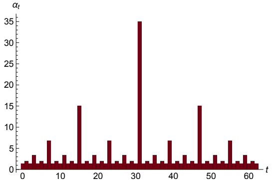
# Accelerating Gradient Descent by Stepsize Hedging

**Pablo A. Parrilo**
Joint work with Jason Altschuler (UPenn)




Laboratory for Information and Decision Systems (LIDS)
Massachusetts Institute of Technology
parrilo@mit.edu

Applied Algorithms for ML
Paris, June 2024

arXiv:2309.07879 arXiv:2309.16530

Massachusetts Institute of Technology

**Q:** Is it possible to accelerate Gradient Descent (GD) *without changing the algorithm*?

# Introduction

**Q:** Can we accelerate Gradient Descent (GD) without changing the algorithm?

- Instead, simply by a judicious choice of stepsizes?

$$GD: x_{k+1} = x_k - \eta_k \nabla f(x_k)$$

# Introduction

**Q:** Can we accelerate Gradient Descent (GD) without changing the algorithm?

- Instead, simply by a judicious choice of stepsizes?

$$GD : x_{k+1} = x_k - \eta_k \nabla f(x_k)$$

- Mainstream GD analysis uses constant (or diminishing) stepsize $\eta$
- Convergence rate: typically $\mathcal{O}(1/\epsilon)$ iterations
- **Example Applications:** Modern optimization, engineering, machine learning
- Earlier empirical works hint at potential advantages (e.g., cyclic schedules in NN training)
- Huge variety of other gradient-based methods (momentum, Nesterov, adaptive, etc) – here we can ONLY change the stepsize (non-adaptively)

# Mainstream GD Analysis

- Typical settings: convex $M$-smooth, or $(M, m)$ strongly convex
- With constant stepsize $\eta$, convergence in $\mathcal{O}(1/\epsilon)$ or $\mathcal{O}(\kappa \log(1/\varepsilon))$ iterations (slow rate, unaccelerated rate)
- E.g., textbooks by Polyak, Nesterov, Boyd, Vandenberghe, Bertsekas, Bubeck, Hazan
- Issue: Constant schedule converges slowly, even after optimizing $\eta$. For instance, for $M$-smooth, $m$-strongly convex functions, optimal (1-step) stepsize gives

$$\eta_\star = \frac{2}{m+M}, \qquad \|x_{k+1} - x_\star\| \leq \left(\frac{M-m}{M+m}\right)\|x_k - x_\star\| \approx (1 - \frac{2}{\kappa})\|x_k - x_\star\|$$

  where $\kappa = m/M$ is the condition number
- Many other stepsize proposals (e.g., line search, Armijo, Goldstein, Barzilai-Borwein), but don't provably help for convex optimization

# Mainstream GD Analysis

- Typical settings: convex $M$-smooth, or $(M, m)$ strongly convex
- With constant stepsize $\eta$, convergence in $\mathcal{O}(1/\epsilon)$ or $\mathcal{O}(\kappa \log(1/\varepsilon))$ iterations (slow rate, unaccelerated rate)
- E.g., textbooks by Polyak, Nesterov, Boyd, Vandenberghe, Bertsekas, Bubeck, Hazan
- Issue: Constant schedule converges slowly, even after optimizing $\eta$. For instance, for $M$-smooth, $m$-strongly convex functions, optimal (1-step) stepsize gives

$$\eta_\star = \frac{2}{m+M}, \qquad \|x_{k+1} - x_\star\| \leq \left(\frac{M-m}{M+m}\right) \|x_k - x_\star\| \approx (1 - \frac{2}{\kappa})\|x_k - x_\star\|$$

  where $\kappa = m/M$ is the condition number
- Many other stepsize proposals (e.g., line search, Armijo, Goldstein, Barzilai-Borwein), but don't provably help for convex optimization

**Any** reason to be hopeful?

# Convex **Quadratic** Functions (Young 1953)

- Minimize $f(x) = \frac{1}{2}x^\top Qx$ where $Q$ is positive definite $(mI \preceq Q \preceq MI)$

$$GD: \; x_{k+1} = x_k - \eta_k \nabla f(x_k) = x_k - \eta_k Q x_k = (I - \eta_k Q)x_k$$

- Nice, because it becomes a question about eigenvalues:

$$eig(I - \eta_k Q) = 1 - \eta_k eig(Q)$$

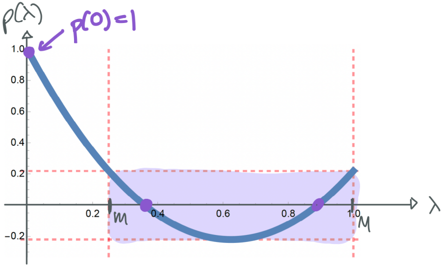
- Stepsize design is a polynomial optimization problem:

$$\min_{\eta} \max_{\lambda \in [m,M]} \underbrace{\left| \prod_{k=1}^{n} (1 - \lambda \eta_k) \right|}_{p_\eta(\lambda)}$$

Find a polynomial $p_\eta(\lambda)$ with $p_\eta(0) = 1$ that is "small" on $[m, M]$.

# Convex **Quadratic** Functions (Young 1953)

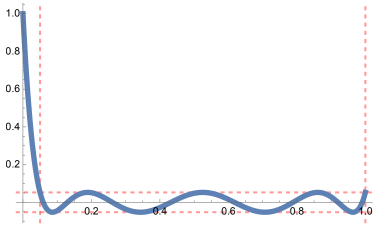
Classic problem, with a classic answer:
(scaled) Chebyshev polynomials.




Young (1953):

- Optimal gradient stepsizes are the inverse roots of (scaled) Chebyshev polynomials.
- Associated convergence rate is $\mathcal{O}(\sqrt{\kappa})$

Proves advantage of non-constant stepsizes. But, unclear whether it extends to other settings!

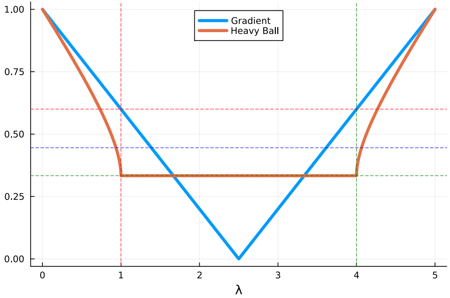



- **Key Point:** Non-constant stepsizes (hedging) can accelerate convergence — at least for quadratics

# Quadratic functions (and polynomials) are *very* special

(At least) three different viewpoints:

- Inverse roots and minimax characterization of Chebyshev polynomials
- Orthogonal polynomials and three-term recurrence (Heavy Ball, momentum, . . . )
- Asymptotic root distribution (arcsine distribution, potential theory, universality)




Unfortunately, most of these methods and proof techniques do not gracefully extend to the general (convex non-quadratic) case... :(

# Convex Optimization Challenges

- Before 2018, it was unknown whether any stepsize schedule leads to speedup over constant steps for any setting beyond quadratics
- Core difficulties: Many phenomena false beyond quadratics, multistep reasoning necessary
- **Additional challenge:** How to *find* optimal stepsizes beyond quadratics

| | **Quadratic** | **Convex** |
|---|---|---|
| **Mainstream** | $\Theta(\kappa)$ by constant stepsizes (folklore) | $\Theta(\kappa)$ by constant stepsizes (folklore) |
| **Mod. Algorithm** | $\Theta(\sqrt{\kappa})$ by Heavy Ball (Polyak'64) | $\Theta(\sqrt{\kappa})$ by Nesterov Acceleration |
| **Hedged Stepsizes** | $\Theta(\sqrt{\kappa})$ by Chebyshev Stepsizes (Young'53) | ??????? |

Table: Iteration complexity of various approaches for minimizing a $\kappa$-conditioned convex function. The dependence on the accuracy $\varepsilon$ is omitted as it is always $\log 1/\varepsilon$.

# Does Hedging Help for Non-Quadratic Convex Functions?

- Consider two possible setups: Minimize $f(x)$, which is either
  - convex and $M$-smooth
  - $m$-strongly convex and $M$-smooth
- **Algorithmic Opportunity:** Similar intuition as in quadratic case. Worst-case functions may not align, so there is an incentive for hedging

Hopefully easier to understand first: what can we do with two stepsizes?

Should they be the same? If not, do we want to do long/short, or short/long?

# The two-step case (Altschuler 2018)

Consider

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

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

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

The question of optimal stepsizes is therefore the minimax problem $\min_{\alpha,\beta} R(\alpha, \beta; \mathcal{F})$

**Theorem (Altschuler 2018, Thm 8.10)**

*For $(m, M)$-convex functions, the optimal two-step schedule and rate are*

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

*where $S = \sqrt{M^2 + (M-m)^2}$. Since $R^\star \approx 1 - \frac{2(1+\sqrt{2})}{\kappa} < \left(\frac{M-m}{M+m}\right)^2 \approx 1 - \frac{4}{\kappa}$, repeating this periodically gives a constant-factor improvement over the 1-step rate.*

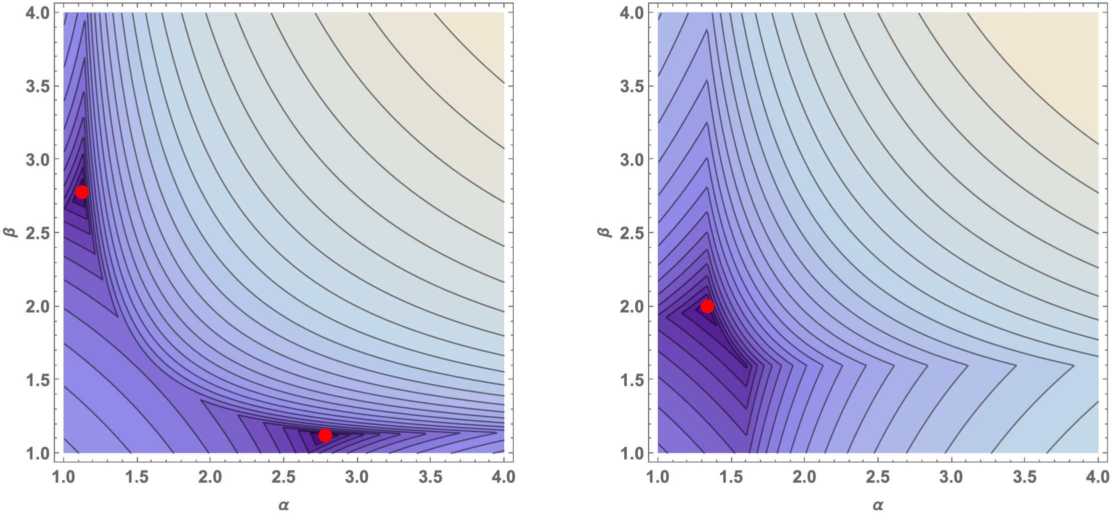



Figure: Stepsize hedging ($m = 1/4, M = 1$): quadratic (left) vs convex (right).
These are level sets of the convergence rate. Notice the symmetry-breaking, short/long is optimal.

# How much better?

OK, can do better with $n = 2$. What about $n = 3, 4, \ldots$. How much better?

- **Altschuler 2018** First results showing that non-constant steps help beyond quadratics.
  - Strongly convex and smooth (optimal 2- and 3-step)
  - Separable functions (iid arcsine stepsize, full acceleration)
- **Daccache 2019, Eloi 2022** Optimal stepsizes for $n = 2, 3$ for smooth case, also different performance criteria.
- **Das Gupta-Van Parys-Ryu 2022** Combined Branch & Bound and PESTO SDP to numerically search for $n$-step schedules (up to $n = 50$)
- **Grimmer 2023** Extend and round B&B solutions to rational numbers to rigorously certify approximate schedules up to $n = 127$, yields larger constant factor improvements.
- **Altschuler-P. 2023** Extends 2-step solution from [A. 2018] via recursion, proving acceleration and first asymptotic improvement: $\mathcal{O}(\kappa^{0.7864})$. For convex, $\mathcal{O}(\varepsilon^{-0.7864})$ (first via black-box reductions, later via simpler limiting case).
- **Grimmer-Shu-Wang 2023** Concurrent, obtain rates $\mathcal{O}(\kappa^{0.947})$ and $\mathcal{O}(\varepsilon^{-0.947})$.

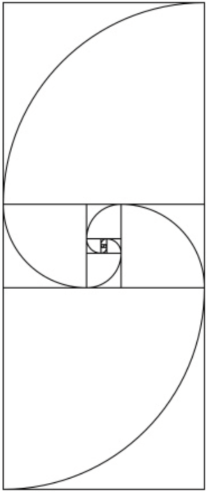
# Aside: the Silver Ratio

Define the number $\rho := 1 + \sqrt{2}$ (from the 2-step solution)
We have $\log_\rho 2 \approx 0.7864$ (from our convergence rate)

One of the "metallic means"

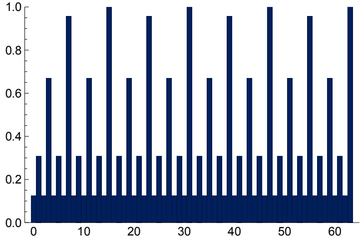
- $n = 1$ : Golden ratio $(1 + \sqrt{5})/2$
- $n = 2$: Silver ratio $1 + \sqrt{2}$
- $n = 3$: Bronze ratio . . .

Apparently used in Eastern architecture, and Japanese anime characters (though, there the ratios seem to be $\sqrt{2} : 1$)



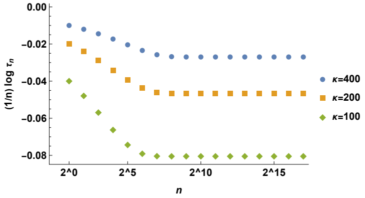
# Good Stepsize Hedging through Silver Stepsizes

- *Silver Stepsize* Schedule: a natural recursive construction (but can be made explicit)
- Non-monotonic fractal order, convergence rate has a phase transition
- Proof of multistep descent by enforcing long-range consistency conditions among iterates
- Non-strongly convex case is the (much simpler) limit of the $(m, M)$ strongly convex case



# Silver Stepsizes in $(m, M)$ Strongly Convex Setting

- Fully explicit recursive construction (later)
- Schedule is near-periodic of period $\kappa^{\log_2 \rho}$
- Largest stepsizes increase exponentially and later saturate
- Convergence rate has phase transition



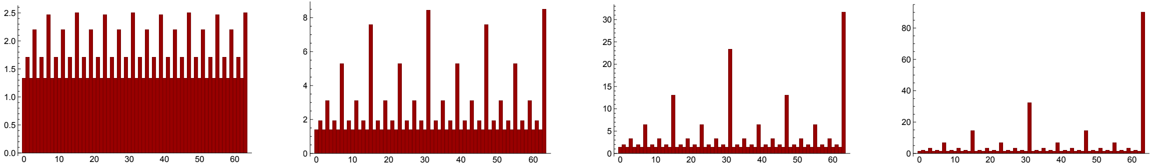



Figure: Silver Stepsizes for condition numbers $\kappa = 4, 16, 64, 256$ (only first 64 steps shown)

Altschuler-P., "Acceleration by Stepsize Hedging I: Multi-Step Descent and the Silver Stepsize Schedule," arXiv:2309.07879

| | **Quadratic** | **Convex** |
|---|---|---|
| **Mainstream** | $\Theta(\kappa)$ by constant stepsizes (folklore) | $\Theta(\kappa)$ by constant stepsizes (folklore) |
| **Mod. Algorithm** | $\Theta(\sqrt{\kappa})$ by Heavy Ball (Polyak'64) | $\Theta(\sqrt{\kappa})$ by Nesterov Acceleration |
| **Hedged Stepsizes** | $\Theta(\sqrt{\kappa})$ by Chebyshev Stepsizes (Young'53) | $\Theta(\kappa^{\log_2 \rho})$ by Silver Stepsizes |

Table: Iteration complexity for $\kappa$-conditioned convex functions. Here $\log_\rho 2 \approx 0.7864$

# Silver Stepsizes in $M$-smooth convex setting

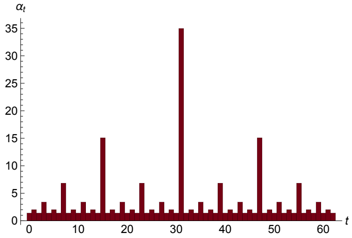
Simpler limiting case as $\kappa \to \infty$. Recursive construction:

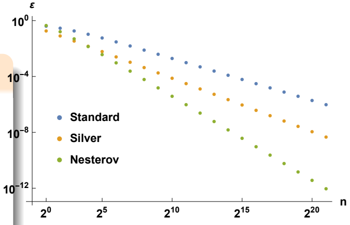
$$h_{2n+1} = [\, h_n, \quad 1 + \rho^{k-1}, \quad h_n \,],$$

with $h_1 := [\sqrt{2}]$.

Can be made explicit, easy to implement (e.g., Python)

```
[1+rho**((k & -k).bit_length()-2) for k in range(1,64)]
```




**Theorem**

*If $f$ is convex and $M$-smooth, Silver Stepsizes yield ($n = 2^k - 1$)*

$$f(x_n) - f_\star \leq \frac{M}{2n^{\log_2 \rho}} \|x_0 - x_\star\|^2 \approx \frac{M}{2n^{1.2716}} \|x_0 - x_\star\|^2$$

Altschuler-P., "Acceleration by Stepsize Hedging II: Silver Stepsize Schedule for Smooth Convex Optimization," arXiv:2309.16530

# How to analyze this?

Techniques have long history in dynamical systems and robust control (Lyapunov, $\mu$-analysis, Linear Matrix Inequalities (LMIs), Integral Quadratic Constraints (IQCs), Sum of Squares (SOS). More recently, PEP/PESTO, neural network certification, etc.)

Essentially:

- Write valid inequalities for the "uncertain" or "nonlinear" part of the system. Typically quadratic or polynomial.
- Use Lagrangian duality (or stronger things, like the Positivstellensatz) to find an identity that "obviously" certifies the desired conclusion
- Key: Proof system is convex optimization-friendly (e.g., SDP)

# Proof strategy for GD

- Desired function class $\mathcal{F}$ is described through interpolability conditions (Rockafellar, Taylor, etc.). For instance, for $(m, M)$ strong convexity, all data $(x_i, g_i, f_j)$ satisfies

$$Q_{ij} := 2(M-m)(f_i - f_j) + 2\langle Mg_j - mg_i, x_j - x_i \rangle - \|g_i - g_j\|^2 - Mm\|x_i - x_j\|^2 \geq 0$$

- Combine valid quadratic inequalities by nonnegative linear combinations (i.e., Lagrangian duality)
- E.g., Drori-Teboulle 2014, Lessard-Recht-Packard 2016, Taylor-Hendrickx-Glineur 2016, ...

Usually works fine for *fixed* $n$.

# In our case (at a high level)

Want to certify that for our stepsize choice $\eta_k$, the set of equations describing:

- Interpolability conditions on the data: $Q_{ij} \geq 0$ for all pairs $1 \leq i, j \leq n$
- Method definition: gradient descent equations

$$x_{k+1} = x_k - \eta_k g_k$$

directly imply the desired rate inequality.

For any finite $n$, this is just a finite collection of linear/quadratic inequalities in $(f_i, g_i, x_i)$. In particular we can do this by finding nonnegative multipliers $\lambda_{ij}$ such that

$$\sum_{ij} \lambda_{ij} Q_{ij} + (\text{something squared}) = \|x_0 - x_\star\|^2 + \frac{1}{R_n}(f_\star - f_n).$$

since this obviously implies $f_n - f_\star \leq R_n \|x_0 - x_\star\|^2$.

# Proof strategies

Caveats (!)

- To prove asymptotic improvements (not just constant factors), this must be done "symbolically," i.e., for all values of $n$
- Finding stepsizes $\eta_k$ is not (yet?) a convex problem. Typically, one proposes an ansatz based on small instances, and attempts to prove it.

In our case, the Silver Stepsizes were motivated by Jason's 2-step solution and numerical work.

We believe they are essentially optimal (work in progress, more soon!)

# Recursive gluing

A recursive certificate that *almost* works, by "gluing" two smaller certificates

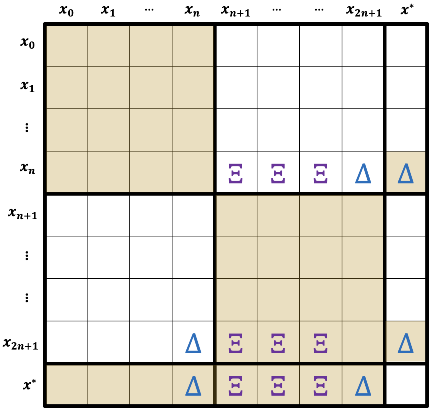
Then don't quite match, but can modify things to fix it

Write perturbation as sum of two quadratic forms:

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

Then an induction argument proves the identity for all $n$

*Proof verification* is fully algorithmic – no need to trust our math!



# Things to think about

- Finer-grained understanding for restricted function classes
- Robustness (cf. Devolder *et al.* for Nesterov's)
- Connections to superacceleration in neural network training?
- Rethink offline to online conversions
- Beyond GD: Re-investigating algorithms that use greedy analyses

# Takeaways

- Why this is interesting: provides a new mechanism for acceleration
- Result: Can (partially) accelerate GD simply by non-adaptive stepsize choice!
- Intuition: Hedging between misaligned worst-case functions
- Analysis: Multi-step descent by enforcing long-range consistency along GD trajectory
- Carefully exploits the “rigidity” of the cost at different timesteps
- Can we make algorithm analysis AND design *fully* algorithmic?