An Inertial Non-smooth Non-convex Bregman Minimization Framework

Peter Ochs

Mathematical Optimization Group

— 22.11.2021 —

Motivation

[Nesterov, 2004]:

Optimization is unsolvable!!!

Motivation

[Nesterov, 2004]:

Optimization is unsolvable!!!

- This statement is meant "**in general**".
- Nature is just too complicated!
- This is a **mathematical fact** rather than intuition.

$\rightsquigarrow$ study classes of optimization problems.

Motivation

[Nesterov, 2004]:

Optimization is unsolvable!!!

- This statement is meant "**in general**".
- Nature is just too complicated!
- This is a **mathematical fact** rather than intuition.

$\rightsquigarrow$ study classes of optimization problems.

Ultimate Goal:

meta algorithm that automatically detects and adapts to the problem structure

Goal of this Talk

Unified framework for the design and analysis of many available algorithms and thereby to explore synergies and to understand the essentials.

Goal of this Talk

Unified framework for the design and analysis of many available algorithms and thereby to explore synergies and to understand the essentials.

Setup: finite dimensional, non-smooth, non-convex, first-order optimization.

Part 1: Model Framework and Global Convergence

- unified framework for algorithm design
- convergence analysis
- examples

Part 2: Inertial Variant based on Nesterov Extrapolation

- introduce in standard Euclidean setting
- and generalize to model framework

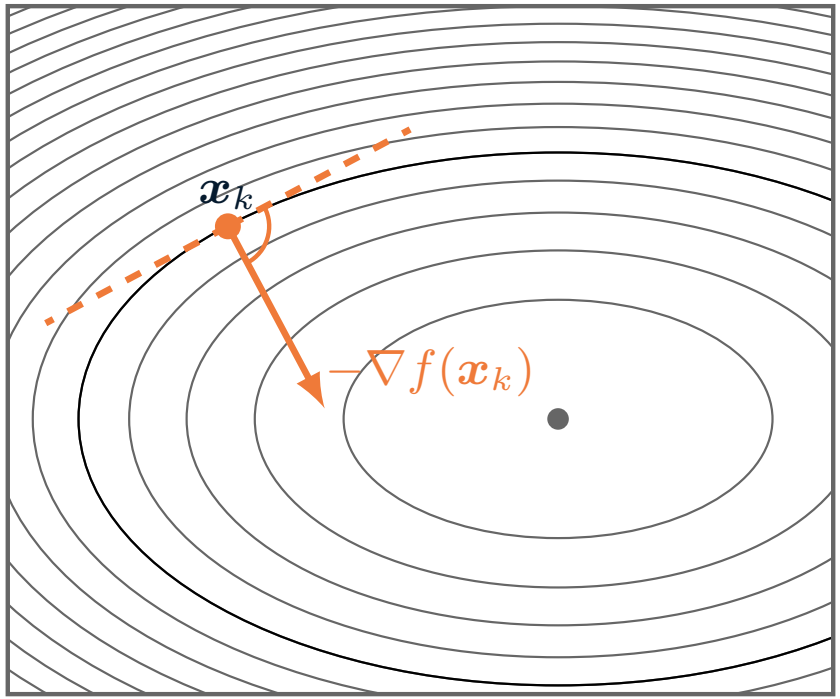
A Simple Illustrative Setting

Smooth optimization problem:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) \qquad (\nabla f \text{ is } L\text{-Lipschitz continuous})$$

- Gradient Descent update step with step size $\tau > 0$:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \tau \nabla f(\boldsymbol{x}_k)$$



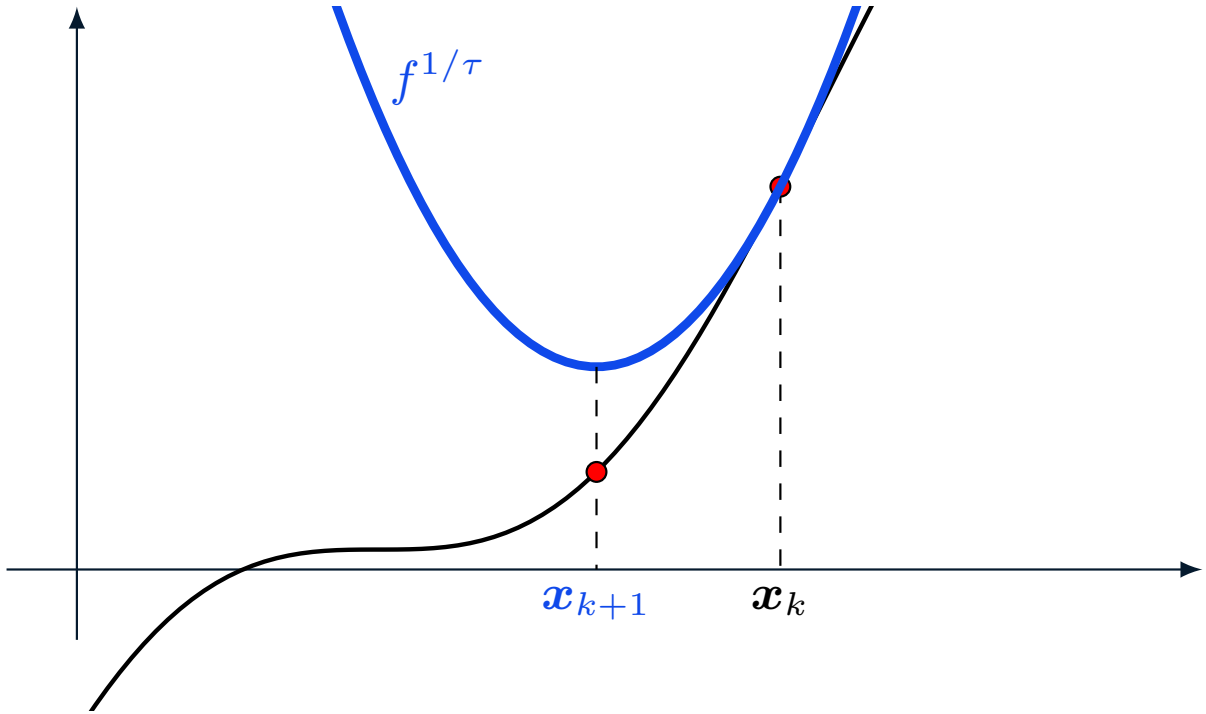
Equivalent Form of Gradient Descent

Equivalent to sequential minimization of quadratic functions:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \tau \nabla f(\boldsymbol{x}_k)$$

$$\iff \boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{1}{2\tau} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2}_{=: f^{1/\tau}}$$

(majorizer if $\tau \leq \frac{1}{L}$)



Algorithmic Model Function Framework

Optimization problem:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) \quad (\nabla f \text{ is } L\text{-Lipschitz})$$

Update step:

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle}_{\text{linearization of } f} + \underbrace{\frac{1}{2\tau} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2}_{\text{distance to } \boldsymbol{x}_k}$$

Algorithmic Model Function Framework

Optimization problem:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) \quad (\nabla f \text{ is } L\text{-Lipschitz})$$

Update step:

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle}_{\text{linearization of } f} + \underbrace{\frac{1}{2\tau} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2}_{\text{distance to } \boldsymbol{x}_k}$$

Classic setting: (Descent Lemma)

$$|f(\boldsymbol{x}) - f(\boldsymbol{x}_k) - \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle| \leq \frac{L}{2} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2$$

Algorithmic Model Function Framework

Optimization problem:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x})$$

Update step:

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle}_{\text{linearization of } f} + \underbrace{\frac{1}{2\tau} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2}_{\text{distance to } \boldsymbol{x}_k}$$

Classic setting: (Descent Lemma)

$$|f(\boldsymbol{x}) - f(\boldsymbol{x}_k) - \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle| \leq \frac{L}{2} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2$$

Algorithmic Model Function Framework

Optimization problem:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x})$$

Update step:

$$\boldsymbol{x}_{k+1} \in \operatorname*{argmin}_{\boldsymbol{x}} \; \underbrace{f_{\boldsymbol{x}_k}(\boldsymbol{x})}_{\text{model function}} + \underbrace{\frac{1}{2\tau}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2}_{\text{distance to } \boldsymbol{x}_k}$$

Classic setting: (Descent Lemma)

$$|f(\boldsymbol{x}) - f(\boldsymbol{x}_k) - \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}-\boldsymbol{x}_k\rangle| \leq \frac{L}{2}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2$$

Algorithmic Model Function Framework

Optimization problem:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x})$$

Update step:

$$\boldsymbol{x}_{k+1} \in \operatorname*{argmin}_{\boldsymbol{x}} \; \underbrace{f_{\boldsymbol{x}_k}(\boldsymbol{x})}_{\text{model function}} + \underbrace{\frac{1}{2\tau}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2}_{\text{distance to } \boldsymbol{x}_k}$$

Model Approximation Property: [Drusvyatskiy et al. 21], [O. et al. 18], [Mukkamala et al. 21], ...

$$|f(\boldsymbol{x}) - f_{\boldsymbol{x}_k}(\boldsymbol{x})| \leq \frac{L}{2}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2$$

Algorithmic Model Function Framework

Optimization problem:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) \quad (f \text{ non-smooth})$$

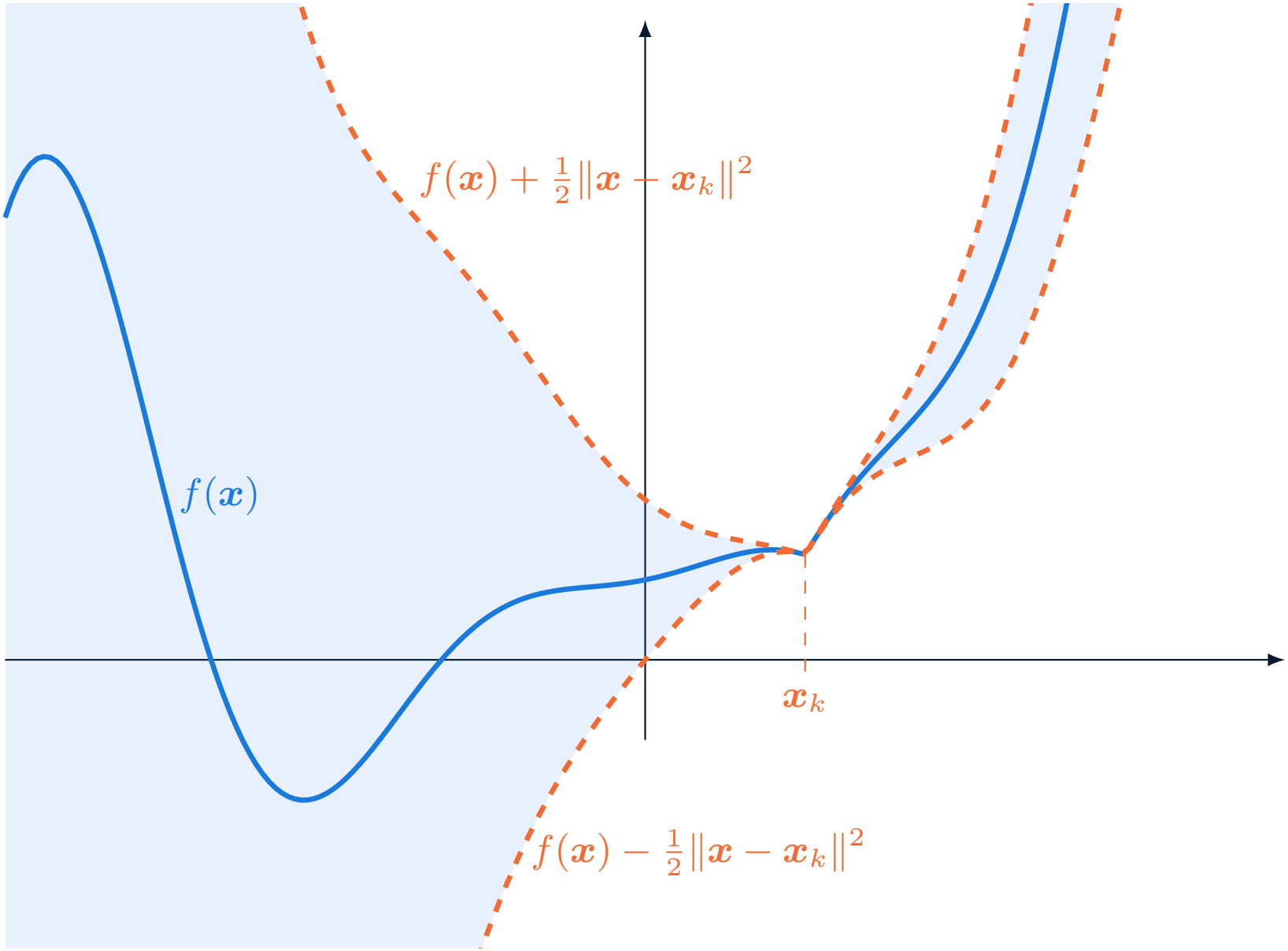
Update step:

$$\boldsymbol{x}_{k+1} \in \operatorname*{argmin}_{\boldsymbol{x}} \; \underbrace{f_{\boldsymbol{x}_k}(\boldsymbol{x})}_{\text{model function}} + \underbrace{\frac{1}{2\tau}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2}_{\text{distance to } \boldsymbol{x}_k}$$

Model Approximation Property: [Drusvyatskiy et al. 21], [O. et al. 18], [Mukkamala et al. 21], ...

$$|f(\boldsymbol{x}) - f_{\boldsymbol{x}_k}(\boldsymbol{x})| \leq \frac{L}{2}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2$$

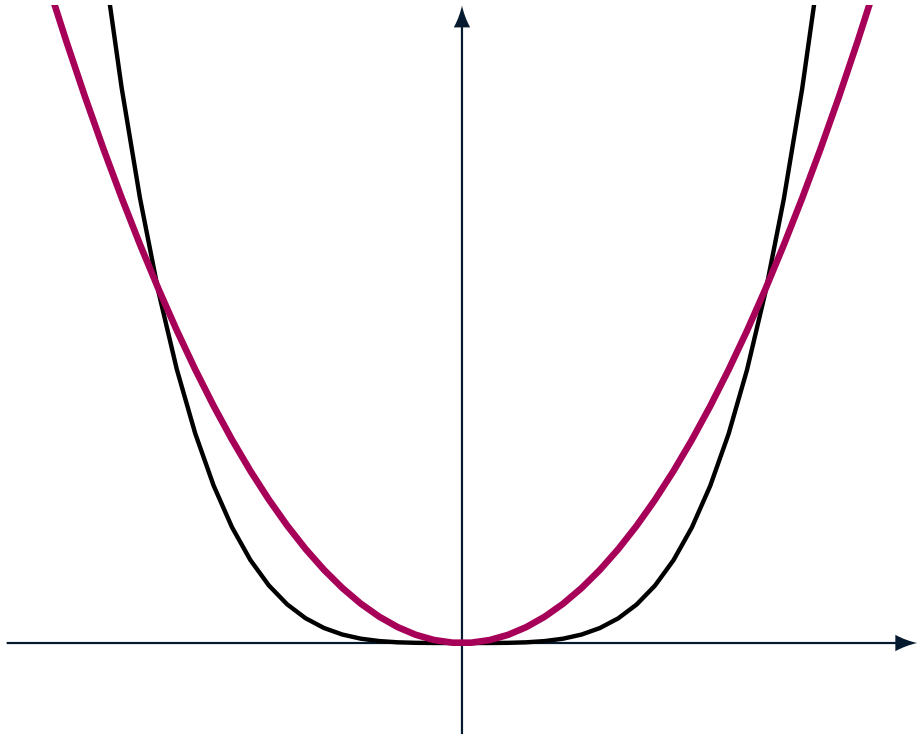
Model Approximation Property



Lack of Lipschitz Continuity

Failure of Quadratic Model Error: $\rightsquigarrow$ relative smoothness

- Simple functions like x^4 **do not** allow for quadratic bounds:



- Same situation in Matrix Factorization:

$$Q(\boldsymbol{X}, \boldsymbol{X}^\top) = \frac{1}{2}\|\boldsymbol{A} - \boldsymbol{X}\boldsymbol{X}^\top\|_F^2 \quad \text{polynomial of degree } 4 \text{ in } \boldsymbol{X}$$

Algorithmic Model Function Framework

Optimization problem:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) \quad (f \text{ non-smooth})$$

Update step:

$$\boldsymbol{x}_{k+1} \in \operatorname*{argmin}_{\boldsymbol{x}} \; \underbrace{f_{\boldsymbol{x}_k}(\boldsymbol{x})}_{\text{model function}} + \underbrace{\frac{1}{2\tau}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2}_{\text{distance to } \boldsymbol{x}_k}$$

Model Approximation Property: [Drusvyatskiy et al. 21], [O. et al. 18], [Mukkamala et al. 21], ...

$$|f(\boldsymbol{x}) - f_{\boldsymbol{x}_k}(\boldsymbol{x})| \leq \frac{L}{2}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2$$

Algorithmic Model Function Framework

Optimization problem:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) \quad (f \text{ non-smooth})$$

Update step:

$$\boldsymbol{x}_{k+1} \in \operatorname*{argmin}_{\boldsymbol{x}} \; \underbrace{f_{\boldsymbol{x}_k}(\boldsymbol{x})}_{\text{model function}} + \underbrace{\frac{1}{\tau} D_h(\boldsymbol{x}, \boldsymbol{x}_k)}_{\text{Bregman distance to } \boldsymbol{x}_k}$$

Model Approximation Property: [Drusvyatskiy et al. 21], [O. et al. 18], [Mukkamala et al. 21], ...

$$|f(\boldsymbol{x}) - f_{\boldsymbol{x}_k}(\boldsymbol{x})| \leq L D_h(\boldsymbol{x}, \boldsymbol{x}_k)$$

Algorithmic Model Function Framework

Optimization problem:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) \quad (f \text{ non-smooth})$$

Update step:

$$\boldsymbol{x}_{k+1} \in \operatorname*{argmin}_{\boldsymbol{x}} \; \underbrace{f_{\boldsymbol{x}_k}(\boldsymbol{x})}_{\text{model function}} + \underbrace{\frac{1}{\tau} D_h(\boldsymbol{x}, \boldsymbol{x}_k)}_{\text{Bregman distance to } \boldsymbol{x}_k}$$

Model Approximation Property: [Drusvyatskiy et al. 21], [O. et al. 18], [Mukkamala et al. 21], ...

$$|f(\boldsymbol{x}) - f_{\boldsymbol{x}_k}(\boldsymbol{x})| \leq L D_h(\boldsymbol{x}, \boldsymbol{x}_k)$$

Generalizes L-smad property (smooth adaptable functions) / relative smoothness

[Birnbaum et al., 2011], [Bauschke et al., 2017], [Bolte et al., 2018], [Lu et al. 2018].

Global Convergence to a Stationary Point

Algorithm: Model BPG $(0 < \inf_k \tau_k \leq \sup_k \tau_k < 1/L)$

$$\boldsymbol{x}_{k+1} \in \operatorname*{argmin}_{\boldsymbol{x}} \; f_{\boldsymbol{x}_k}(\boldsymbol{x}) + \frac{1}{\tau_k} D_h(\boldsymbol{x}, \boldsymbol{x}_k)$$

Global Convergence to a Stationary Point

Algorithm: Model BPG $(0 < \inf_k \tau_k \leq \sup_k \tau_k < 1/L)$

$$\boldsymbol{x}_{k+1} \in \operatorname*{argmin}_{\boldsymbol{x}} \; f_{\boldsymbol{x}_k}(\boldsymbol{x}) + \frac{1}{\tau_k} D_h(\boldsymbol{x}, \boldsymbol{x}_k)$$

Theorem:

- $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ has finite length and converges a critical point of f.
- Standard convergence rates depending on KL-exponent of a Lyapunov function.

Global Convergence to a Stationary Point

Algorithm: Model BPG ($0 < \inf_k \tau_k \leq \sup_k \tau_k < 1/L$)

$$\boldsymbol{x}_{k+1} \in \operatorname*{argmin}_{\boldsymbol{x}} \; f_{\boldsymbol{x}_k}(\boldsymbol{x}) + \frac{1}{\tau_k} D_h(\boldsymbol{x}, \boldsymbol{x}_k)$$

Theorem:

- $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ has finite length and converges a critical point of f.
- Standard convergence rates depending on KL-exponent of a Lyapunov function.

Assumptions:

- D_h generated by Legendre function h that is C^2 on $\operatorname{int}\operatorname{dom} h$. Moreover h is σ_B-strongly convex and $\nabla^2 h$ is bounded on compact convex subsets $B \subset \operatorname{int}\operatorname{dom} h$.
- $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ is bounded (e.g. by coercivity) and set of limit points $\omega(\boldsymbol{x}_0) \subset \operatorname{int}\operatorname{dom} h$.
- model consistency: $\|\partial f_{\boldsymbol{y}}(\boldsymbol{x})\|_- \leq c\|\boldsymbol{x} - \boldsymbol{y}\|$.
- Functions are definable. (to apply KL-framework; details omitted)

Analysis based on **Lyapunov Function**: $F_L^h(\boldsymbol{x}, \bar{\boldsymbol{x}}) := f_{\bar{\boldsymbol{x}}}(\boldsymbol{x}) + L D_h(\boldsymbol{x}, \bar{\boldsymbol{x}})$

Proximal Gradient Method and Mirror Descent

- **Optimization problem**:

$$\min_{\boldsymbol{x}} \underbrace{f_0(\boldsymbol{x})}_{\text{simple}} + \underbrace{f_1(\boldsymbol{x})}_{\substack{\text{smooth} \\ \text{non-convex}}}$$

- **Model function**:

$$f_{\bar{\boldsymbol{x}}}(\boldsymbol{x}) = f_0(\boldsymbol{x}) + f_1(\bar{\boldsymbol{x}}) + \langle \boldsymbol{x} - \bar{\boldsymbol{x}}, \nabla f_1(\bar{\boldsymbol{x}}) \rangle$$

- **Update step:** (**Proximal Gradient Step**)

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \operatorname*{argmin}_{\boldsymbol{x}} \; f_0(\boldsymbol{x}) + f_1(\boldsymbol{x}_k) + \langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f_1(\boldsymbol{x}_k) \rangle + \frac{1}{2\tau} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2 \\ &= \operatorname{prox}_{\tau f_0}(\boldsymbol{x}_k - \tau \nabla f_1(\boldsymbol{x}_k)) \end{aligned}$$

Proximal Gradient Method and Mirror Descent

- **Optimization problem**:

$$\min_{\boldsymbol{x}} \underbrace{f_0(\boldsymbol{x})}_{\text{simple}} + \underbrace{f_1(\boldsymbol{x})}_{\substack{\text{smooth} \\ \text{non-convex}}}$$

- **Model function**:

$$f_{\bar{\boldsymbol{x}}}(\boldsymbol{x}) = f_0(\boldsymbol{x}) + f_1(\bar{\boldsymbol{x}}) + \langle \boldsymbol{x} - \bar{\boldsymbol{x}}, \nabla f_1(\bar{\boldsymbol{x}}) \rangle$$

- **Update step:** (**Bregman Proximal Gradient Method**)

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \operatorname*{argmin}_{\boldsymbol{x}} \; f_0(\boldsymbol{x}) + f_1(\boldsymbol{x}_k) + \langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f_1(\boldsymbol{x}_k) \rangle + D_h(\boldsymbol{x}, \boldsymbol{x}_k) \\ &= \operatorname{prox}^h_{\tau f_0}(\boldsymbol{x}_k - \tau \nabla f_1(\boldsymbol{x}_k)) \end{aligned}$$

Examples of Bregman functions

Bregman distance generated by a Legendre function h:

$$D_h(\boldsymbol{x}, \bar{\boldsymbol{x}}) = h(\boldsymbol{x}) - h(\bar{\boldsymbol{x}}) - \langle \nabla h(\bar{\boldsymbol{x}}), \boldsymbol{x} - \bar{\boldsymbol{x}} \rangle$$

Examples: (in terms of the Legendre function / kernel function)

- Energy: $h(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|^2$
- Burg's entropy: $h(\boldsymbol{x}) = -\sum_i \log(\boldsymbol{x}_i)$
- Boltzmann–Shannon entropy: $h(\boldsymbol{x}) = \sum_i \boldsymbol{x}_i \log(\boldsymbol{x}_i)$
- Polynomials of type: $h(\boldsymbol{x}) = \sum_{j=1}^{m} a_i \|\boldsymbol{x}\|_2^{2j}$

  For example in Phase retrieval: $h(\boldsymbol{x}) = \frac{1}{4}\|\boldsymbol{x}\|_2^4 + \frac{1}{2}\|\boldsymbol{x}\|_2^2$; (e.g. [Bolte et al. 2018]).
- Fermi-Dirac: $h(x) = x\log(x) + (1-x)\log(1-x)$
- Hellinger function: $h(x) = -\sqrt{1-x^2}$

Examples of Bregman functions

Matrix Factorization (**non-smooth, non-convex**)

$$\min_{\boldsymbol{X},\boldsymbol{Y}} \ \tfrac{1}{2}\|\boldsymbol{A}-\boldsymbol{X}\boldsymbol{Y}\|^2 + R_1(\boldsymbol{X}) + R_2(\boldsymbol{Y})$$

- $\frac{1}{2}\|\boldsymbol{A}-\boldsymbol{X}\boldsymbol{Y}\|^2$ does not have Lipschitz gradient.
- Line search can be too expensive (requires matrix multiplication).
- *Alternating Minimization:* limited convergence guarantees, sometimes biased, not applicable for $\boldsymbol{X}=\boldsymbol{Y}$.

Examples of Bregman functions

Matrix Factorization (**non-smooth, non-convex**)

$$\min_{\boldsymbol{X},\boldsymbol{Y}} \tfrac{1}{2}\|\boldsymbol{A}-\boldsymbol{X}\boldsymbol{Y}\|^2 + R_1(\boldsymbol{X}) + R_2(\boldsymbol{Y})$$

- $\frac{1}{2}\|\boldsymbol{A}-\boldsymbol{X}\boldsymbol{Y}\|^2$ does not have Lipschitz gradient.
- Line search can be too expensive (requires matrix multiplication).
- *Alternating Minimization:* limited convergence guarantees, sometimes biased, not applicable for $\boldsymbol{X} = \boldsymbol{Y}$.

Apply Model BPG: (**= Bregman Proximal Gradient Method**) ([Bolte et al. 2018])

Examples of Bregman functions

Matrix Factorization (**non-smooth, non-convex**)

$$\min_{\boldsymbol{X},\boldsymbol{Y}} \ \tfrac{1}{2}\|\boldsymbol{A}-\boldsymbol{X}\boldsymbol{Y}\|^2 + R_1(\boldsymbol{X}) + R_2(\boldsymbol{Y})$$

- $\tfrac{1}{2}\|\boldsymbol{A}-\boldsymbol{X}\boldsymbol{Y}\|^2$ does not have Lipschitz gradient.
- Line search can be too expensive (requires matrix multiplication).
- *Alternating Minimization:* limited convergence guarantees, sometimes biased, not applicable for $\boldsymbol{X}=\boldsymbol{Y}$.

Apply Model BPG: (**= Bregman Proximal Gradient Method**) ([Bolte et al. 2018])

- **New Bregman Distance:** [Mukkamala, O. 19] (see also [Teboulle, Vaisbourd 20])

$$h(\boldsymbol{X},\boldsymbol{Y}) = 3\left(\frac{\|\boldsymbol{X}\|_F^2+\|\boldsymbol{Y}\|_F^2}{2}\right)^2 + \|\boldsymbol{A}\|_F\left(\frac{\|\boldsymbol{X}\|_F^2+\|\boldsymbol{Y}\|_F^2}{2}\right).$$

- Symmetric MF setting developed in [Dragomir, Bolte, d'Aspremont 19].

Deep Linear Neural Networks or Deep Matrix Factorization:

$$\min_{W_1,\dots,W_N} \frac{1}{2}\|\boldsymbol{Y} - W_1 W_2 \cdots W_N \boldsymbol{X}\|_F^2 + \sum_{i=1}^{N} \mathcal{R}_i(W_i)$$

Deep Linear Neural Networks or Deep Matrix Factorization:

$$\min_{W_1,\dots,W_N} \frac{1}{2}\|\boldsymbol{Y} - W_1 W_2 \cdots W_N \boldsymbol{X}\|_F^2 + \sum_{i=1}^{N} \mathcal{R}_i(W_i)$$

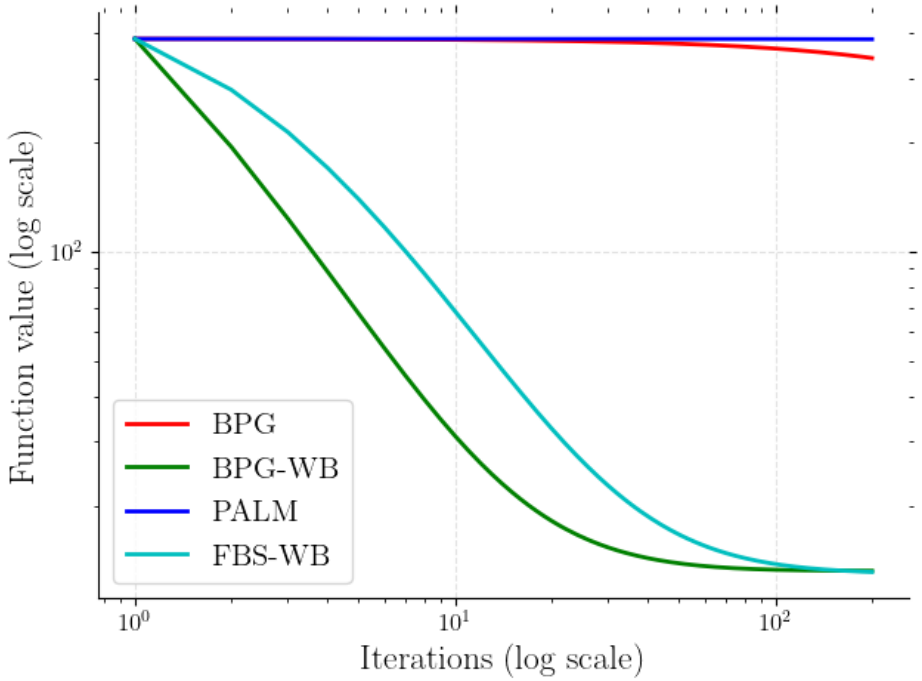
- Write: $\boldsymbol{W} := (W_1, \dots, W_N)$ and $\|\boldsymbol{W}\|_F^2 := \sum_{i=1}^N \|W_i\|_F^2$.
- [Mukkamala et al. 2021] shows relative smoothness w.r.t.

$$h(\boldsymbol{W}) = \begin{cases} \frac{\|\boldsymbol{X}\|_F^2}{N^{N-2}}\|\boldsymbol{W}\|_F^{2N} + \frac{\|\boldsymbol{Y}\|_F\|\boldsymbol{X}\|_F}{(N-2)^{\frac{N-2}{2}}}\|\boldsymbol{W}\|_F^N, & \text{if } N \text{ is even} \\ \frac{\|\boldsymbol{X}\|_F^2}{N^{N-2}}\|\boldsymbol{W}\|_F^{2N} + \frac{\|\boldsymbol{Y}\|_F\|\boldsymbol{X}\|_F}{(N-1)^{\frac{N-1}{2}}}\left(\|\boldsymbol{W}\|_F^2 + 1\right)^{\frac{N+1}{2}}, & \text{if } N \text{ is odd}. \end{cases}$$

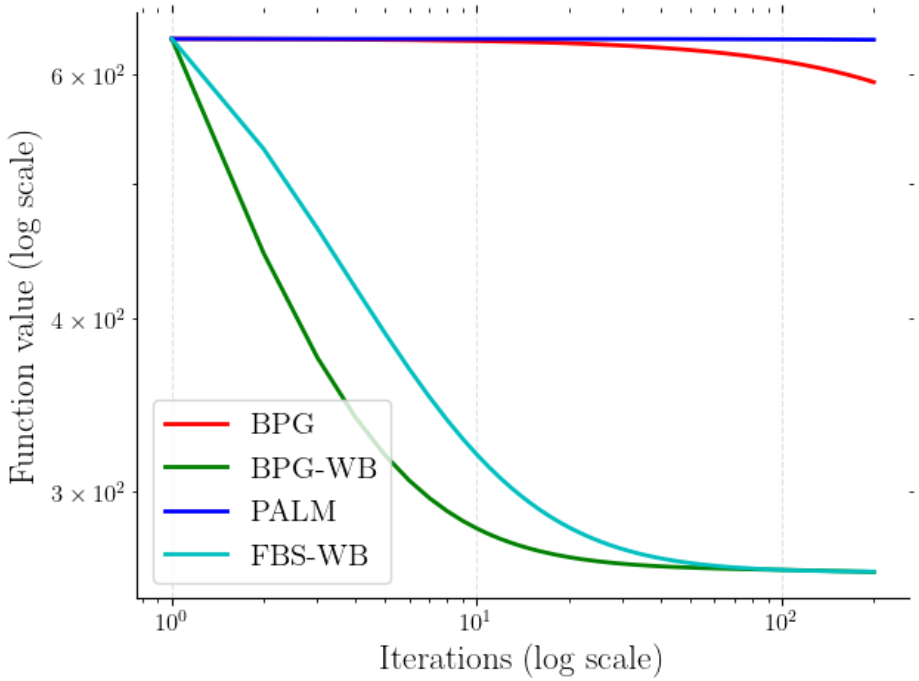
optimization / training with **constant step size rule** using BPG.

- **Non-linear Neural Networks:** see [Mukkamala, PhD Thesis, 2021].

Deep Matrix Completion on MovieLens Datasets:



L2-Regularization ($N = 4$)
MovieLens-100K



L1-Regularization ($N = 4$)
MovieLens-100K

Clipped Proximal Gradient Method

- **Optimization problem**:

$$\min_{\boldsymbol{x}} \underbrace{f_0(\boldsymbol{x})}_{\text{simple}} + \underbrace{f_1(\boldsymbol{x})}_{\substack{\text{smooth} \\ \text{non-convex} \\ \textbf{non-negative}}}$$

- **Model function**:

$$f_{\bar{\boldsymbol{x}}}(\boldsymbol{x}) = f_0(\boldsymbol{x}) + \max\Big(0, f_1(\bar{\boldsymbol{x}}) + \langle \boldsymbol{x} - \bar{\boldsymbol{x}}, \nabla f_1(\bar{\boldsymbol{x}}) \rangle\Big)$$

Examples of Model Functions

Proximal Newton Method

- **Optimization problem**:

$$\min_{\boldsymbol{x}} \underbrace{f_0(\boldsymbol{x})}_{\text{simple}} + \underbrace{f_1(\boldsymbol{x})}_{\substack{C^2-\text{smooth} \\ \text{strongly convex}}}$$

- **Model function**:

$$f_{\bar{\boldsymbol{x}}}(\boldsymbol{x}) = f_0(\boldsymbol{x}) + f_1(\bar{\boldsymbol{x}}) + \langle \boldsymbol{x} - \bar{\boldsymbol{x}}, \nabla f_1(\bar{\boldsymbol{x}}) \rangle + \langle \boldsymbol{x} - \bar{\boldsymbol{x}}, \nabla^2 f_1(\bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}}) \rangle$$

Examples of Model Functions

ProxDescent [Lewis and Wright, 2016], [Drusvyatskiy and Lewis, 2016]

- **Optimization problem**:

$$\min_{\boldsymbol{x}} \quad \underbrace{f_0(\boldsymbol{x})}_{\substack{\text{non-smooth}\\ \text{convex}}} + \underbrace{g\Big(}_{\substack{\text{non-smooth}\\ \text{convex}\\ \text{finite-valued}}} \underbrace{F(\boldsymbol{x})}_{\text{smooth}} \Big)$$

- **Model function**: ($DF(\bar{\boldsymbol{x}})$ is the Jacobian matrix of F at $\bar{\boldsymbol{x}}$)

$$f_{\bar{\boldsymbol{x}}}(\boldsymbol{x}) = f_0(\boldsymbol{x}) + g\Big(F(\bar{\boldsymbol{x}}) + DF(\bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})\Big)$$

Example: Levenberg–Marquardt Algorithm for Non-linear least-squares problem

More Examples:

- Outer-linearization of the composite problem:

$$\min_{\boldsymbol{x}} \; \underbrace{f_0(\boldsymbol{x})}_{\substack{\text{non-smooth} \\ \text{convex}}} + \underbrace{g\Big(}_{\substack{\text{smooth} \\ \text{non-convex} \\ \text{non-decreasing}}} \underbrace{F(\boldsymbol{x})}_{\substack{\text{non-smooth} \\ \text{coordinate-wise} \\ \text{convex}}} \Big)$$

- Combine previous concepts of model functions, e.g. for block problems.
- Higher order approximations.

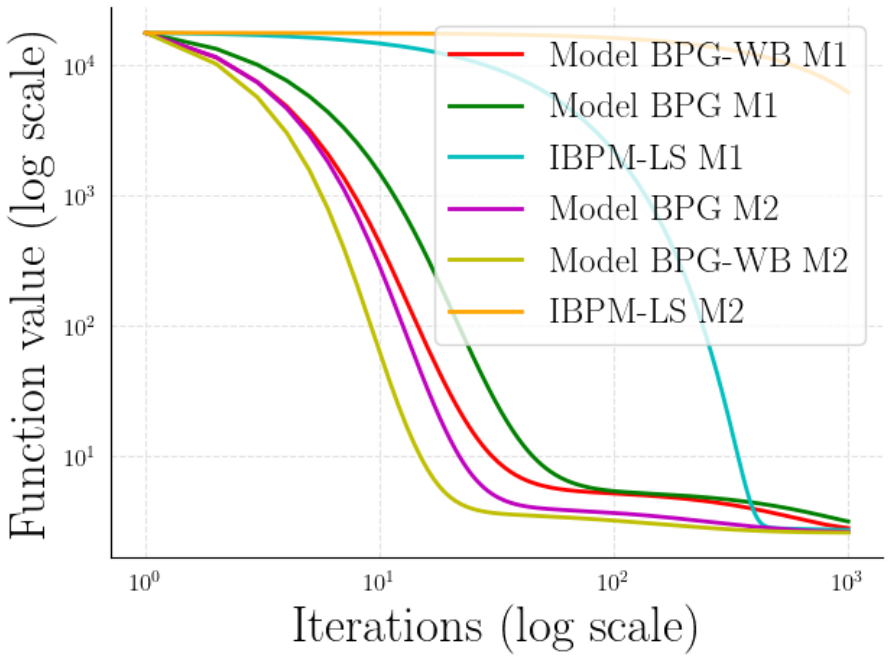
Be creative! Design good model functions for **your** problem.

Comparison of Model Functions for a Phase Retrieval problem

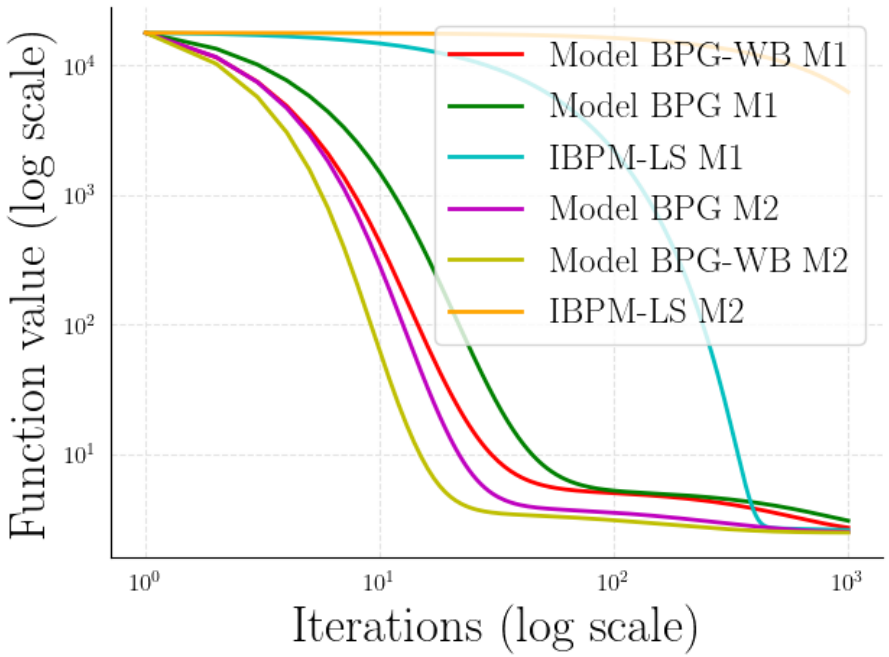
$$\min_{\boldsymbol{x}} \frac{1}{M} \sum_{i=1}^{M} (\boldsymbol{x}^\top A_i \boldsymbol{x} - \boldsymbol{b}_i)^2 + \mathcal{R}(\boldsymbol{x})$$

- **Model 1:** Bregman Proximal Gradient Model.
- **Model 2:** Non-linear composite ($\sim$ProxDescent): Linearize inner part of $|(\boldsymbol{x}^\top A_i \boldsymbol{x} - \boldsymbol{b}_i)^2|$.

For both the model functions we use $h(\boldsymbol{x}) = \frac{1}{4}\|\boldsymbol{x}\|^4 + \frac{1}{2}\|\boldsymbol{x}\|^2$.



L1 regulariztion



Squared L2 regularization

Model BPG with Model 2 is faster in both the settings.

Outline

Conclusion of Part 1:

- unified flexible algorithm with several interesting special cases
- convergence in non-smooth non-convex setting to stationary point
- do not require line search

However:

If problem size permits, line search / backtracking can yield a significant speed up.

Motivation for Part 2: Improve practical performance

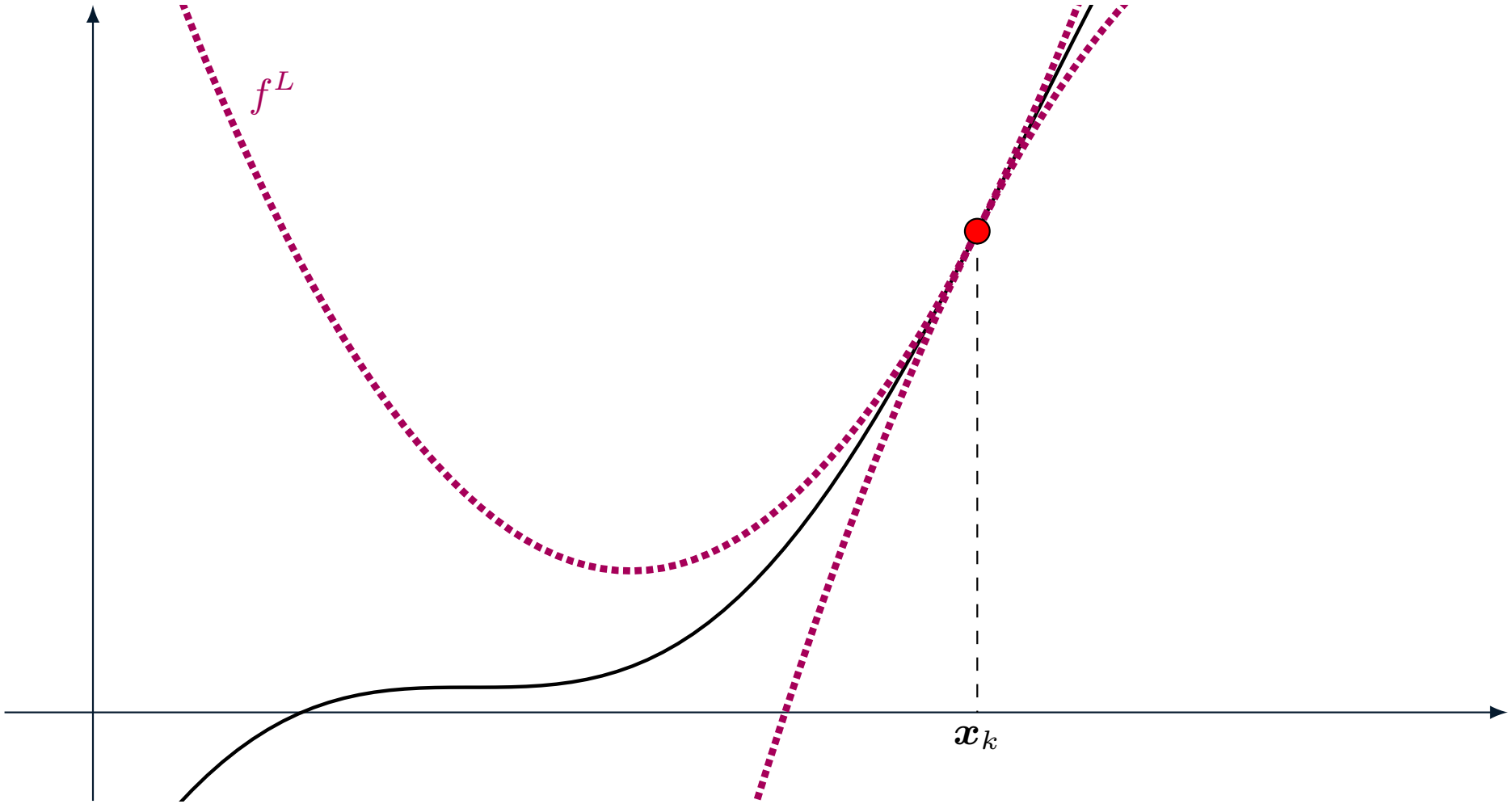
- Adaptive Bregman distances (similar to quasi-Newton methods).
- Acceleration / Momentum (**this talk**) based on Nesterov extrapolation:

$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \\ \boldsymbol{x}_{k+1} &= \boldsymbol{y}_k - \tau \nabla f(\boldsymbol{y}_k) \end{aligned}$$

  with novel COnvex ConcAve INertial backtracking procedure.

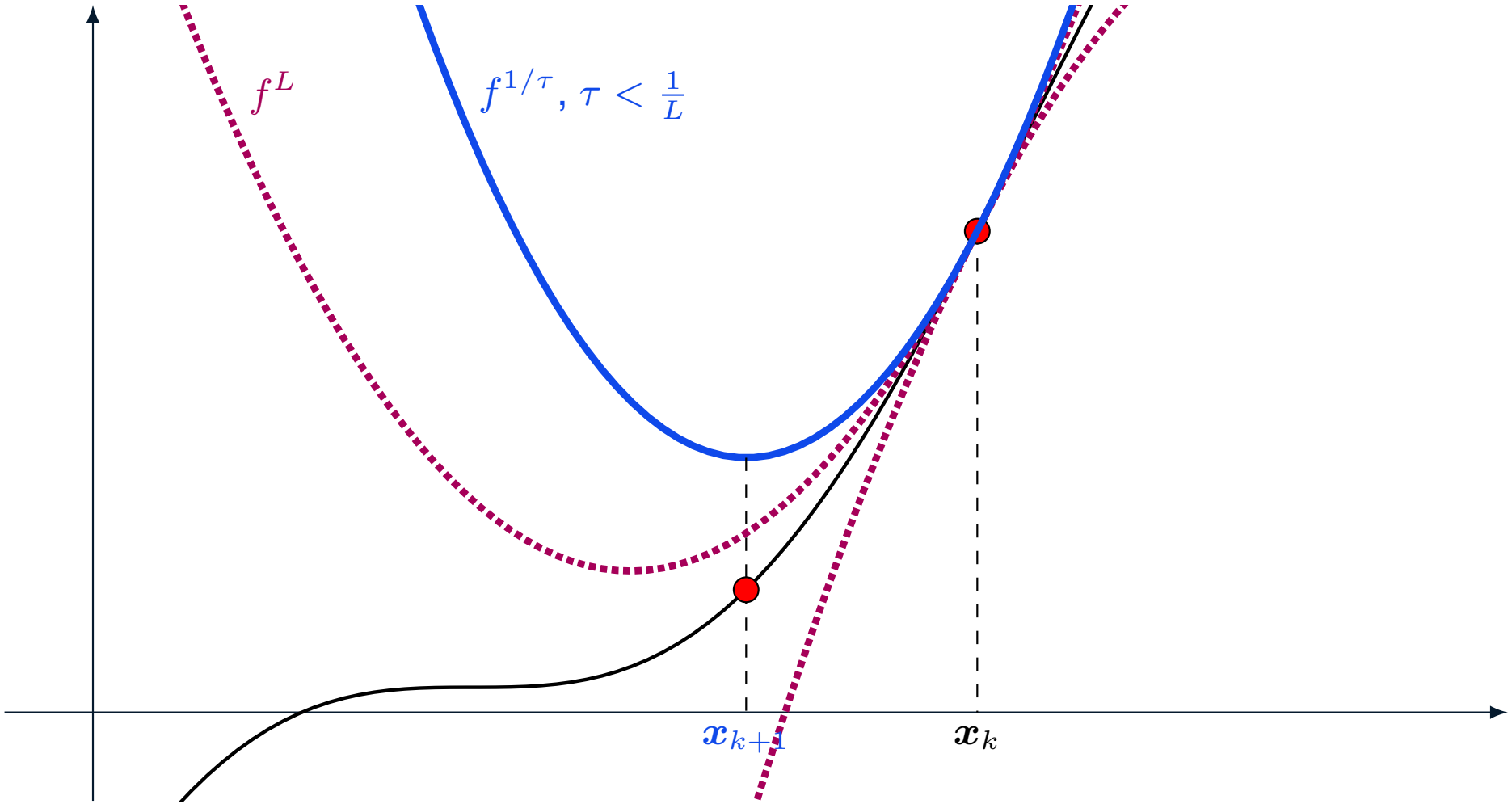
Gradient Descent: Sufficient decrease

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{1}{2\tau} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2}_{=: f^{1/\tau}(\boldsymbol{x}; \boldsymbol{x}_k)} .$$



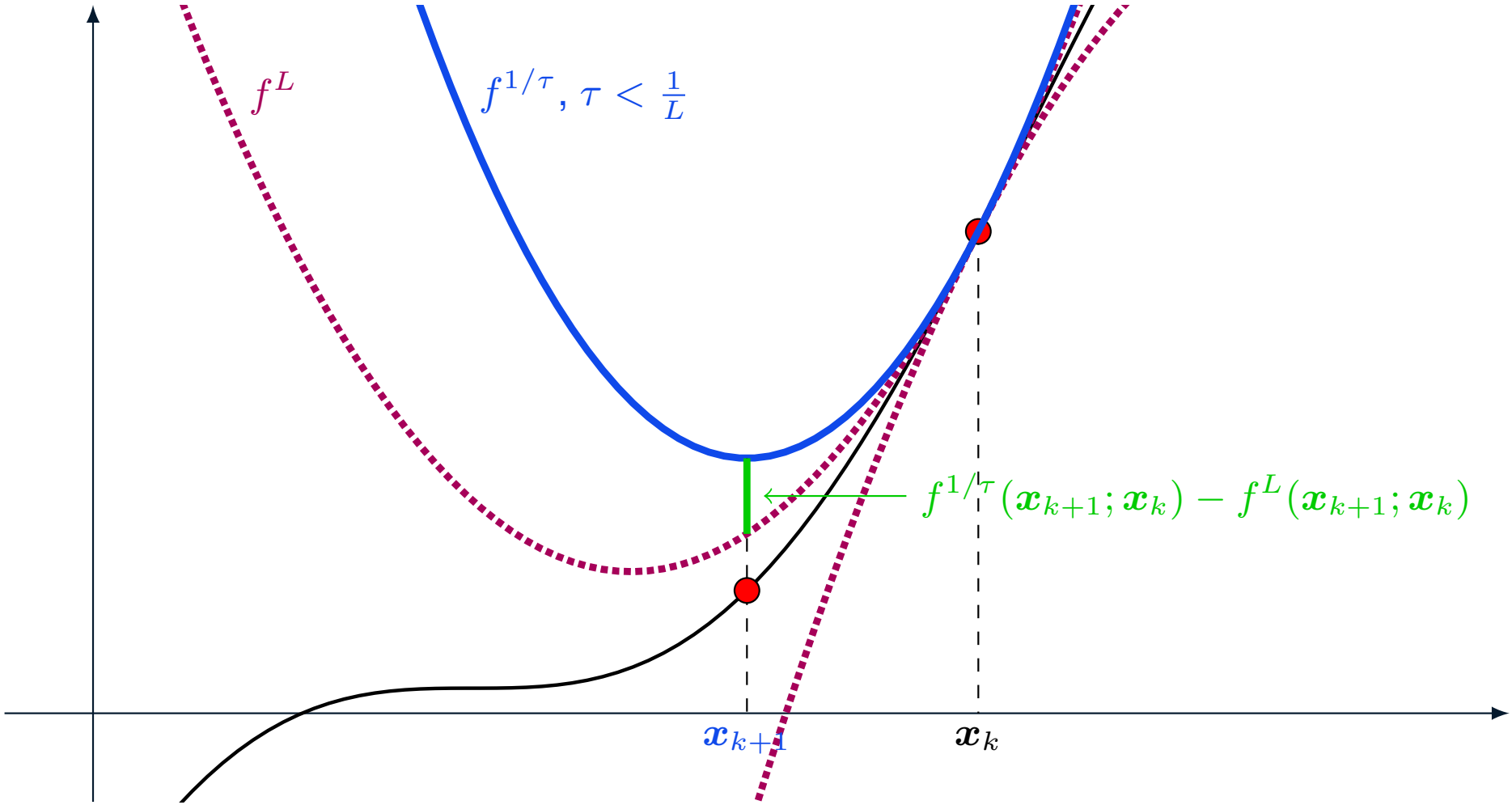
Gradient Descent: Sufficient decrease

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{1}{2\tau} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2}_{=: f^{1/\tau}(\boldsymbol{x}; \boldsymbol{x}_k)} .$$



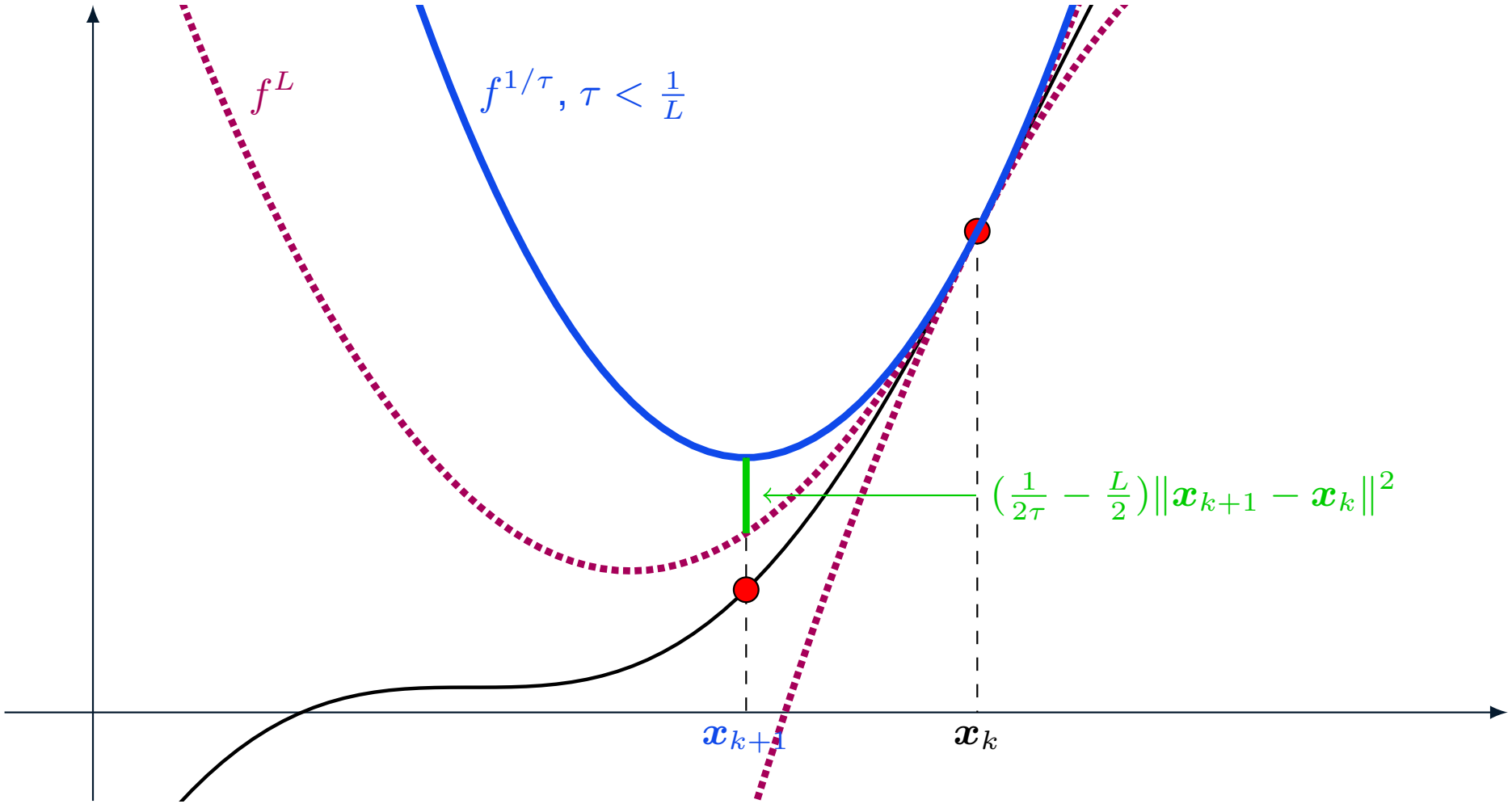
Gradient Descent: Sufficient decrease

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{1}{2\tau} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2}_{=: f^{1/\tau}(\boldsymbol{x}; \boldsymbol{x}_k)} .$$



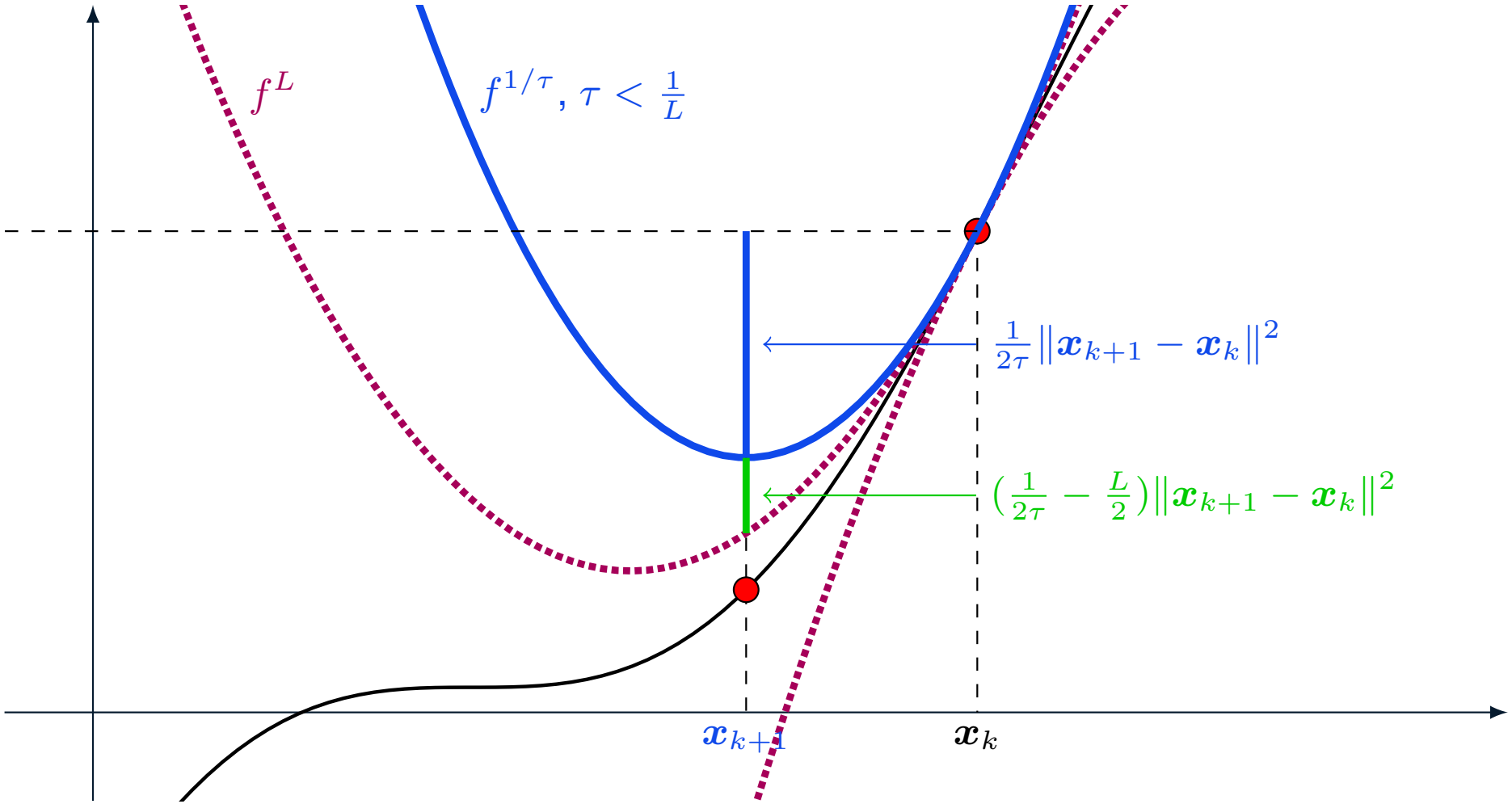
Gradient Descent: Sufficient decrease

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{1}{2\tau} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2}_{=: f^{1/\tau}(\boldsymbol{x}; \boldsymbol{x}_k)} .$$

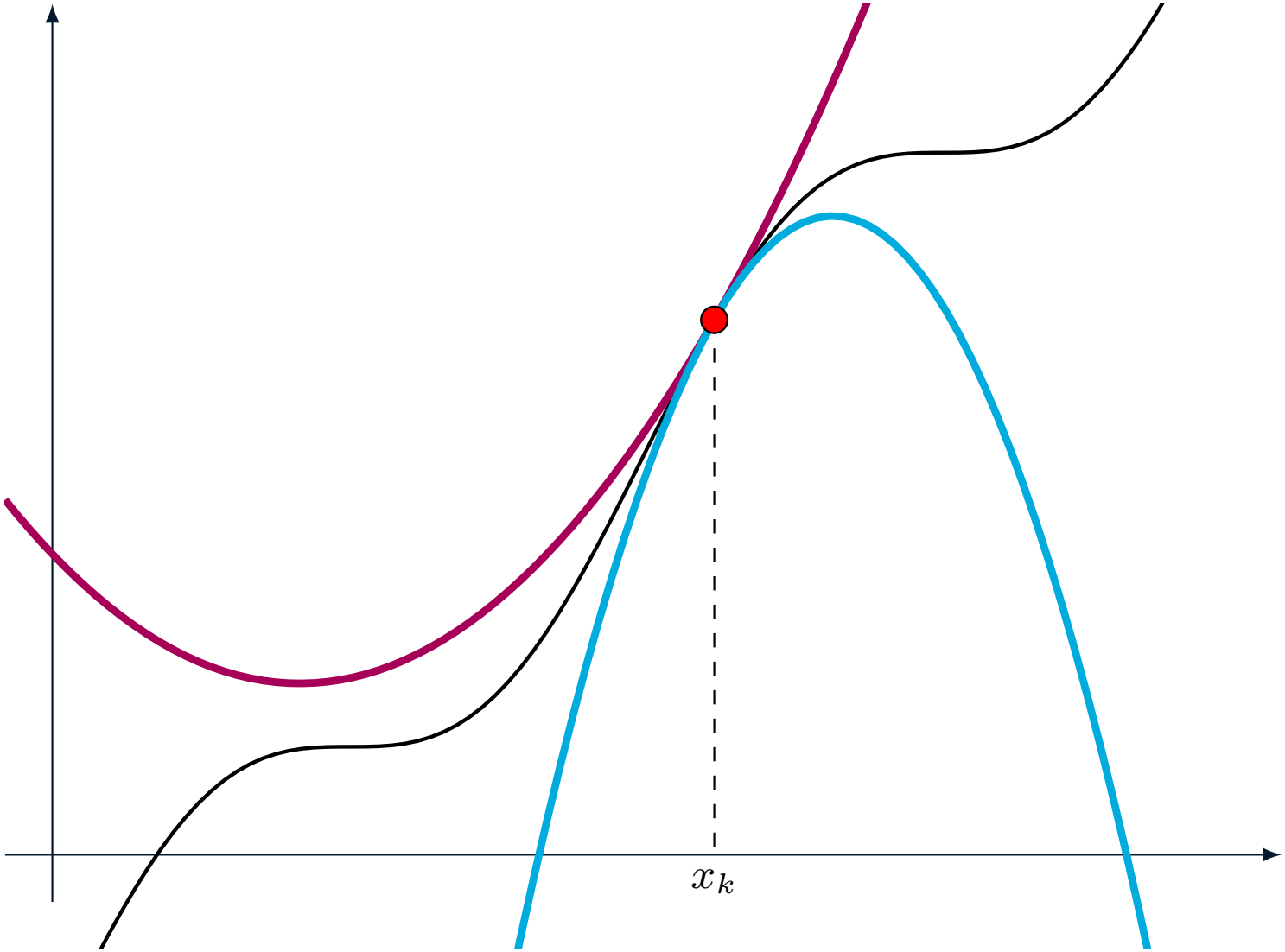


Gradient Descent: Sufficient decrease

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{1}{2\tau} \|\boldsymbol{x} - \boldsymbol{x}_k\|^2}_{=: f^{1/\tau}(\boldsymbol{x}; \boldsymbol{x}_k)} .$$



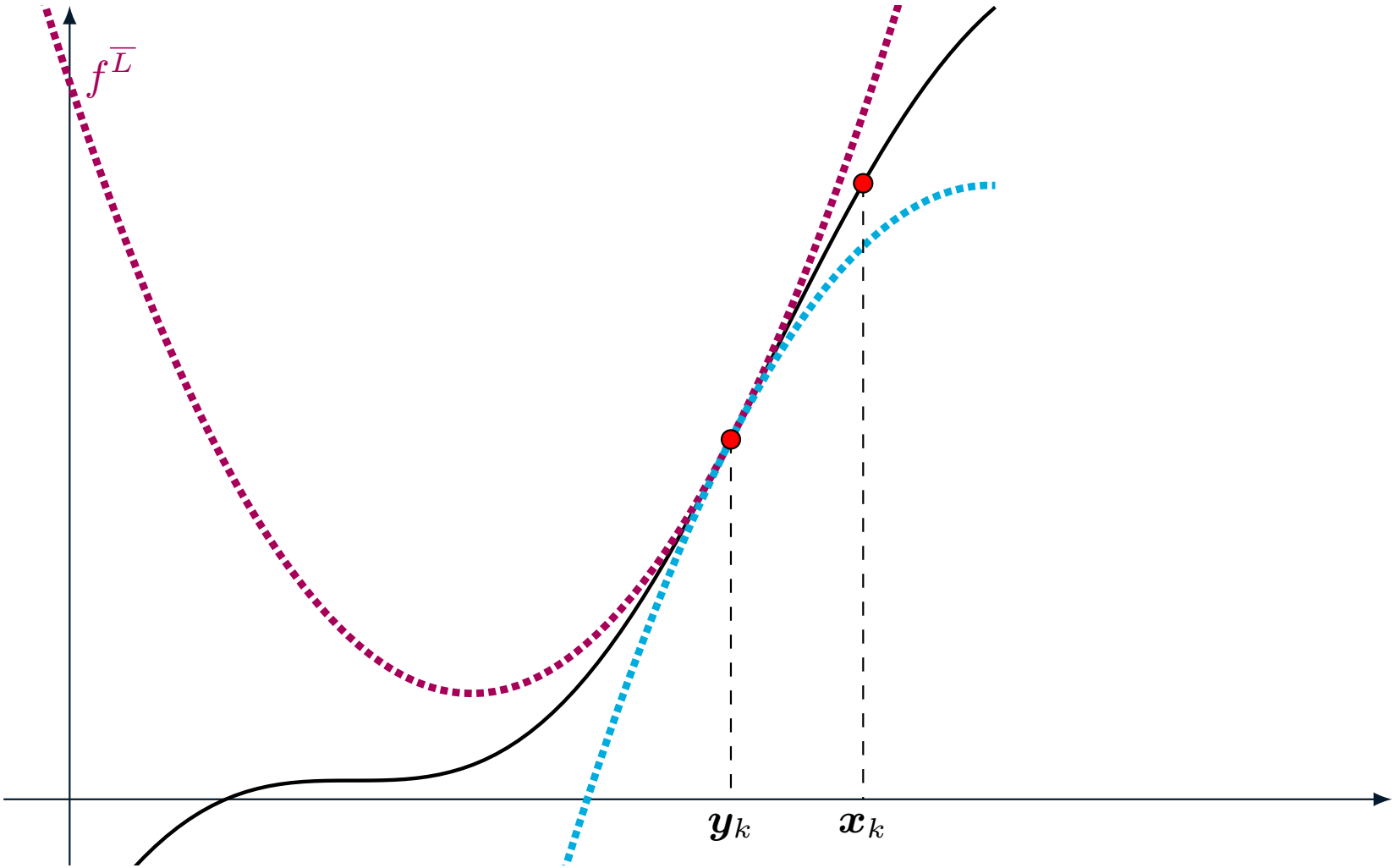
Different Constants for Tight Upper and Lower Quadratic Bounds



$$-\frac{\underline{L}}{2}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2 \leq f(\boldsymbol{x}) - f(\boldsymbol{x}_k) - \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x}-\boldsymbol{x}_k\rangle \leq \frac{\overline{L}}{2}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2$$

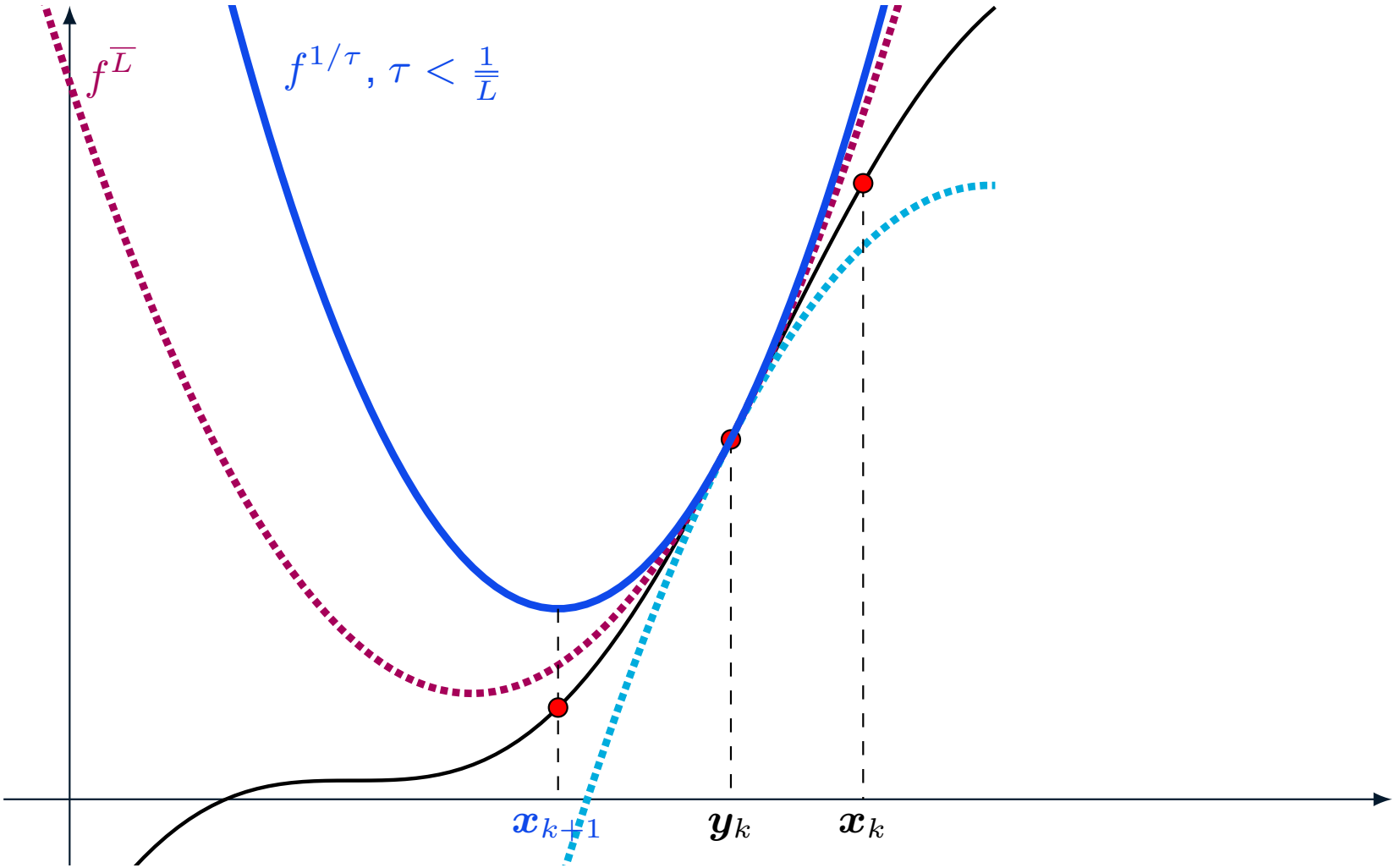
Nesterov's Extrapolation: Sufficient decrease

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \frac{1}{2\tau} \|\boldsymbol{x} - \boldsymbol{y}_k\|^2}_{=: f^{1/\tau}(\boldsymbol{x}; \boldsymbol{y}_k)}, \quad \boldsymbol{y}_k = \boldsymbol{x}_k + \gamma_k (\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$



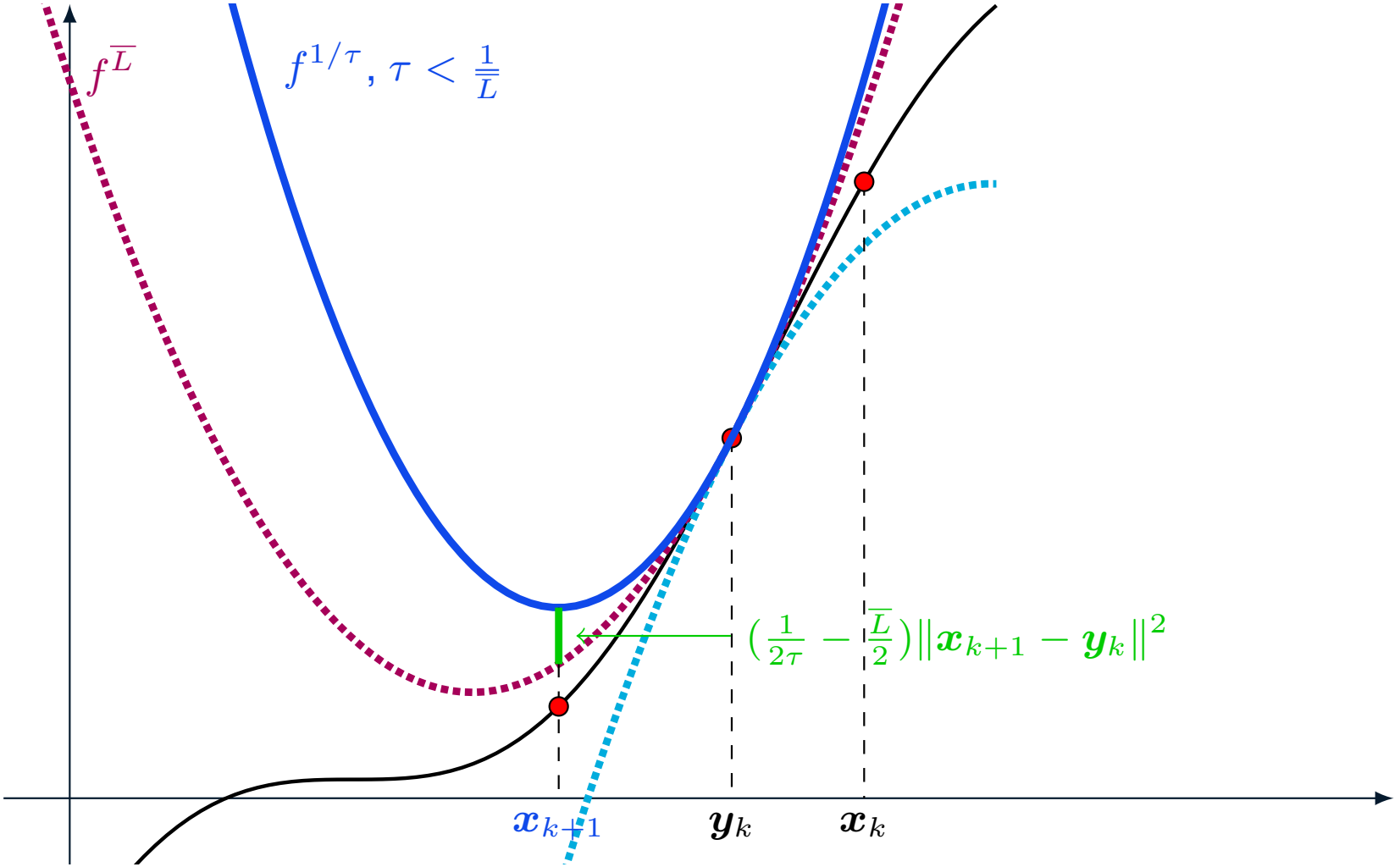
Nesterov's Extrapolation: Sufficient decrease

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \frac{1}{2\tau} \|\boldsymbol{x} - \boldsymbol{y}_k\|^2}_{=: f^{1/\tau}(\boldsymbol{x}; \boldsymbol{y}_k)}, \quad \boldsymbol{y}_k = \boldsymbol{x}_k + \gamma_k (\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$



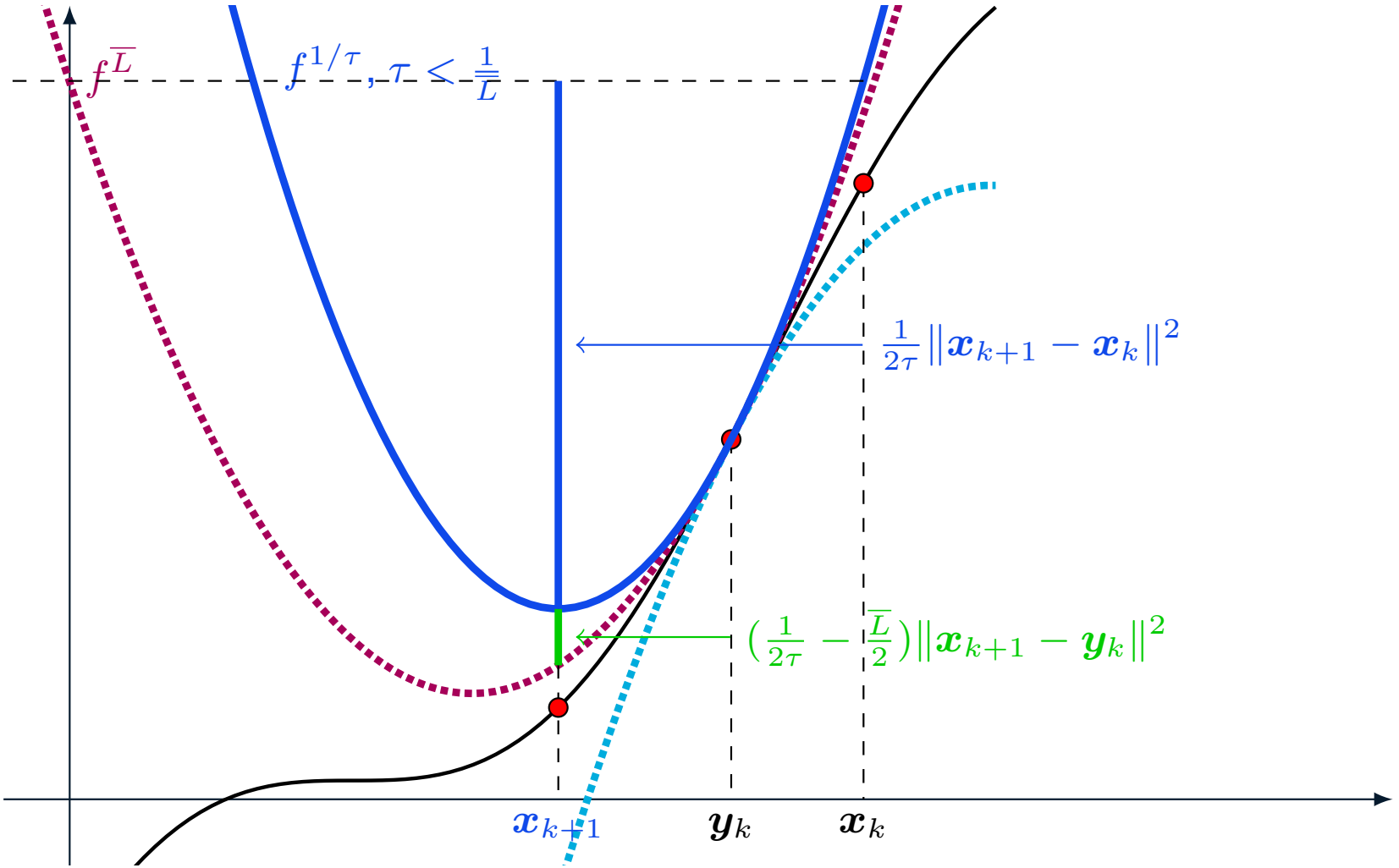
Nesterov's Extrapolation: Sufficient decrease

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \frac{1}{2\tau}\|\boldsymbol{x} - \boldsymbol{y}_k\|^2}_{=: f^{1/\tau}(\boldsymbol{x}; \boldsymbol{y}_k)}, \quad \boldsymbol{y}_k = \boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$



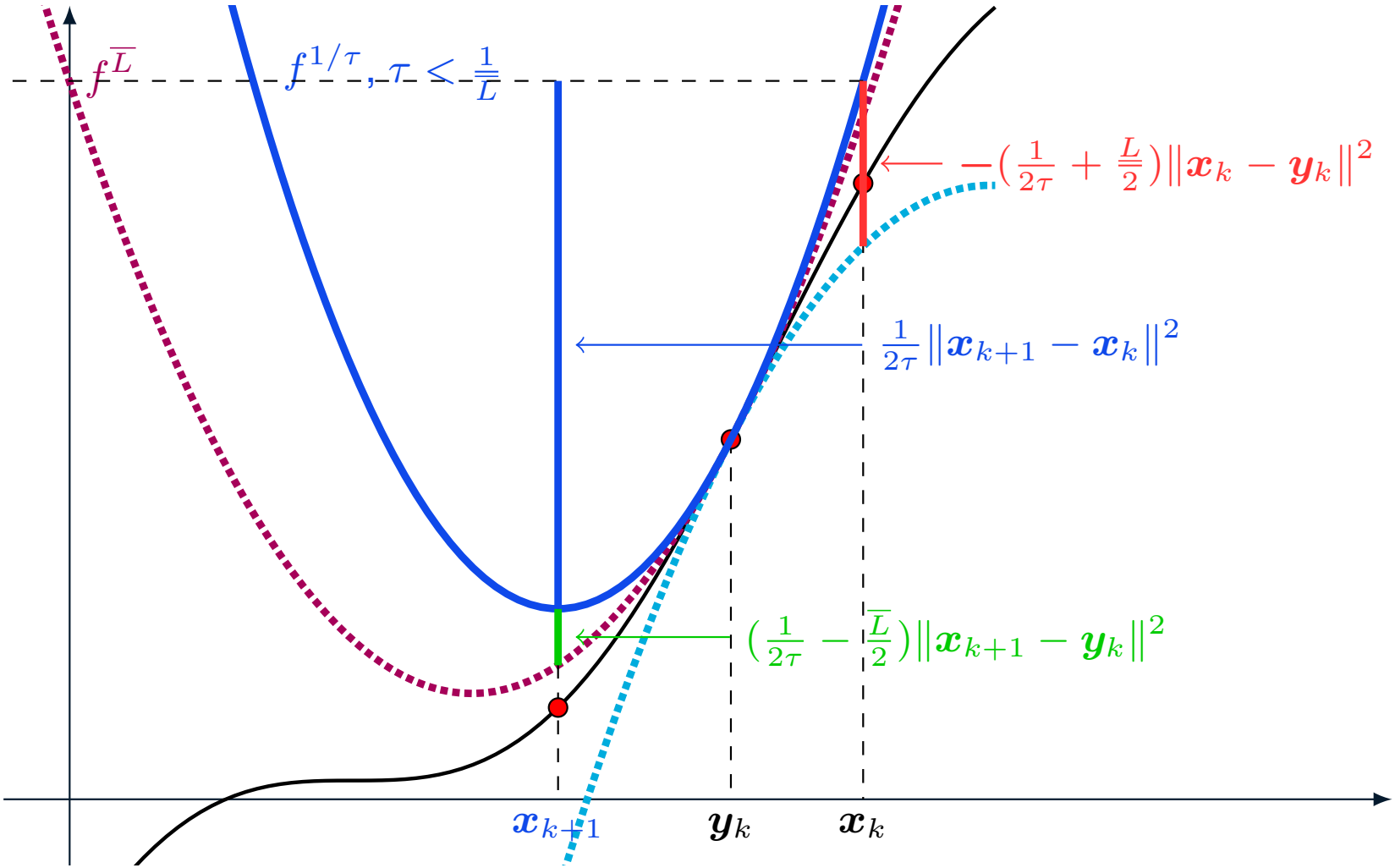
Nesterov's Extrapolation: Sufficient decrease

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \frac{1}{2\tau}\|\boldsymbol{x} - \boldsymbol{y}_k\|^2}_{=: f^{1/\tau}(\boldsymbol{x}; \boldsymbol{y}_k)}, \quad \boldsymbol{y}_k = \boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$



Nesterov's Extrapolation: Sufficient decrease

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \frac{1}{2\tau}\|\boldsymbol{x} - \boldsymbol{y}_k\|^2}_{=: f^{1/\tau}(\boldsymbol{x}; \boldsymbol{y}_k)}, \quad \boldsymbol{y}_k = \boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$



Proof of Convergence for Nesterov's Extrapolation

- From geometric derivation:

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) + \left(\frac{L}{2} + \frac{1}{2\tau}\right)\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 - \frac{1}{2\tau}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2 .$$

Proof of Convergence for Nesterov's Extrapolation

- From geometric derivation:

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) + \left(\frac{L}{2} + \frac{1}{2\tau}\right)\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 - \frac{1}{2\tau}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2 .$$

- Define $F_\tau(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k) := f(\boldsymbol{x}_{k+1}) + \frac{1}{2\tau}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2$, then this is equivalent to

$$F_\tau(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k) \leq F_\tau(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}) + \left(\frac{L}{2} + \frac{1}{2\tau}\right)\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 - \frac{1}{2\tau}\|\boldsymbol{x}_{k-1} - \boldsymbol{x}_k\|^2$$

Proof of Convergence for Nesterov's Extrapolation

- From geometric derivation:

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) + \left(\frac{L}{2} + \frac{1}{2\tau}\right)\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 - \frac{1}{2\tau}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2 .$$

- Define $F_\tau(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k) := f(\boldsymbol{x}_{k+1}) + \frac{1}{2\tau}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2$, then this is equivalent to

$$F_\tau(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k) \leq F_\tau(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}) + \left(\frac{L}{2} + \frac{1}{2\tau}\right)\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 - \frac{1}{2\tau}\|\boldsymbol{x}_{k-1} - \boldsymbol{x}_k\|^2$$

Decrease Condition: $\left(\frac{L}{2} + \frac{1}{2\tau}\right)\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 - \frac{1}{2\tau}\|\boldsymbol{x}_{k-1} - \boldsymbol{x}_k\|^2 \leq 0$

Proof of Convergence for Nesterov's Extrapolation

- From geometric derivation:

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) + \left(\frac{\underline{L}}{2} + \frac{1}{2\tau}\right)\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 - \frac{1}{2\tau}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2 .$$

- Define $F_\tau(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k) := f(\boldsymbol{x}_{k+1}) + \frac{1}{2\tau}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2$, then this is equivalent to

$$F_\tau(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k) \leq F_\tau(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}) + \left(\frac{\underline{L}}{2} + \frac{1}{2\tau}\right)\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 - \frac{1}{2\tau}\|\boldsymbol{x}_{k-1} - \boldsymbol{x}_k\|^2$$

Decrease Condition: $\left(\frac{\underline{L}}{2} + \frac{1}{2\tau}\right)\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 - \frac{1}{2\tau}\|\boldsymbol{x}_{k-1} - \boldsymbol{x}_k\|^2 \leq 0$

- For $\boldsymbol{y}_k = \boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$:

$$\sup_k \gamma_k^2\left(\frac{\underline{L}}{2} + \frac{1}{2\tau}\right) < \frac{1}{2\tau} \iff \sup_k \gamma_k^2 < \frac{\tau^{-1}}{\underline{L} + \tau^{-1}} \overset{\boxed{\tau^{-1} = \overline{L}}}{=} \frac{\overline{L}}{\underline{L} + \overline{L}} \overset{\boxed{\underline{L} = \overline{L}}}{=} \frac{1}{2} .$$

[Wen, Chen, Pong 2017]

- For f convex, i.e., $\underline{L} = 0$, we obtain convergence for $\sup_{k\in\mathbb{N}} \gamma_k < 1$.

Proof of Convergence for Nesterov's Extrapolation

- From geometric derivation:

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) + \left(\frac{\underline{L}}{2} + \frac{1}{2\tau}\right)\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 - \frac{1}{2\tau}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2 .$$

- Define $F_\tau(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k) := f(\boldsymbol{x}_{k+1}) + \frac{1}{2\tau}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2$, then this is equivalent to

$$F_\tau(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k) \leq F_\tau(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}) + \left(\frac{\underline{L}}{2} + \frac{1}{2\tau}\right)\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 - \frac{1}{2\tau}\|\boldsymbol{x}_{k-1} - \boldsymbol{x}_k\|^2$$

Decrease Condition: $\left(\frac{\underline{L}}{2} + \frac{1}{2\tau}\right)\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 - \frac{1}{2\tau}\|\boldsymbol{x}_{k-1} - \boldsymbol{x}_k\|^2 \leq 0$

- For $\boldsymbol{y}_k = \boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$:

$$\sup_k \gamma_k^2\left(\frac{\underline{L}}{2} + \frac{1}{2\tau}\right) < \frac{1}{2\tau} \iff \sup_k \gamma_k^2 < \frac{\tau^{-1}}{\underline{L} + \tau^{-1}} \overset{\boxed{\tau^{-1} = \overline{L}}}{=} \frac{\overline{L}}{\underline{L} + \overline{L}} \overset{\boxed{\underline{L} = \overline{L}}}{=} \frac{1}{2} .$$

[Wen, Chen, Pong 2017]

- For f convex, i.e., $\underline{L} = 0$, we obtain convergence for $\sup_{k\in\mathbb{N}} \gamma_k < 1$.

$\rightsquigarrow$ **tight lower bounds $\underline{L}$ allow for large extrapolation.**

Convex–Concave Inertial Backtracking (CoCaIn)

Convex–Concave Inertial Backtracking: (simple version) [Mukkamala, O., Sabach, Pock, 2021]

Convex–Concave Inertial Backtracking (CoCaIn)

Convex–Concave Inertial Backtracking: (simple version) [Mukkamala, O., Sabach, Pock, 2021]

- Find $\underline{L}_k$ (via backtracking) such that

$$\boldsymbol{y}_k = \boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \quad \text{with } \gamma_k = \sqrt{\frac{\overline{L}_{k-1}}{\overline{L}_{k-1} + \underline{L}_k}} \qquad \texttt{[extrapolation step]}$$

satisfies

$$f(\boldsymbol{x}_k) \geq f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x}_k - \boldsymbol{y}_k \rangle - \frac{\underline{L}_k}{2} \|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 . \qquad \texttt{[lower bound]}$$

Convex–Concave Inertial Backtracking (CoCaIn)

Convex–Concave Inertial Backtracking: (simple version) [Mukkamala, O., Sabach, Pock, 2021]

- Find $\underline{L}_k$ (via backtracking) such that

$$\boldsymbol{y}_k = \boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \quad \text{with } \gamma_k = \sqrt{\frac{\overline{L}_{k-1}}{\overline{L}_{k-1} + \underline{L}_k}} \qquad \texttt{[extrapolation step]}$$

satisfies

$$f(\boldsymbol{x}_k) \geq f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x}_k - \boldsymbol{y}_k \rangle - \frac{\underline{L}_k}{2} \|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 . \qquad \texttt{[lower bound]}$$

- Find $\overline{L}_k \geq \overline{L}_{k-1}$ (via backtracking) such that

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \frac{\overline{L}_k}{2} \|\boldsymbol{x} - \boldsymbol{y}_k\|^2 \qquad \texttt{[update]}$$

satisfies

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x}_{k+1} - \boldsymbol{y}_k \rangle + \frac{\overline{L}_k}{2} \|\boldsymbol{x}_{k+1} - \boldsymbol{y}_k\|^2 . \qquad \texttt{[upper bound]}$$

Convex–Concave Inertial Backtracking (CoCaIn)

Convex–Concave Inertial Backtracking: $\min_{\boldsymbol{x}} f(\boldsymbol{x})$

- Find $\underline{L}_k$ and γ_k (via backtracking) such that

$$\boldsymbol{y}_k = \boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \qquad \texttt{[extrapolation step]}$$

satisfies

$$\begin{cases} f(\boldsymbol{x}_k) \geq f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x}_k - \boldsymbol{y}_k \rangle - \dfrac{\underline{L}_k}{2}\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 & \texttt{[lower bound]} \\ \|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 \leq \dfrac{\overline{L}_{k-1}}{\overline{L}_{k-1} + \underline{L}_k}\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|^2 . & \texttt{[extrapolation bound]} \end{cases}$$

- Find $\overline{L}_k \geq \overline{L}_{k-1}$ (via backtracking) such that

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \frac{\overline{L}_k}{2}\|\boldsymbol{x} - \boldsymbol{y}_k\|^2 \qquad \texttt{[update]}$$

satisfies

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x}_{k+1} - \boldsymbol{y}_k \rangle + \frac{\overline{L}_k}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{y}_k\|^2 . \qquad \texttt{[upper bound]}$$

Convex–Concave Inertial Backtracking (CoCaIn)

Convex–Concave Inertial Backtracking: $\min_{\boldsymbol{x}} f(\boldsymbol{x})$

- Find $\underline{L}_k$ and γ_k (via backtracking) such that

$$\boldsymbol{y}_k = \boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \qquad \texttt{[extrapolation step]}$$

satisfies

$$\begin{cases} f(\boldsymbol{x}_k) \geq f_{\boldsymbol{y}_k}(\boldsymbol{x}_k) - \dfrac{\underline{L}_k}{2}\|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 & \texttt{[lower bound]} \\ \|\boldsymbol{x}_k - \boldsymbol{y}_k\|^2 \leq \dfrac{\overline{L}_{k-1}}{\overline{L}_{k-1} + \underline{L}_k}\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|^2 . & \texttt{[extrapolation bound]} \end{cases}$$

- Find $\overline{L}_k \geq \overline{L}_{k-1}$ (via backtracking) such that

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} f_{\boldsymbol{y}_k}(\boldsymbol{x}) + \frac{\overline{L}_k}{2}\|\boldsymbol{x} - \boldsymbol{y}_k\|^2 \qquad \texttt{[update]}$$

satisfies

$$f(\boldsymbol{x}_{k+1}) \leq f_{\boldsymbol{y}_k}(\boldsymbol{x}_{k+1}) + \frac{\overline{L}_k}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{y}_k\|^2 . \qquad \texttt{[upper bound]}$$

Convex–Concave Inertial Backtracking (CoCaIn)

Convex–Concave Inertial Backtracking: $\min_{\boldsymbol{x}} f(\boldsymbol{x})$

- Find $\underline{L}_k$ and γ_k (via backtracking) such that

$$\boldsymbol{y}_k = \boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \qquad \texttt{[extrapolation step]}$$

satisfies

$$\begin{cases} f(\boldsymbol{x}_k) \geq f_{\boldsymbol{y}_k}(\boldsymbol{x}_k) - \underline{L}_k D_h(\boldsymbol{x}_k, \boldsymbol{y}_k) & \texttt{[lower bound]} \\ D_h(\boldsymbol{x}_k, \boldsymbol{y}_k) \leq \dfrac{\overline{L}_{k-1}}{\overline{L}_{k-1} + \underline{L}_k} D_h(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k)\,. & \texttt{[extrapolation bound]} \end{cases}$$

- Find $\overline{L}_k \geq \overline{L}_{k-1}$ (via backtracking) such that

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} f_{\boldsymbol{y}_k}(\boldsymbol{x}) + \overline{L}_k D_h(\boldsymbol{x}, \boldsymbol{y}_k) \qquad \texttt{[update]}$$

satisfies

$$f(\boldsymbol{x}_{k+1}) \leq f_{\boldsymbol{y}_k}(\boldsymbol{x}_{k+1}) + \overline{L}_k D_h(\boldsymbol{x}_{k+1}, \boldsymbol{y}_k)\,. \qquad \texttt{[upper bound]}$$

Here, parameters for convex model functions. Can be extended to weakly convex model function.

Convex–Concave Inertial Backtracking (CoCaIn)

Convex–Concave Inertial Backtracking: $\min_{\boldsymbol{x}} f(\boldsymbol{x})$

- Find $\underline{L}_k$ and γ_k (via backtracking) such that

$$\boldsymbol{y}_k = \boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \qquad \texttt{[extrapolation step]}$$

satisfies

$$\begin{cases} f(\boldsymbol{x}_k) \geq f_{\boldsymbol{y}_k}(\boldsymbol{x}_k) - \underline{L}_k D_h(\boldsymbol{x}_k, \boldsymbol{y}_k) & \texttt{[lower bound]} \\ D_h(\boldsymbol{x}_k, \boldsymbol{y}_k) \leq \dfrac{\overline{L}_{k-1}}{\overline{L}_{k-1} + \underline{L}_k} D_h(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k)\,. & \texttt{[extrapolation bound]} \end{cases}$$

- Find $\overline{L}_k \geq \overline{L}_{k-1}$ (via backtracking) such that

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} f_{\boldsymbol{y}_k}(\boldsymbol{x}) + \overline{L}_k D_h(\boldsymbol{x}, \boldsymbol{y}_k) \qquad \texttt{[update]}$$

satisfies

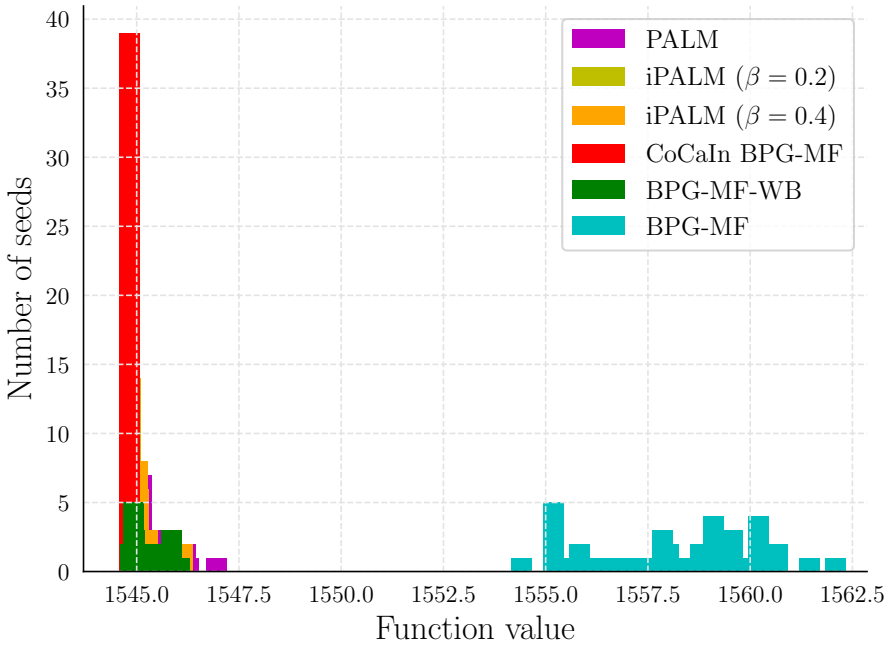
$$f(\boldsymbol{x}_{k+1}) \leq f_{\boldsymbol{y}_k}(\boldsymbol{x}_{k+1}) + \overline{L}_k D_h(\boldsymbol{x}_{k+1}, \boldsymbol{y}_k)\,. \qquad \texttt{[upper bound]}$$

Global convergence to a stationary point in KL-framework.

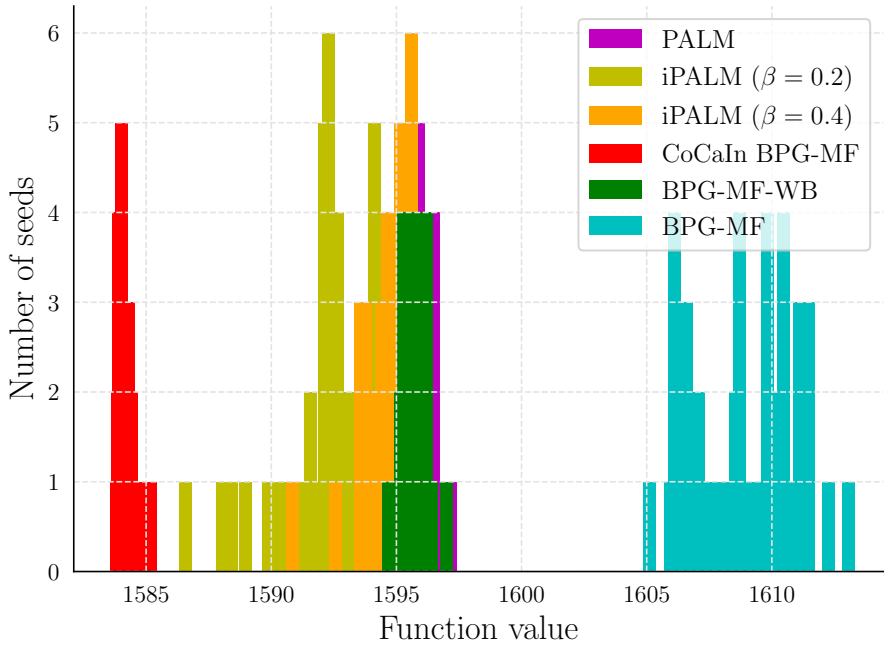
Matrix Factorization

$$\min_{\boldsymbol{X},\boldsymbol{Y}} \ \tfrac{1}{2}\|\boldsymbol{A}-\boldsymbol{X}\boldsymbol{Y}\|^2 + R_1(\boldsymbol{X}) + R_2(\boldsymbol{Y})$$

Statistical evaluation on simple matrix factorization ($N = 2$).



L2-regularization

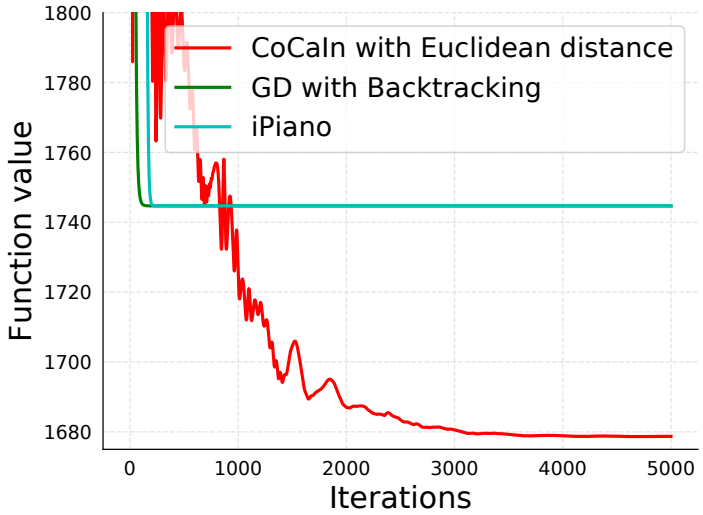


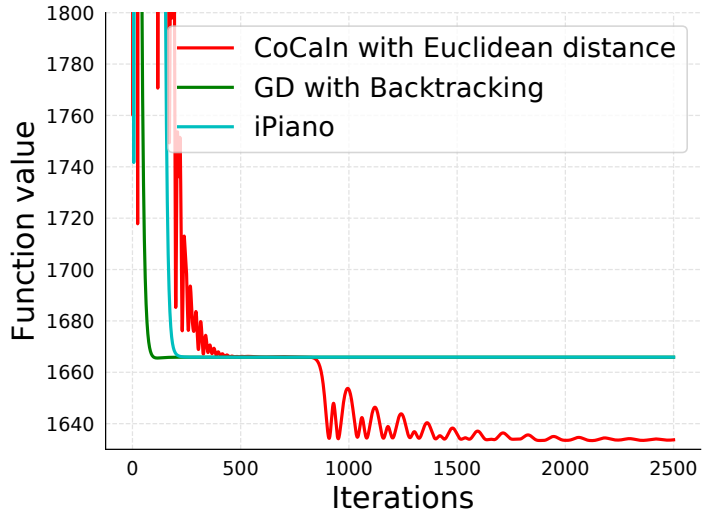
L1-regularization

Inertial strategy often results in a smaller objective value.

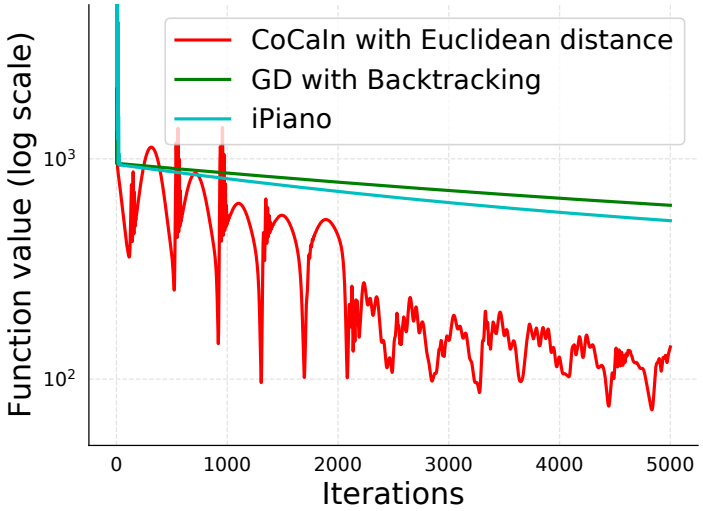
Matrix Factorization

toy experiment

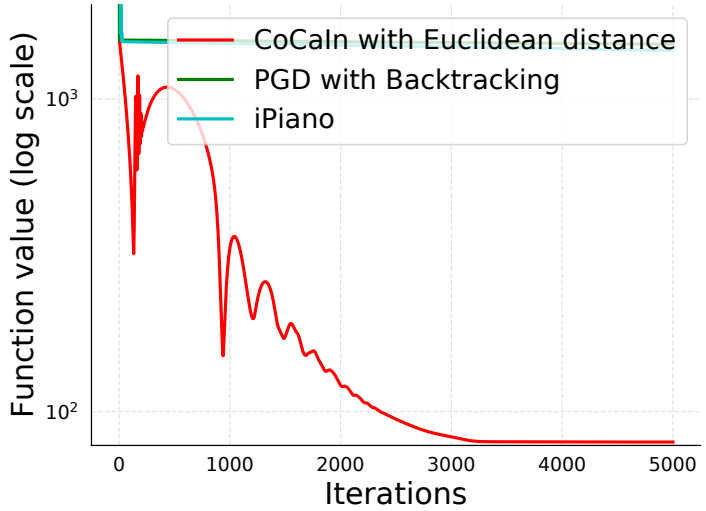




Medulloblastoma dataset
[Brunet, Tamayo,
Golub, Mesirov 2004]



L2 regularization



L1 regularization

Summary / Conclusion

Conclusion:

- Model BPG and CoCaIn Model BPG:

  unified framework for design and analysis of algorithms.

- Global Convergence to a stationary point
- Good performance in experiments.
- Requires expert knowledge of the problem and models.
- CoCaIn Backtracking seeks to locally adapt to the convexity of the objective.