > 0.$$

Also, $\delta_k > 0$ for $0 \leq k < N$, $D_0 - D_1 = 1$, and $D_{k-1} - D_k = (k+2)/4 > 0$ for $k \geq 2$. Code Cell 19 verifies the denominator identities, weight differences, and the polynomial certificate below.

- **Positivity of a_k.** These coefficients are those of (19), so Lemma F.2 gives $a_k > 0$ for $0 \leq k \leq N$.
- **Positivity of b_k.** The same lemma gives $b_k > 0$ for $0 \leq k \leq N$.
- **Positivity of $\widetilde{a}_k$.** For $1 \leq k \leq N$, $\widetilde{a}_k = (D_k + 1)a_{k-1}/D_k > 0$. At index zero, $\widetilde{a}_0 = 0$.
- **Positivity of $\widetilde{b}_k$.** Equation (116) gives, for $1 \leq k \leq N$,

$$\widetilde{b}_k = \frac{\delta_{k-1} + \widetilde{a}_{k-1}(D_{k-1} - D_k)}{D_k} > 0.$$

[34] When $N = 2$, these are the only two transitions; Code Cells 21 and 22 verify both.

[35] Code Cell 23 covers both $N = 1$ and general N.

[36] The boundary values $\widetilde{a}_0 = \alpha_0 = 0$ are excluded from these strict inequalities; the unused value a_{-1} is also zero.

- **Positivity of α_k.** The value $\alpha_0 = 0$ follows directly from (115). For $N \geq 2$, the terminal formula gives $\alpha_{N-1} > 0$. For $2 \leq k < N$, write $\ell = N - k \in \mathbb{Z}$, with $\ell \geq 1$. Substitution in (115) gives

$$\begin{aligned}
&\frac{1024N(N+1)(k+2)^2(D_{k+1}+1)}{k-1}\alpha_{k-1} \\
&\quad = 8(5k+4)\ell^6 + 8(5k+4)(6k+11)\ell^5 \\
&\qquad + 4(133k^3 + 599k^2 + 772k + 296)\ell^4 \\
&\qquad + 8(78k^4 + 491k^3 + 969k^2 + 699k + 148)\ell^3 \\
&\qquad + 2(223k^5 + 1854k^4 + 5153k^3 + 5774k^2 + 2408k + 176)\ell^2 \\
&\qquad + 2(70k^6 + 779k^5 + 2964k^4 + 4855k^3 + 3576k^2 + 988k + 16)\ell \\
&\qquad + k(k+1)(k+6)(k^2+5k+2)(15k^2+35k+22) > 0.
\end{aligned}$$

  The multiplier of α_{k-1} on the left and every term on the right are positive, so $\alpha_{k-1} > 0$ for $2 \leq k < N$.

□

F.6 Matching the original FGM conjectures

The following lemma reconciles the C_k notation used in Theorems 5.1 and 5.2 with the notation of the original conjectures in Taylor et al. (2017c).

Lemma F.11 (Relation to the FGM conjecture notation)**.** Following Taylor et al. (2017c), define the fixed-step coefficients $h_{i,j} \in \mathbb{R}$ for $1 \leq i \leq N$ by

$$x_i = x_0 - \frac{1}{L}\sum_{j=0}^{i-1} h_{i,j}\nabla f(x_j), \qquad y_i = x_{i-1} - \frac{1}{L}\nabla f(x_{i-1}). \tag{127}$$

Then, for $0 \leq i \leq N$, with the sum understood to be empty when $i = 0$,

$$\sum_{j=0}^{i-1} h_{i,j} = C_{i+1} - 1.$$

Proof. Let $\widetilde{C}_k \in \mathbb{R}$ be given by

$$\widetilde{C}_k = \sum_{j=0}^{k-1} h_{k,j}, \qquad \widetilde{C}_0 = 0.$$

For $1 \leq k \leq N$, expanding the y- and z-updates gives

$$\begin{aligned}
y_k &= x_0 - \frac{1}{L}\left[\sum_{j=0}^{k-2} h_{k-1,j}\nabla f(x_j) + \nabla f(x_{k-1})\right], \\
z_k &= x_0 - \frac{1}{L}\sum_{j=0}^{k-1} \theta_j \nabla f(x_j),
\end{aligned}$$

with empty sums understood as zero. Substituting these expressions into $x_k = (1-1/\theta_k)y_k + z_k/\theta_k$ from (3) and collecting the terms involving each $\nabla f(x_j)$ gives

$$x_k = x_0 - \frac{1}{L}\sum_{j=0}^{k-2}\left[\left(1 - \frac{1}{\theta_k}\right)h_{k-1,j} + \frac{\theta_j}{\theta_k}\right]\nabla f(x_j) - \frac{1}{L}\left[1 - \frac{1}{\theta_k} + \frac{\theta_{k-1}}{\theta_k}\right]\nabla f(x_{k-1}).$$

Writing this expansion in the form of (127) with $i = k$ identifies the bracketed expressions as $h_{k,j}$. Hence

$$\widetilde{C}_k = \sum_{j=0}^{k-1} h_{k,j} = \sum_{j=0}^{k-2}\left[\left(1 - \frac{1}{\theta_k}\right)h_{k-1,j} + \frac{\theta_j}{\theta_k}\right] + \left[1 - \frac{1}{\theta_k} + \frac{\theta_{k-1}}{\theta_k}\right].$$

Collecting terms and using the definition of $\widetilde{C}_{k-1}$ gives

$$\widetilde{C}_k = \left(1 - \frac{1}{\theta_k}\right)\left(\sum_{j=0}^{k-2} h_{k-1,j} + 1\right) + \frac{1}{\theta_k}\sum_{j=0}^{k-1}\theta_j = \left(1 - \frac{1}{\theta_k}\right)(\widetilde{C}_{k-1} + 1) + \frac{1}{\theta_k}\sum_{j=0}^{k-1}\theta_j. \quad (128)$$

Using $\sum_{j=0}^{k-1}\theta_j = \theta_{k-1}^2$, obtained by telescoping (12), and $\theta_{k-1}^2 = \theta_k^2 - \theta_k$, we obtain

$$\widetilde{C}_k + 1 = \theta_k + \left(1 - \frac{1}{\theta_k}\right)(\widetilde{C}_{k-1} + 1).$$

This is the recurrence (14) for C_{k+1}. Since $\widetilde{C}_0 + 1 = C_1 = 1$, induction yields $\widetilde{C}_k = C_{k+1} - 1$ for every $0 \le k \le N$.

□

Applying Lemma F.11 with $i = N-1$ and $i = N$ shows that the bounds in Theorems 5.1 and 5.2 are exactly the FGM expressions in Conjectures 4 and 5 of Taylor et al. (2017c), respectively:

$$\frac{LR^2\theta_{N-1}^2}{2\left(2\sum_{j=0}^{N-1}\theta_j^3 + \theta_{N-1}^2\right)} = \frac{LR^2}{2(2C_N+1)} = \frac{LR^2}{2}\frac{1}{2\sum_{j=0}^{N-2} h_{N-1,j} + 3},$$

$$\frac{LR^2\theta_N^2}{2\left(2\sum_{j=0}^{N}\theta_j^3 - \theta_N^2\right)} = \frac{LR^2}{2(2C_{N+1}-1)} = \frac{LR^2}{2}\frac{1}{2\sum_{j=0}^{N-1} h_{N,j} + 1}.$$

The first equalities follow from (13); the second follow from Lemma F.11.

F.6.1 Smooth lower bounds

Taylor et al. (2017c, Section 4.2, Conjectures 4–5) give Huber instances for the two outputs of FGM. For rational coefficients, Taylor et al. (2017a, Section 4.2.2, Table 1 and the following paragraph) give the value at y_N and identify a Huber instance. The z form of (3) has inertial coefficient $(\theta_k - 1)/\theta_{k+1} = k/(k+3)$ for FGM-rational, matching that paper's FPGM1 update and y_N output. We verify attainment for every horizon.

Lemma F.12. *Fix $N \ge 1$, $L > 0$, and $R \ge 0$. For $R > 0$, work on $\mathbb{R}$, set $x_0 = y_0 = z_0 = R$ and $x_\star = 0$, and use the convex function $f_\tau : \mathbb{R} \to \mathbb{R}$ given by*

$$f_\tau(t) = \begin{cases} \dfrac{L}{2}t^2, & |t| \le \tau, \\ L\tau|t| - \dfrac{L}{2}\tau^2, & |t| \ge \tau. \end{cases}$$

For (FGM) *at y_N and x_N, and* (FGM-rational) *at y_N, choose, respectively,*

$$\tau = \frac{R}{2C_N + 1}, \qquad \tau = \frac{R}{2C_{N+1} - 1}, \qquad \tau = \frac{4R}{N^2 + 5N + 6}.$$

Then (4)*,* (5)*, and* (7) *hold with equality in the corresponding cases. For $R = 0$, the zero function attains all three bounds.*

Proof. Assume $R > 0$. The derivative of f_τ is continuous, nondecreasing, and has Lipschitz constant L. Thus $f_\tau \in \mathcal{F}_L$, $x_\star = 0$ is a minimizer, and $|x_0 - x_\star| = R$.

For each method, write the expansion of (3) in the form (127) and set $\widetilde{C}_k = \sum_{j=0}^{k-1} h_{k,j} \in \mathbb{R}$, with $\widetilde{C}_0 = 0$. Equation (128) uses only the common updates (3) with $\eta = 1$, without (12), so it applies to both parameter sequences. Replacing k by $k+1$ in (128) gives

$$\widetilde{C}_{k+1} = \left(1 - \frac{1}{\theta_{k+1}}\right)(\widetilde{C}_k + 1) + \frac{1}{\theta_{k+1}}\sum_{j=0}^{k}\theta_j.$$

Since $\theta_k \ge 1$, the recurrence is a convex combination of $\widetilde{C}_k + 1$ and $\sum_{j=0}^{k}\theta_j$. Starting from $\widetilde{C}_0 = 0$, induction gives $0 \le \widetilde{C}_k \le \sum_{j=0}^{k-1}\theta_j$ and $\widetilde{C}_{k+1} \ge \widetilde{C}_k + 1$, since $\widetilde{C}_k + 1 \le \sum_{j=0}^{k-1}\theta_j + 1 \le \sum_{j=0}^{k}\theta_j$. We first verify identities involving only $\widetilde{C}_k$ and τ:

- **FGM at y_N and x_N.** Lemma F.11 gives $\widetilde{C}_k = C_{k+1} - 1$. For $\tau = R/(2C_N + 1)$, the identity $\widetilde{C}_{N-1} + 1 = C_N$ gives $\tau(\widetilde{C}_{N-1} + 1) = (R - \tau)/2$. For $\tau = R/(2C_{N+1} - 1)$, the identity $\widetilde{C}_N = C_{N+1} - 1$ gives $\tau\widetilde{C}_N = (R - \tau)/2$.

- **FGM-rational at y_N.** Here $\theta_k = (k+2)/2$ and $\sum_{j=0}^{k} \theta_j = (k+1)(k+4)/4$. Substituting these into the common recurrence proves $\widetilde{C}_k = k(k + 7)/8$ by induction from $\widetilde{C}_0 = 0$, since

$$\widetilde{C}_{k+1} = \frac{k+1}{k+3}\left(\frac{k(k+7)}{8} + 1\right) + \frac{(k+1)(k+4)}{2(k+3)} = \frac{(k+1)(k+8)}{8}.$$

  Thus $\widetilde{C}_{N-1} + 1 = (N^2 + 5N + 2)/8$, so again $\tau(\widetilde{C}_{N-1} + 1) = (R - \tau)/2$.

These identities imply $0 < \tau \leq R$ and, by monotonicity, $\tau\widetilde{C}_k \leq (R - \tau)/2$ for $0 \leq k < N$ in each case. We now prove by induction that $x_k \in [\tau, \infty)$ for $0 \leq k < N$. On this interval, the Huber function satisfies $f_\tau(t) = L\tau t - L\tau^2/2$ and $\nabla f_\tau(t) = L\tau$.

- Since $x_0 = R \geq \tau$, we have $\nabla f_\tau(x_0) = L\tau$.

- For $1 \leq k < N$, assume $\nabla f_\tau(x_j) = L\tau$ for every $0 \leq j < k$. The gradient expansion (127) for the chosen method gives

$$x_k = R - \frac{1}{L}\sum_{j=0}^{k-1} h_{k,j}\nabla f_\tau(x_j) = R - \tau\widetilde{C}_k \geq \frac{R+\tau}{2} \geq \tau.$$

  Thus x_k also lies in this affine interval and $\nabla f_\tau(x_k) = L\tau$, closing the induction.

Using the same expansion, the y-update in (3), and the respective choices of τ, we obtain

$$y_N = R - \tau(\widetilde{C}_{N-1} + 1) = \frac{R+\tau}{2}, \qquad x_N = R - \tau\widetilde{C}_N = \frac{R+\tau}{2}.$$

Thus each final output lies in the affine part of f_τ, and

$$f_\tau\left(\frac{R+\tau}{2}\right) - f_\tau(0) = L\tau\left(\frac{R+\tau}{2} - \frac{\tau}{2}\right) = \frac{LR\tau}{2}.$$

Substituting the three choices of τ gives the claimed bounds. □

F.6.2 Composite lower bound

Taylor et al. (2017a, Section 4.2.2, Table 1 and the following paragraph) give the constrained linear instance and its y_N value for FPGM1. Its update and output coincide with (FISTA-rational) after shifting the index by one. We verify its value directly.

Lemma F.13. *Fix $N \geq 1$, $L > 0$, and $R \geq 0$. On $\mathbb{R}$, set $x_0 = y_0 = R$, $x_\star = 0$, and $c_N = 4LR/(N^2 + 5N + 2)$. Let $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ be*

$$f(t) = c_N t, \qquad g(t) = \begin{cases} 0, & t \geq 0, \\ +\infty, & t < 0. \end{cases}$$

Then (FISTA-rational) *attains $F(y_N) - F(x_\star) = 2LR^2/(N^2 + 5N + 2)$.*

Proof. The gradient of f is constant, so $f \in \mathcal{F}_L$. The function g is proper, closed, and convex. The point 0 minimizes $F = f + g$, satisfies $0 \in \nabla f(0) + \partial g(0)$, and lies at distance R from x_0. If $R = 0$, then $c_N = 0$ and the claim is immediate. Assume $R > 0$. The proximal step is projection onto $[0, \infty)$. We prove by induction that

$$x_k = R - \frac{c_N}{8L}k(k+7) \quad (0 \leq k \leq N), \qquad y_k = R - \frac{c_N}{8L}(k^2 + 5k + 2) \quad (1 \leq k \leq N).$$

The formula for x_0 is immediate. At each step $0 \le k < N$, the inductive expression for x_k gives

$$x_k - \frac{c_N}{L} = R - \frac{c_N}{8L}(k^2 + 7k + 8) \ge R - \frac{c_N}{8L}(N^2 + 5N + 2) = \frac{R}{2} > 0.$$

Thus the projection is inactive, and $y_{k+1} = x_k - c_N/L$ has the stated form. For $k = 0$, the extrapolation coefficient vanishes and $x_1 = y_1$, as required. For $k \ge 1$, the formulas give $y_{k+1} - y_k = -c_N(k+3)/(4L)$, so the extrapolation in (FISTA-rational) yields the stated expression for x_{k+1}. This closes the induction. In particular $y_N = R/2$, and hence

$$F(y_N) - F(0) = c_N \frac{R}{2} = \frac{2LR^2}{N^2 + 5N + 2}.$$

□