Adaptive Acceleration for First-order Methods

Trajectory & Linear Prediction

Jingwei LIANG

Institute of Natural Sciences, Shanghai Jiao Tong University

Joint work with: **Clarice POON, University of Bath**

Table of contents

Introduction

Non-smooth optimization

Composite non-smooth optimization

$$\min_{x\in\mathbb{R}^n} \; F(x)+\sum\nolimits_{i=1}^{r} R_i(K_i x)$$

Non-smooth optimization

Composite non-smooth optimization

$$\min_{x\in\mathbb{R}^n} \; F(x)+\sum_{i=1}^{r} R_i(K_i x)$$

Basic assumptions

- F is smooth differentiable with L-Lipschitz continuous gradient.
- $R_i, i = 1,\ldots,r$ are proper, convex and lower semi-continuous.
- $K_i, i = 1,\ldots,r$ are linear operators.

Non-smooth optimization

Composite non-smooth optimization

$$\min_{x\in\mathbb{R}^n} \; F(x)+\sum\nolimits_{i=1}^{r} R_i(K_i x)$$

Basic assumptions

- F is smooth differentiable with L-Lipschitz continuous gradient.
- $R_i, i = 1,\ldots,r$ are proper, convex and lower semi-continuous.
- $K_i, i = 1,\ldots,r$ are linear operators.

Applications: signal/image processing, inverse problems, data science, machine learning…

Challenges: non-smooth, (non-convex), composite, high dimension…

First-order methods: two basic ingredients

Gradient descent [Cauchy '1847]

$$\min_x \; F(x)$$

where F is convex, smooth differentiable with ∇F being L-Lipschitz.

Forward Euler scheme: $\gamma_k \in \,]0,\, 2/L]$

$$x_k = x_{k-1} - \gamma_k \nabla F(x_{k-1})\,.$$

Gradient flow

$$\dot{x}(t) = -\nabla F(x(t))$$

First-order methods: two basic ingredients

Gradient descent [Cauchy '1847]

$$\min_x \; F(x)$$

where F is convex, smooth differentiable with ∇F being L-Lipschitz.

Forward Euler scheme: $\gamma_k \in \,]0,\, 2/L]$

$$x_k = x_{k-1} - \gamma_k \nabla F(x_{k-1})\,.$$

Gradient flow

$$\dot{x}(t) = -\nabla F(x(t))$$

Proximal point [Rockafellar '76]

$$\min_x \; R(x)$$

where R is proper convex and lower semi-continuous.

Proximal point: $\gamma > 0$

$$\mathrm{prox}_{\gamma R}(z) = \arg\min_x \; \gamma R(x) + \frac{1}{2}\|x - z\|^2.$$

Backward Euler scheme: $\gamma_k > 0$

$$x_k = x_{k-1} - \gamma_k \tilde{\partial} R(x_k)\,.$$

First-order methods: a rich class

Origins from numerical PDE back to 1950s, now ubiquitous in signal/image processing, inverse problems, data science, statistics, machine learning...

$F + R$	Forward-Backward splitting [Lions & Mercier '79;...]
$R_1 + R_2$	Douglas-Rachford splitting [Douglas & Rachford '56; Lions & Mercier '79;...] ADMM [Glowinski & Marrocco '75; Gabay & Mercier '76;...]
$F + R \circ K$	Primal-Dual splitting [Arrow, Hurwicz & Uzawa '58; Esser, Zhang & Chan '10; Chambolle & Pock '11;...]
$F + \sum_i R_i$	Generalized Froward-Backward splitting [Raguet, Fadili & Peyré '13;...]
	Many others....

Fixed-point formulation

Fix-point formulation Let $\mathscr{F} : \mathscr{H} \to \mathscr{H}$ be non-expansive with $\mathrm{fix}(\mathscr{F}) := \{z \in \mathscr{H} \,|\, z = \mathscr{F}(z)\} \neq \emptyset$

$$z_{k+1} = \mathscr{F}(z_k)\,.$$

Fixed-point formulation

Fix-point formulation Let $\mathscr{F} : \mathscr{H} \to \mathscr{H}$ be non-expansive with $\mathrm{fix}(\mathscr{F}) := \{z \in \mathscr{H} \,|\, z = \mathscr{F}(z)\} \neq \emptyset$

$$z_{k+1} = \mathscr{F}(z_k)\,.$$

- Convergence rate

$$\|z_k - z_{k-1}\| = o(1/\sqrt{k})\,.$$

Fixed-point formulation

Fix-point formulation Let $\mathscr{F}: \mathscr{H} \to \mathscr{H}$ be non-expansive with $\operatorname{fix}(\mathscr{F}) := \{z \in \mathscr{H} \,|\, z = \mathscr{F}(z)\} \neq \varnothing$

$$z_{k+1} = \mathscr{F}(z_k)\,.$$

- Convergence rate

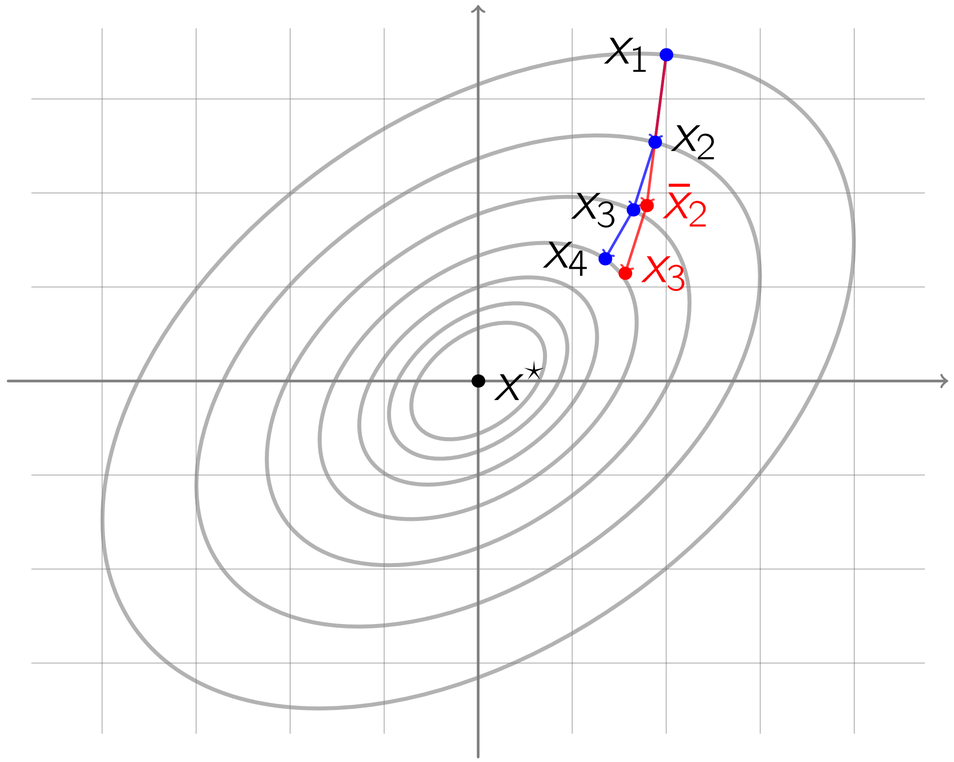
$$\|z_k - z_{k-1}\| = o(1/\sqrt{k})\,.$$

- Objective function values
  - Forward-Backward splitting: $\Phi(x_k) - \min_x \Phi(x) = o(1/k)$.
  - Other methods: **NA** in general.

Acceleration: inertial and relaxation

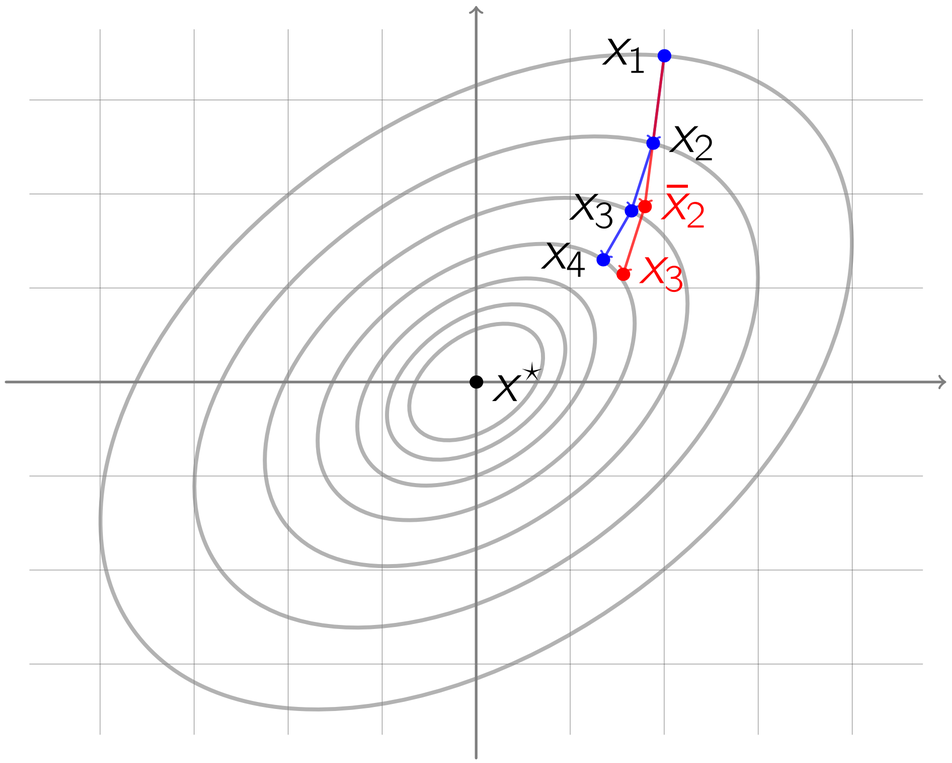
Inertial [Polyak '64; Nesterov '83; Beck & Teboulle '09...]

$$\bar{x}_k = x_k + a_k(x_k - x_{k-1}),$$
$$x_{k+1} = \mathscr{F}(\bar{x}_k)\,.$$



Acceleration: inertial and relaxation

Inertial [Polyak '64; Nesterov '83; Beck & Teboulle '09...]	**Relaxation** [Richardson '1911; Young '50; ...]
$\bar{x}_k = x_k + a_k(x_k - x_{k-1}),$ $x_{k+1} = \mathscr{F}(\bar{x}_k)\,.$	$x_{k+1} = (1-\lambda_k)x_k + \lambda_k \mathscr{F}(x_k)$



Acceleration: inertial and relaxation

Inertial [Polyak '64; Nesterov '83; Beck & Teboulle '09...]

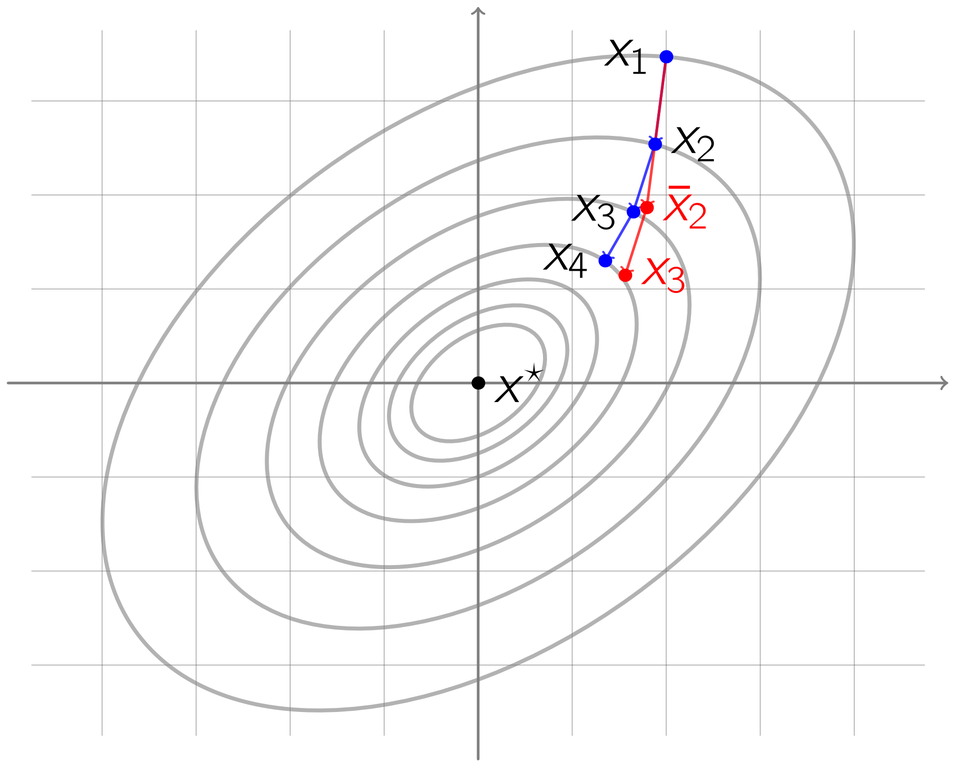
$$\bar{x}_k = x_k + a_k(x_k - x_{k-1}),$$
$$x_{k+1} = \mathscr{F}(\bar{x}_k)\,.$$

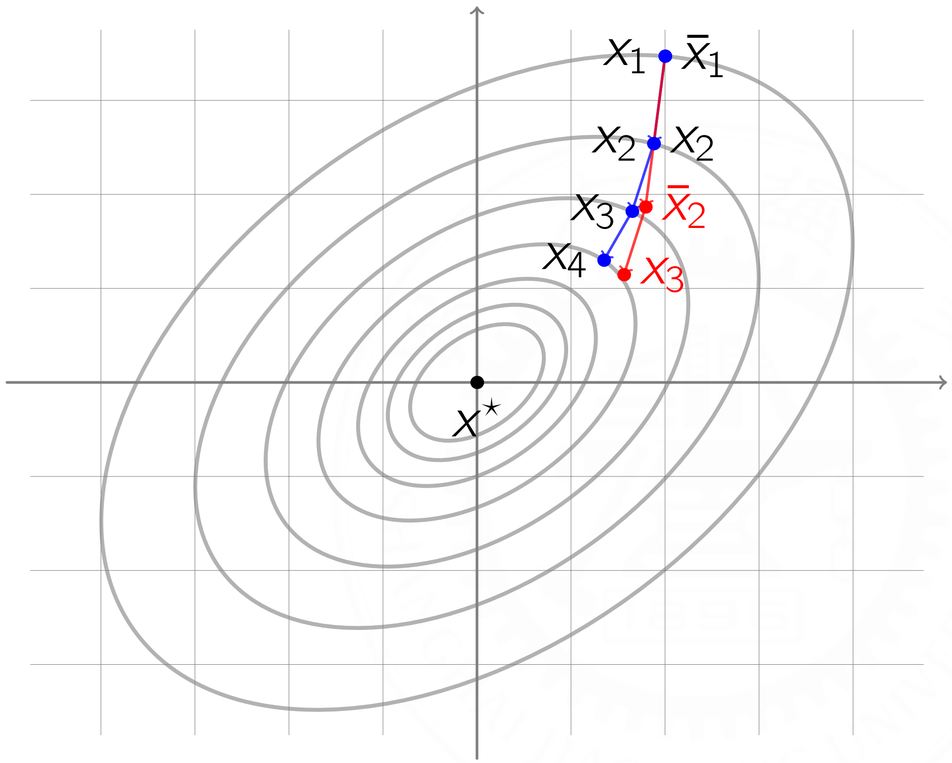
Relaxation [Richardson '1911; Young '50; ...]

$$x_{k+1} = (1-\lambda_k)x_k + \lambda_k \mathscr{F}(x_k)$$

$$a_k = \lambda_k - 1$$

$$\bar{x}_k = x_k + a_{k-1}(x_k - \bar{x}_{k-1}),$$
$$x_{k+1} = \mathscr{F}(\bar{x}_k)\,.$$





A comparison

$$\min_x \; F(x) + R(x)$$

A comparison

$$\min_x \; F(x) + R(x)$$

Let $\gamma > 0$

$$\mathscr{F}_{\mathrm{FB}} := \mathrm{prox}_{\gamma R}(\mathrm{Id} - \gamma \nabla F)\,.$$

A comparison

$$\min_x \; F(x) + R(x)$$

Let $\gamma > 0$

$$\mathscr{F}_{\mathrm{FB}} := \mathrm{prox}_{\gamma R}(\mathrm{Id} - \gamma \nabla F)\,.$$

Forward-Backward $\gamma \in \,]0,\, 2/L]$

$$x_{k+1} = \mathscr{F}_{\mathrm{FB}}(x_k)\,.$$

- Sequence $o(1/\sqrt{k})$, objective $o(1/k)$

A comparison

$$\min_x \; F(x) + R(x)$$

Let $\gamma > 0$

$$\mathscr{F}_{\mathrm{FB}} := \mathrm{prox}_{\gamma R}(\mathrm{Id} - \gamma \nabla F)\,.$$

Forward-Backward $\gamma \in \,]0,\, 2/L]$

$$x_{k+1} = \mathscr{F}_{\mathrm{FB}}(x_k)\,.$$

- Sequence $o(1/\sqrt{k})$, objective $o(1/k)$

FISTA $\gamma \in \,]0,\, 1/L]$, $d > 2$

$$\bar{x}_k = x_k + \frac{k-1}{k+d}(x_k - x_{k-1}),$$
$$x_{k+1} = \mathscr{F}_{\mathrm{FB}}(\bar{x}_k)\,.$$

- Sequence $o(1/k)$, objective $o(1/k^2)$

A comparison

$$\min_x \; F(x) + R(x)$$

Let $\gamma > 0$

$$\mathscr{F}_{\mathrm{FB}} := \mathrm{prox}_{\gamma R}(\mathrm{Id} - \gamma \nabla F)\,.$$

Forward-Backward $\gamma \in \,]0,\, 2/L]$

$$x_{k+1} = \mathscr{F}_{\mathrm{FB}}(x_k)\,.$$

- Sequence $o(1/\sqrt{k})$, objective $o(1/k)$

FISTA $\gamma \in \,]0,\, 1/L]$, $d > 2$

$$\bar{x}_k = x_k + \frac{k-1}{k+d}(x_k - x_{k-1}),$$
$$x_{k+1} = \mathscr{F}_{\mathrm{FB}}(\bar{x}_k)\,.$$

- Sequence $o(1/k)$, objective $o(1/k^2)$

Relaxation accelerates FB for

$$\lambda_k \in \left]1,\, \frac{4/L - \gamma}{\gamma}\right].$$

A comparison

$\min_x \; F(x) + R(x)$	$\min_x \; R(x) + J(x)$
Let $\gamma > 0$ $\mathscr{F}_{\text{FB}} := \text{prox}_{\gamma R}(\text{Id} - \gamma \nabla F)$.	
Forward-Backward $\gamma \in \,]0,\, 2/L]$ $x_{k+1} = \mathscr{F}_{\text{FB}}(x_k)$. ✦ Sequence $o(1/\sqrt{k})$, objective $o(1/k)$	
FISTA $\gamma \in \,]0,\, 1/L]$, $d > 2$ $\bar{x}_k = x_k + \frac{k-1}{k+d}(x_k - x_{k-1})$, $x_{k+1} = \mathscr{F}_{\text{FB}}(\bar{x}_k)$. ✦ Sequence $o(1/k)$, objective $o(1/k^2)$	
Relaxation accelerates FB for $\lambda_k \in \left]1,\, \frac{4/L - \gamma}{\gamma}\right]$.	

A comparison

$\min_x \; F(x) + R(x)$	$\min_x \; R(x) + J(x)$
Let $\gamma > 0$ $\mathscr{F}_{\mathrm{FB}} := \mathrm{prox}_{\gamma R}(\mathrm{Id} - \gamma \nabla F)$.	Let $\gamma > 0$ $\mathscr{F}_{\mathrm{DR}} := \frac{1}{2}\big((2\mathrm{prox}_{\gamma R} - \mathrm{Id})(2\mathrm{prox}_{\gamma J} - \mathrm{Id}) + \mathrm{Id}\big)$.
Forward-Backward $\gamma \in \,]0,\, 2/L]$ $x_{k+1} = \mathscr{F}_{\mathrm{FB}}(x_k)$. ✦ Sequence $o(1/\sqrt{k})$, objective $o(1/k)$	
FISTA $\gamma \in \,]0,\, 1/L]$, $d > 2$ $\bar{x}_k = x_k + \frac{k-1}{k+d}(x_k - x_{k-1})$, $x_{k+1} = \mathscr{F}_{\mathrm{FB}}(\bar{x}_k)$. ✦ Sequence $o(1/k)$, objective $o(1/k^2)$	
Relaxation accelerates FB for $\lambda_k \in \,]1,\, \frac{4/L - \gamma}{\gamma}]$.	

A comparison

$\min_x \; F(x) + R(x)$	$\min_x \; R(x) + J(x)$
Let $\gamma > 0$ $\mathscr{F}_{\mathrm{FB}} := \mathrm{prox}_{\gamma R}(\mathrm{Id} - \gamma \nabla F)$.	Let $\gamma > 0$ $\mathscr{F}_{\mathrm{DR}} := \frac{1}{2}\big((2\mathrm{prox}_{\gamma R} - \mathrm{Id})(2\mathrm{prox}_{\gamma J} - \mathrm{Id}) + \mathrm{Id}\big)$.
Forward-Backward $\gamma \in \,]0,\, 2/L]$ $x_{k+1} = \mathscr{F}_{\mathrm{FB}}(x_k)$. ✦ Sequence $o(1/\sqrt{k})$, objective $o(1/k)$	Douglas-Rachford $z_{k+1} = \mathscr{F}_{\mathrm{DR}}(z_k)$. ✦ Sequence $o(1/\sqrt{k})$, objective **NA**
FISTA $\gamma \in \,]0,\, 1/L]$, $d > 2$ $\bar{x}_k = x_k + \frac{k-1}{k+d}(x_k - x_{k-1})$, $x_{k+1} = \mathscr{F}_{\mathrm{FB}}(\bar{x}_k)$. ✦ Sequence $o(1/k)$, objective $o(1/k^2)$	
Relaxation accelerates FB for $\lambda_k \in \left]1,\, \frac{4/L - \gamma}{\gamma}\right]$.	

A comparison

$\min_x \; F(x) + R(x)$	$\min_x \; R(x) + J(x)$
Let $\gamma > 0$ $\mathscr{F}_{\mathrm{FB}} := \mathrm{prox}_{\gamma R}(\mathrm{Id} - \gamma \nabla F)$.	Let $\gamma > 0$ $\mathscr{F}_{\mathrm{DR}} := \frac{1}{2}\big((2\mathrm{prox}_{\gamma R} - \mathrm{Id})(2\mathrm{prox}_{\gamma J} - \mathrm{Id}) + \mathrm{Id}\big)$.
Forward-Backward $\gamma \in \,]0,\, 2/L]$ $x_{k+1} = \mathscr{F}_{\mathrm{FB}}(x_k)$. ✦ Sequence $o(1/\sqrt{k})$, objective $o(1/k)$	Douglas-Rachford $z_{k+1} = \mathscr{F}_{\mathrm{DR}}(z_k)$. ✦ Sequence $o(1/\sqrt{k})$, objective **NA**
FISTA $\gamma \in \,]0,\, 1/L]$, $d > 2$ $\bar{x}_k = x_k + \frac{k-1}{k+d}(x_k - x_{k-1})$, $x_{k+1} = \mathscr{F}_{\mathrm{FB}}(\bar{x}_k)$. ✦ Sequence $o(1/k)$, objective $o(1/k^2)$	Inertial Douglas-Rachford $\bar{z}_k = z_k + a_k(z_k - z_{k-1})$, $z_{k+1} = \mathscr{F}_{\mathrm{DR}}(\bar{z}_k)$. ✦ **NA**, may fail to provide acceleration
Relaxation accelerates FB for $\lambda_k \in \left]1,\, \frac{4/L - \gamma}{\gamma}\right]$.	

A comparison

$\min_x \; F(x) + R(x)$	$\min_x \; R(x) + J(x)$
Let $\gamma > 0$ $\mathscr{F}_{\mathrm{FB}} := \mathrm{prox}_{\gamma R}(\mathrm{Id} - \gamma \nabla F)$.	Let $\gamma > 0$ $\mathscr{F}_{\mathrm{DR}} := \frac{1}{2}\big((2\mathrm{prox}_{\gamma R} - \mathrm{Id})(2\mathrm{prox}_{\gamma J} - \mathrm{Id}) + \mathrm{Id}\big)$.
Forward-Backward $\gamma \in \,]0,\, 2/L]$ $x_{k+1} = \mathscr{F}_{\mathrm{FB}}(x_k)$. ✦ Sequence $o(1/\sqrt{k})$, objective $o(1/k)$	Douglas-Rachford $z_{k+1} = \mathscr{F}_{\mathrm{DR}}(z_k)$. ✦ Sequence $o(1/\sqrt{k})$, objective **NA**
FISTA $\gamma \in \,]0,\, 1/L]$, $d > 2$ $\bar{x}_k = x_k + \frac{k-1}{k+d}(x_k - x_{k-1})$, $x_{k+1} = \mathscr{F}_{\mathrm{FB}}(\bar{x}_k)$. ✦ Sequence $o(1/k)$, objective $o(1/k^2)$	Inertial Douglas-Rachford $\bar{z}_k = z_k + a_k(z_k - z_{k-1})$, $z_{k+1} = \mathscr{F}_{\mathrm{DR}}(\bar{z}_k)$. ✦ **NA**, may fail to provide acceleration
Relaxation accelerates FB for $\lambda_k \in \,]1,\, \frac{4/L - \gamma}{\gamma}]$.	**Relaxation** problem and parameter dependent! $\lambda_k \in \,]1,\, 2]$.

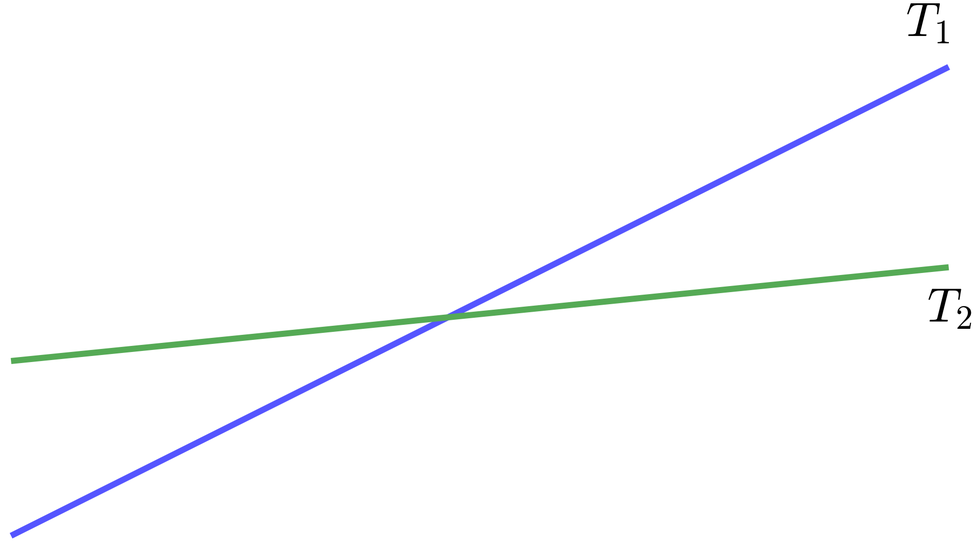
A comparison

$\min_x \; F(x) + R(x)$	$\min_x \; R(x) + J(x)$
Let $\gamma > 0$ $\mathscr{F}_{\mathrm{FB}} := \mathrm{prox}_{\gamma R}(\mathrm{Id} - \gamma \nabla F)$.	Let $\gamma > 0$ $\mathscr{F}_{\mathrm{DR}} := \frac{1}{2}\big((2\mathrm{prox}_{\gamma R} - \mathrm{Id})(2\mathrm{prox}_{\gamma J} - \mathrm{Id}) + \mathrm{Id}\big)$.
Forward-Backward $\gamma \in \,]0,\, 2/L]$ $x_{k+1} = \mathscr{F}_{\mathrm{FB}}(x_k)$. ✦ Sequence $o(1/\sqrt{k})$, objective $o(1/k)$	Douglas-Rachford $z_{k+1} = \mathscr{F}_{\mathrm{DR}}(z_k)$. ✦ Sequence $o(1/\sqrt{k})$, objective **NA**
FISTA $\gamma \in \,]0,\, 1/L]$, $d > 2$ $\bar{x}_k = x_k + \frac{k-1}{k+d}(x_k - x_{k-1})$, $x_{k+1} = \mathscr{F}_{\mathrm{FB}}(\bar{x}_k)$. ✦ Sequence $o(1/k)$, objective $o(1/k^2)$	Inertial Douglas-Rachford $\bar{z}_k = z_k + a_k(z_k - z_{k-1})$, $z_{k+1} = \mathscr{F}_{\mathrm{DR}}(\bar{z}_k)$. ✦ **NA**, may fail to provide acceleration
Relaxation accelerates FB for $\lambda_k \in \,]1,\, \frac{4/L - \gamma}{\gamma}]$.	**Relaxation** problem and parameter dependent! $\lambda_k \in \,]1,\, 2]$.

An example

Feasibility problem Let $T_1,\ T_2$ be two subspaces such that $T_1 \cap T_2 \neq \varnothing$

$$x \in \mathbb{R}^2 \quad \text{such that} \quad x \in T_1 \cap T_2 .$$



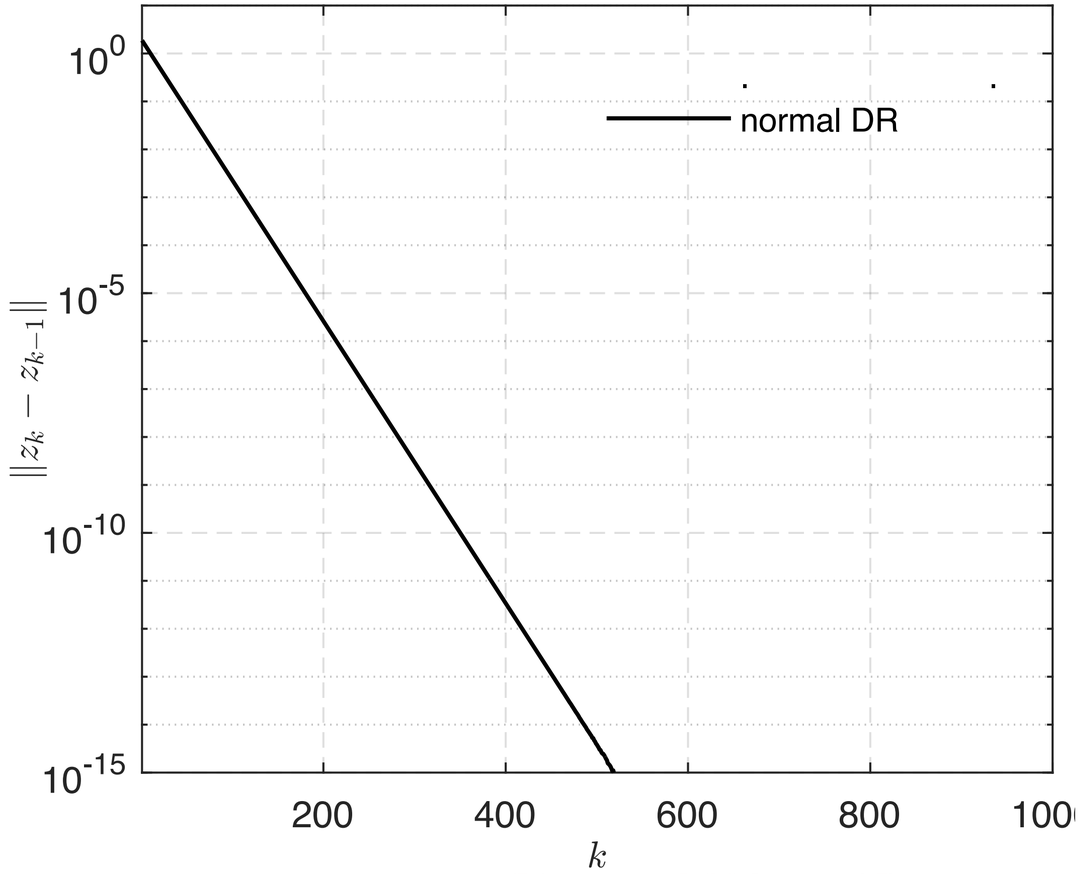
An example

Feasibility problem Let $T_1,\ T_2$ be two subspaces such that $T_1 \cap T_2 \neq \emptyset$

$$x \in \mathbb{R}^2 \quad \text{such that} \quad x \in T_1 \cap T_2 \,.$$

Douglas-Rachford splitting

$$z_{k+1} = \mathscr{F}_{\mathrm{DR}}(z_k) \,.$$



An example

Feasibility problem Let $T_1,\ T_2$ be two subspaces such that $T_1 \cap T_2 \neq \emptyset$

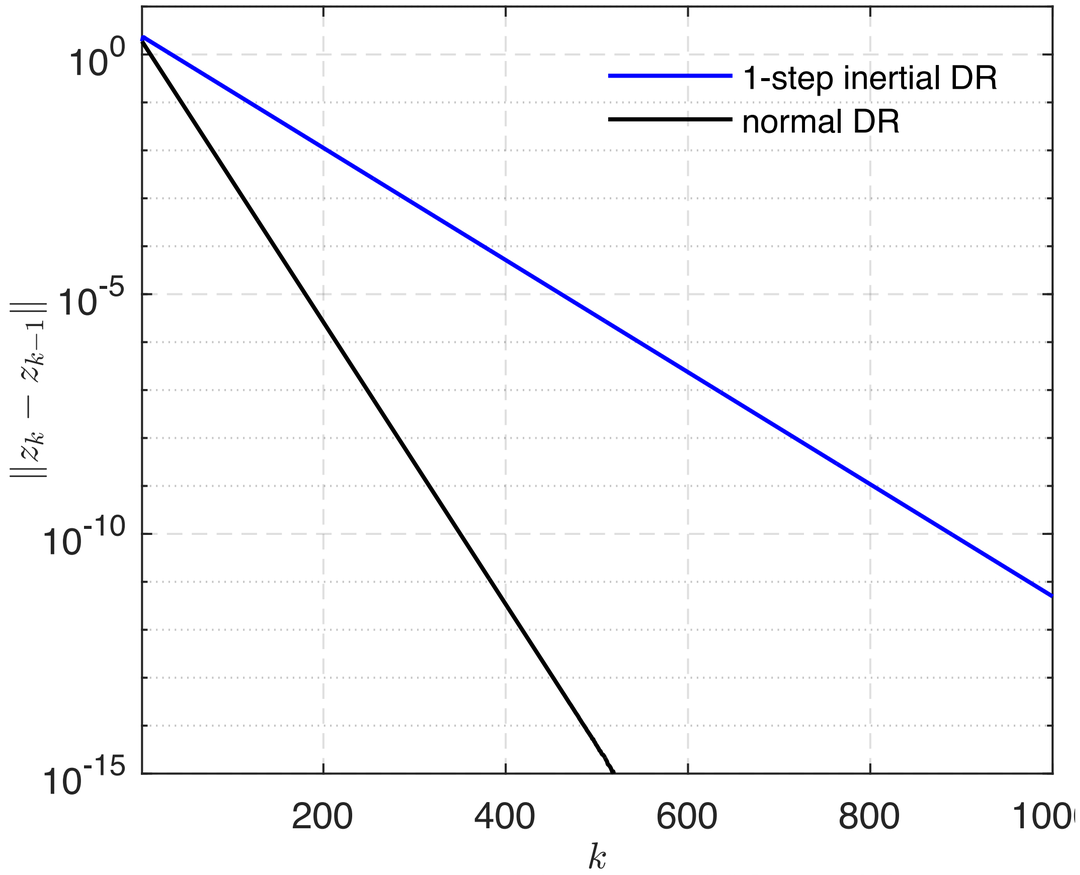
$$x \in \mathbb{R}^2 \quad \text{such that} \quad x \in T_1 \cap T_2 .$$

Douglas-Rachford splitting

$$z_{k+1} = \mathscr{F}_{\mathrm{DR}}(z_k) .$$

Inertial DR: $a = 0.3$

$$\bar{z}_k = z_k + a(z_k - z_{k-1}),$$
$$z_{k+1} = \mathscr{F}_{\mathrm{DR}}(\bar{z}_k) .$$



An example

Feasibility problem Let $T_1,\ T_2$ be two subspaces such that $T_1 \cap T_2 \neq \emptyset$

$$x \in \mathbb{R}^2 \quad \text{such that} \quad x \in T_1 \cap T_2 .$$

Douglas-Rachford splitting

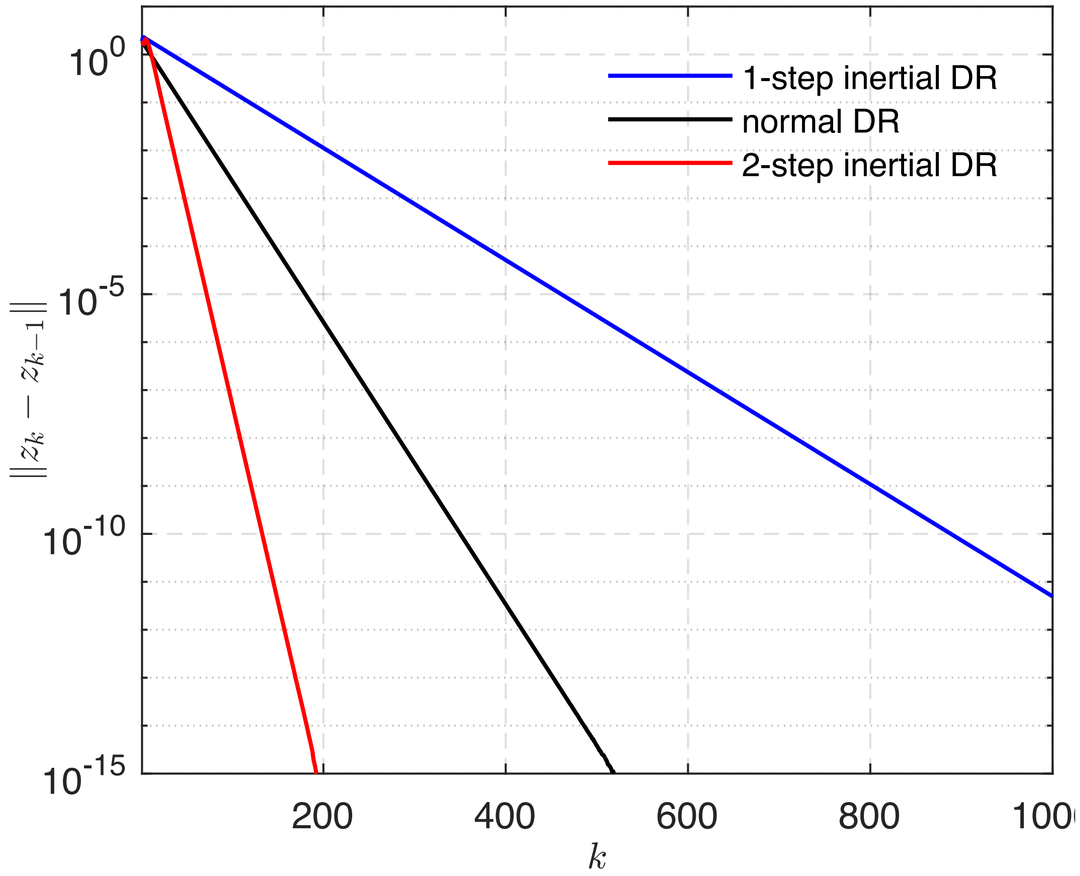
$$z_{k+1} = \mathscr{F}_{\mathrm{DR}}(z_k) .$$

Inertial DR: $a = 0.3$

$$\bar{z}_k = z_k + a(z_k - z_{k-1}),$$
$$z_{k+1} = \mathscr{F}_{\mathrm{DR}}(\bar{z}_k) .$$

Inertial DR: $a = 0.6,\ b = -0.3$

$$\bar{z}_k = z_k + a(z_k - z_{k-1}) + b(z_{k-1} - z_{k-2}),$$
$$z_{k+1} = \mathscr{F}_{\mathrm{DR}}(\bar{z}_k) .$$



Problems

- Nesterov/FISTA achieve worst case optimal convergence rate. *However, only for "smooth" or "smooth + non-smooth" type problems.*

Problems

- Nesterov/FISTA achieve worst case optimal convergence rate. *However, only for "smooth" or "smooth + non-smooth" type problems.*

- The performance of relaxation in general is not clear, *e.g.* no rate improvements.

Problems

- Nesterov/FISTA achieve worst case optimal convergence rate. *However, only for "smooth" or "smooth + non-smooth" type problems.*

- The performance of relaxation in general is not clear, *e.g.* no rate improvements.

- Extending inertial to first-order methods, or in general fixed-point iteration, is possible
  - Guaranteed convergence of sequence.
  - **NO** acceleration guarantees. Unless stronger assumptions are imposed.

Problems

- Nesterov/FISTA achieve worst case optimal convergence rate. *However, only for "smooth" or "smooth + non-smooth" type problems.*

- The performance of relaxation in general is not clear, *e.g.* no rate improvements.

- Extending inertial to first-order methods, or in general fixed-point iteration, is possible
  - Guaranteed convergence of sequence.
  - **NO** acceleration guarantees. Unless stronger assumptions are imposed.

- For a given problem, e.g. Douglas-Rachford/ADMM, the outcome of inertial/relaxation is problem and **parameters** dependent.

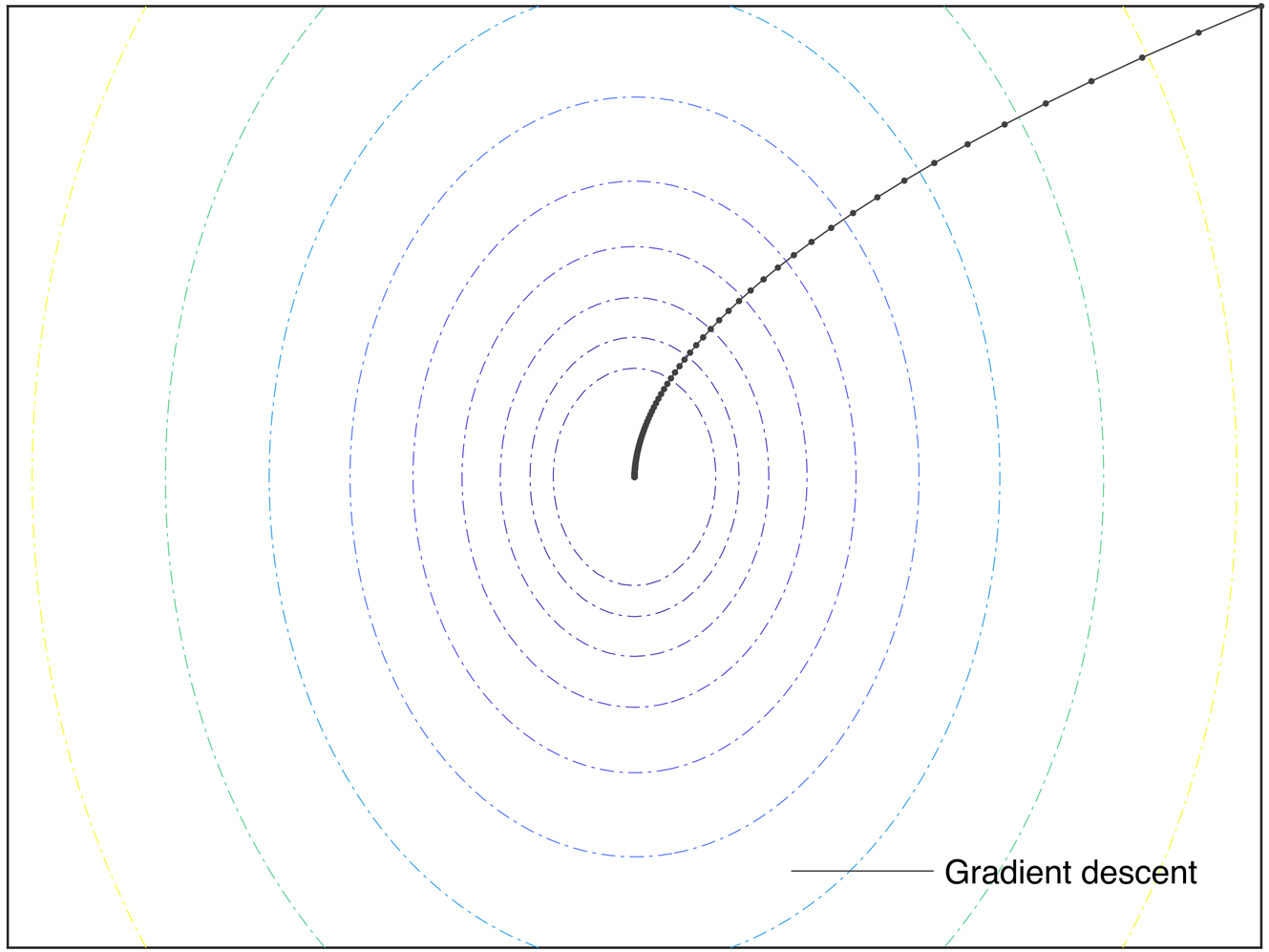
Trajectory of first-order methods

What makes the difference?

Trivial: they are simply different problems and different methods…

What makes the difference?

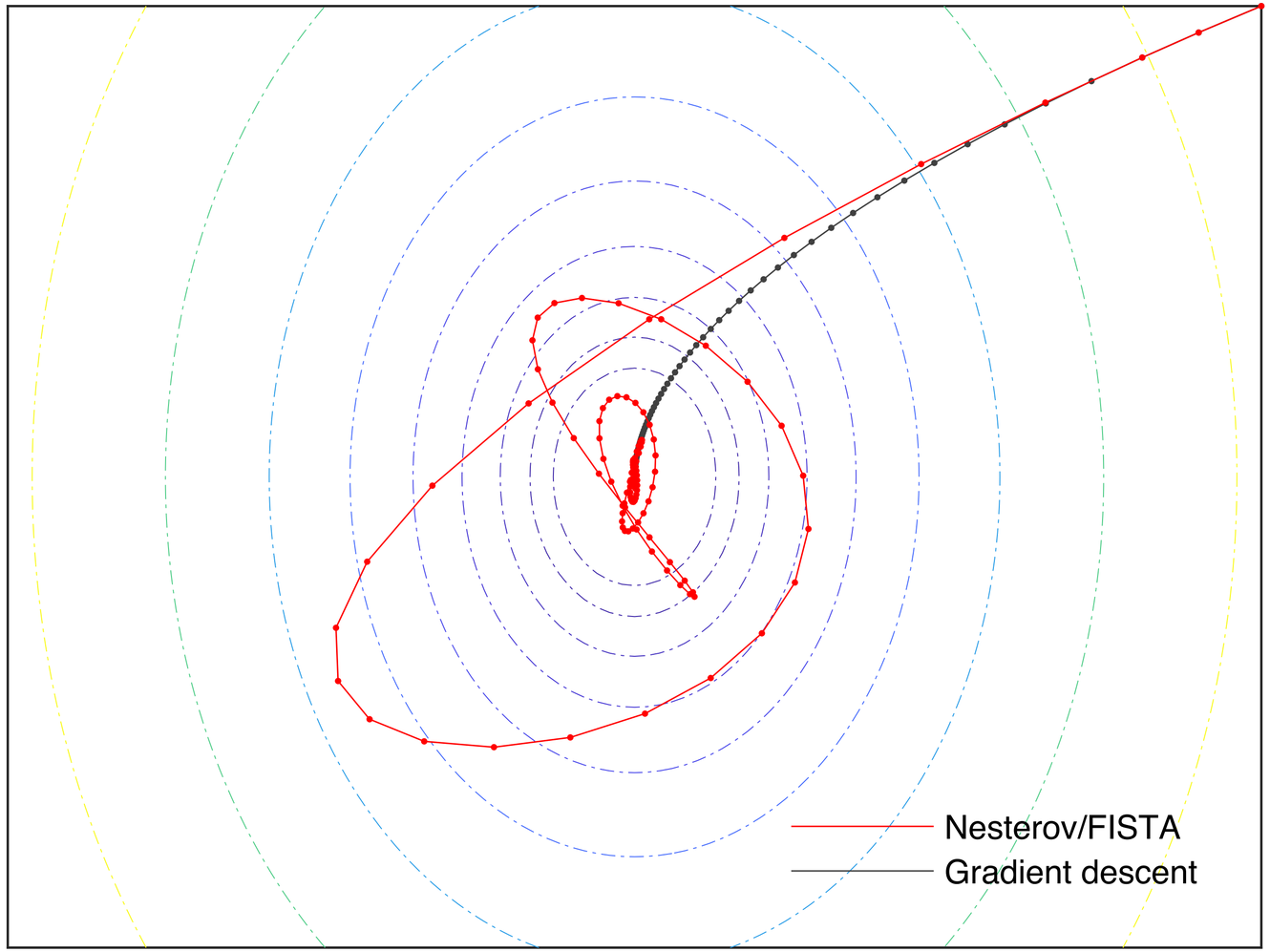
Trivial: they are simply different problems and different methods...



Trajectory of $\{x_k\}_{k\in\mathbb{N}}$

What makes the difference?

Trivial: they are simply different problems and different methods...

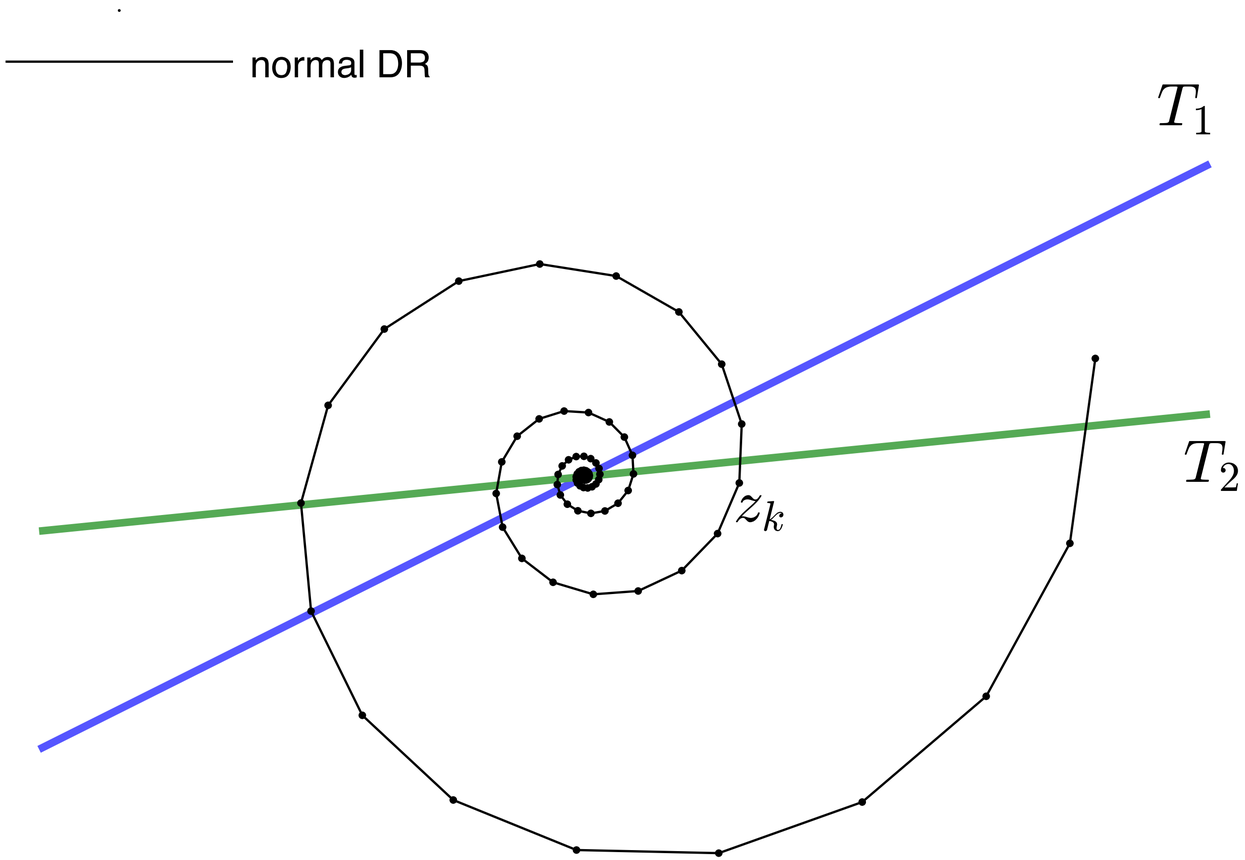


Trajectory of $\{x_k\}_{k\in\mathbb{N}}$

FISTA: Projection of elliptical spiral in $\mathbb{R}^4$ onto $\mathbb{R}^2$.

What makes the difference?

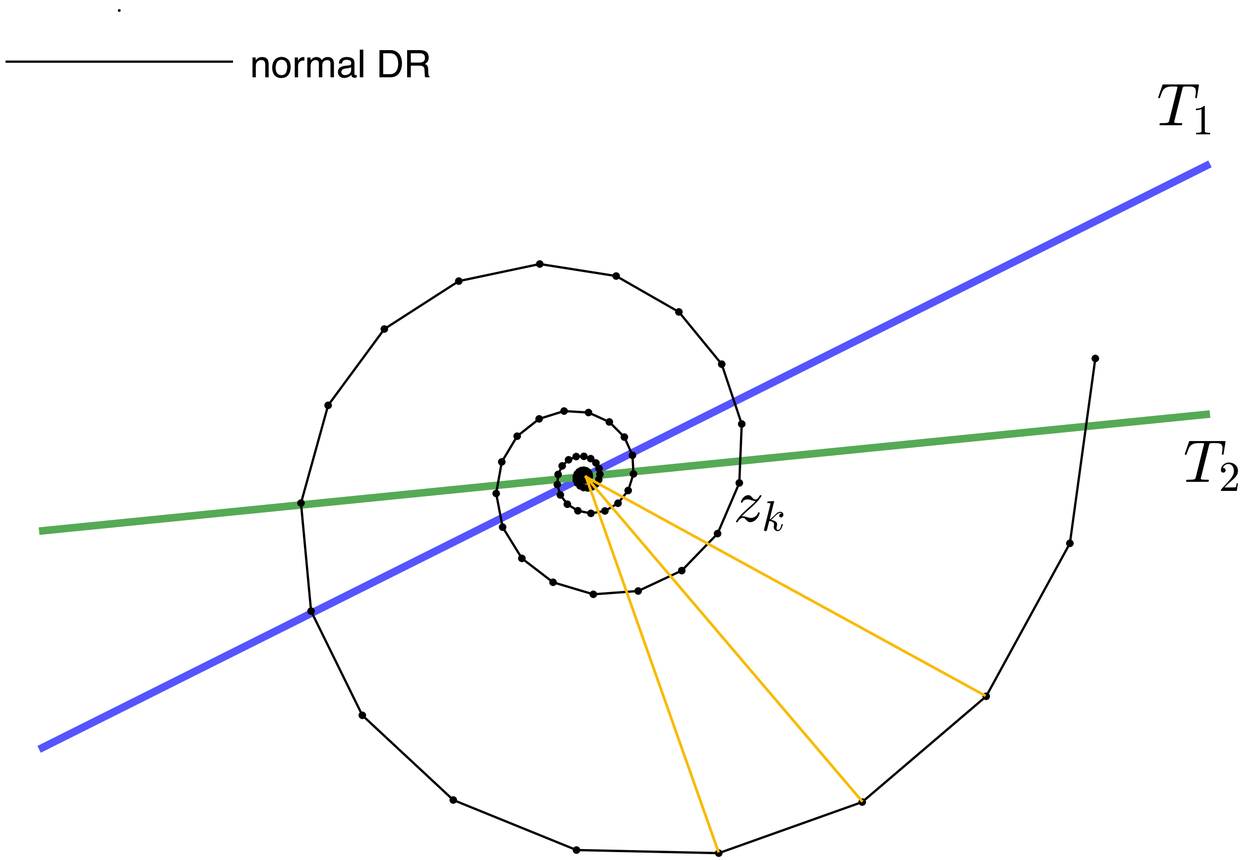
Trivial: they are simply different problems and different methods…



Trajectory of $\{z_k\}_{k\in\mathbb{N}}$

What makes the difference?

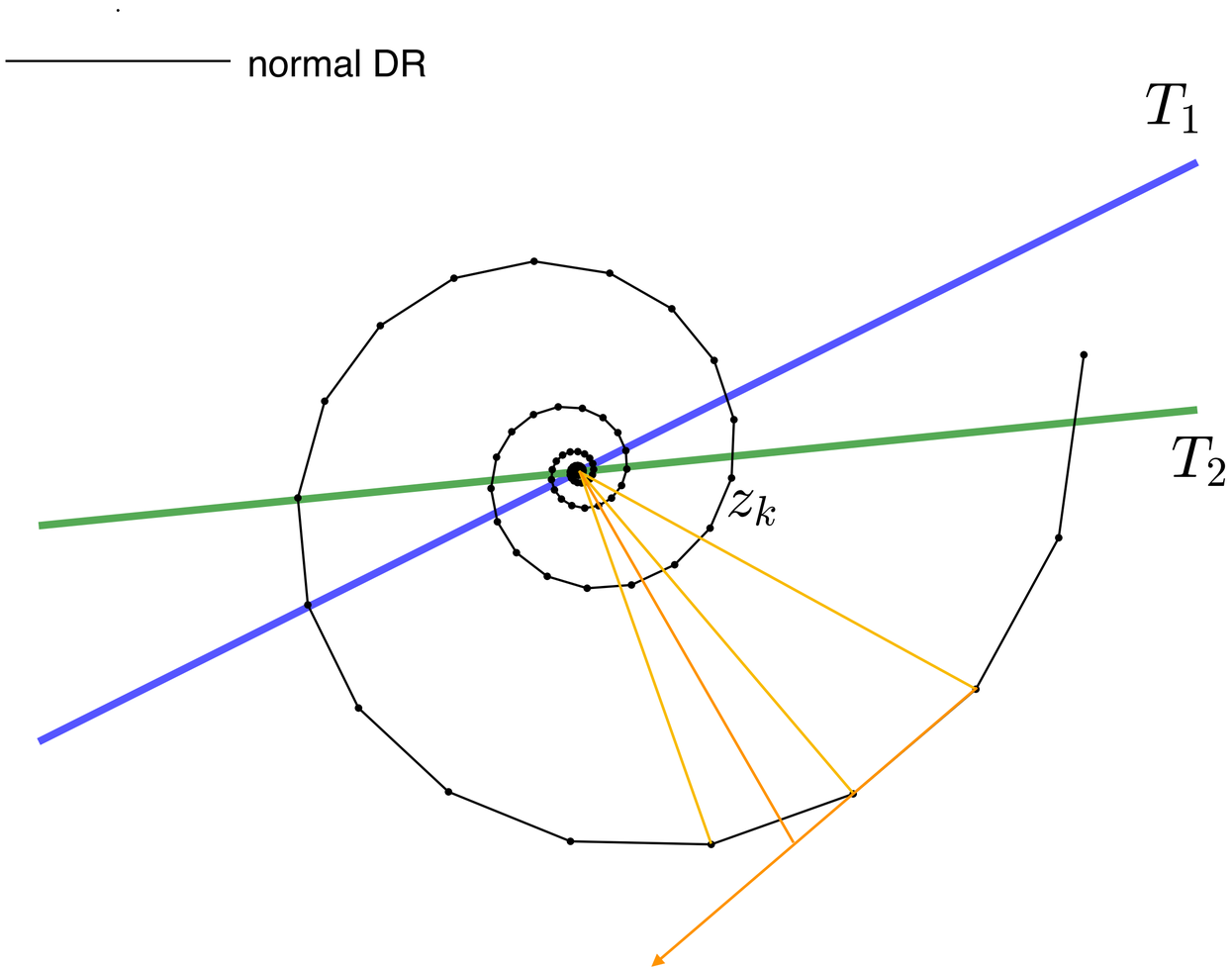
Trivial: they are simply different problems and different methods...



Trajectory of $\{z_k\}_{k\in\mathbb{N}}$

What makes the difference?

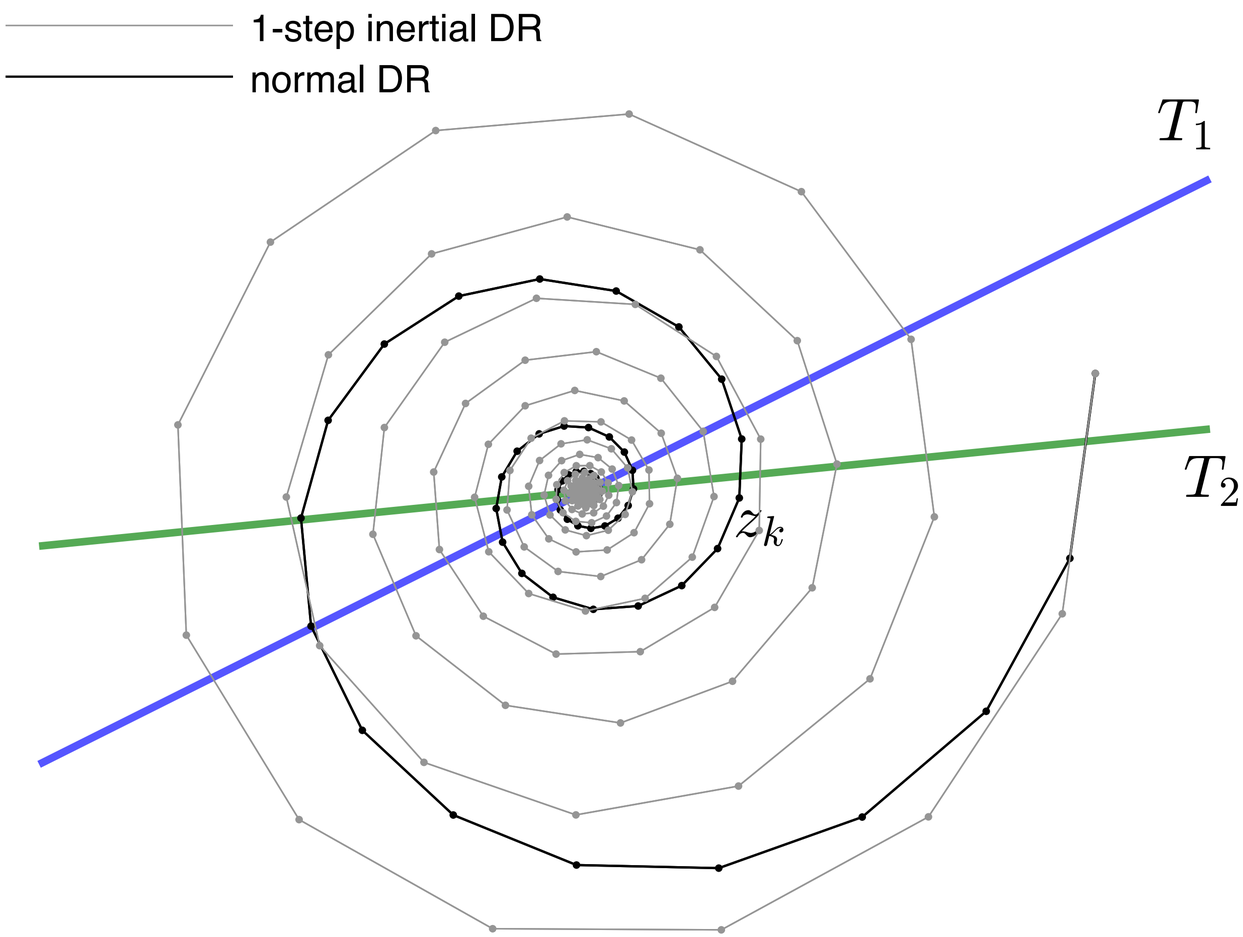
Trivial: they are simply different problems and different methods...



Trajectory of $\{z_k\}_{k\in\mathbb{N}}$

What makes the difference?

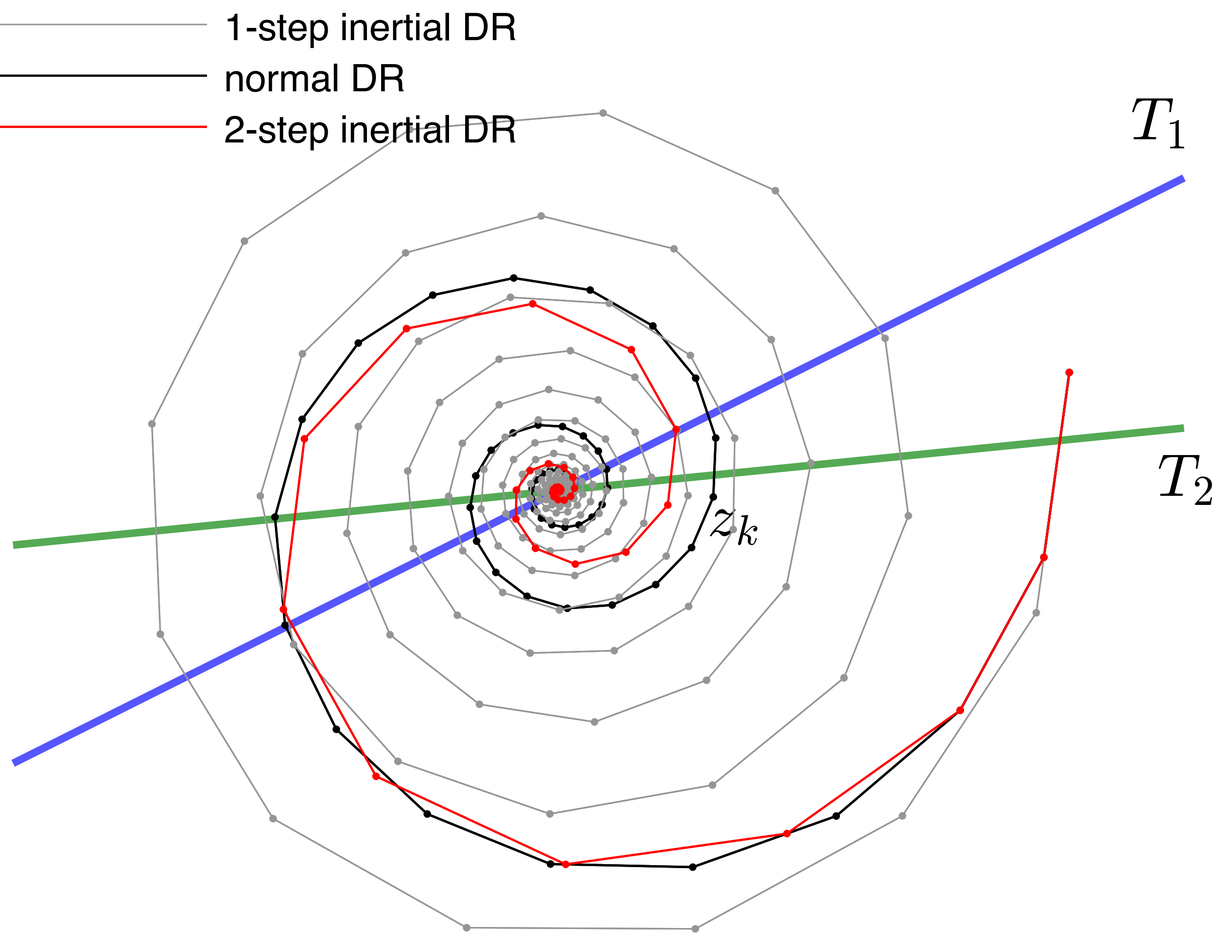
Trivial: they are simply different problems and different methods...



Trajectory of $\{z_k\}_{k\in\mathbb{N}}$

What makes the difference?

Trivial: they are simply different problems and different methods...



Trajectory of $\{z_k\}_{k\in\mathbb{N}}$

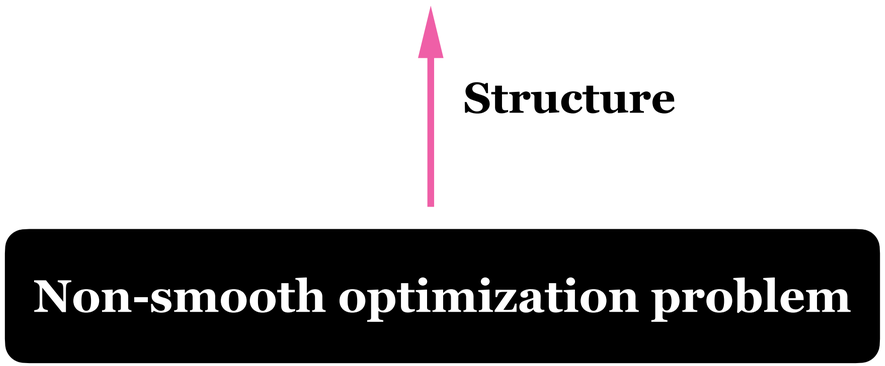
What makes the difference?

Trivial: they are simply different problems and different methods…

Non-smooth optimization problem

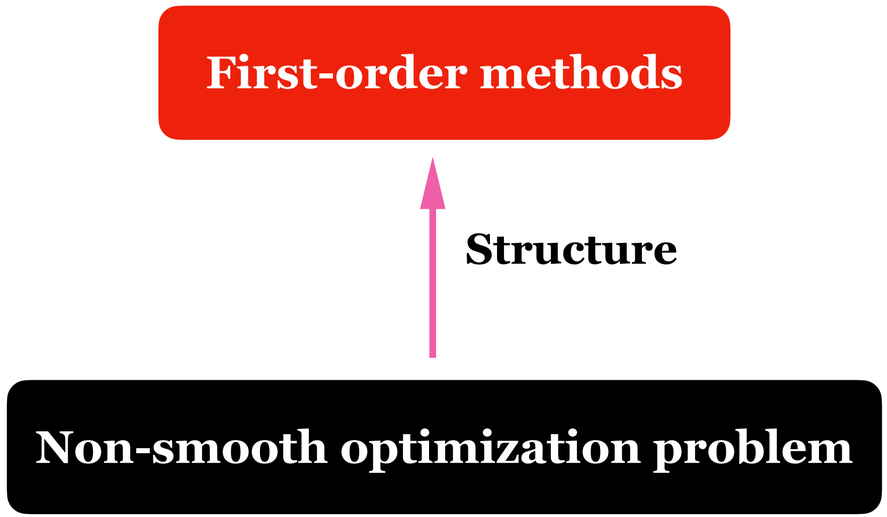
What makes the difference?

Trivial: they are simply different problems and different methods…



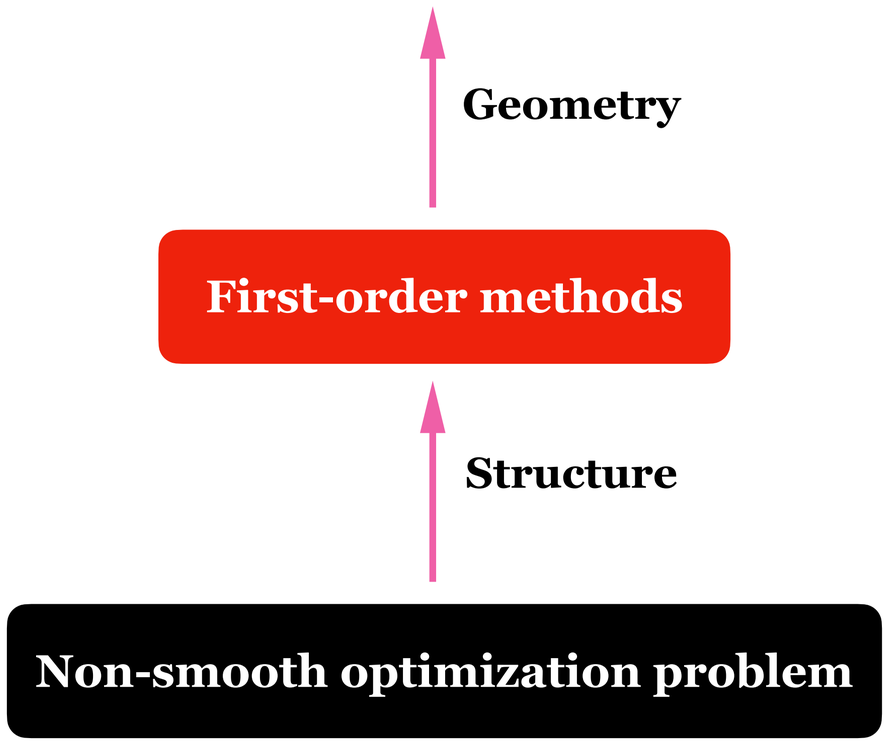
What makes the difference?

Trivial: they are simply different problems and different methods…



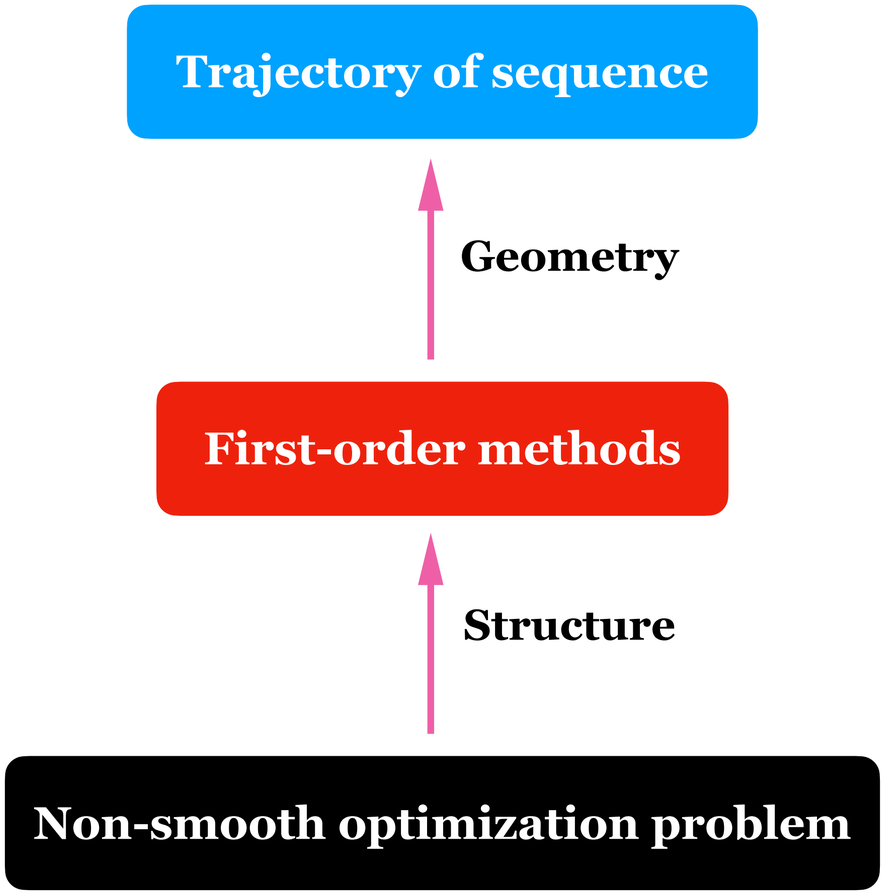
What makes the difference?

Trivial: they are simply different problems and different methods…



What makes the difference?

Trivial: they are simply different problems and different methods…



What makes the difference?

Trivial: they are simply different problems and different methods…

Trajectory of sequence

↑ **Geometry**

First-order methods

↑ **Structure**

Non-smooth optimization problem

First-order methods are **non-linear**….

$\mathscr{F}$

Partial smoothness [Lewis '03]

A function R is partly smooth at x relative to a set $\mathfrak{M}_x$ containing x if $\partial R(x) \neq \varnothing$ and

Smoothness $\mathfrak{M}_x$ is a C^2-smooth manifold, $R|_{\mathfrak{M}_x}$ is C^2 near x.

Sharpness Tangent space $T_{\mathfrak{M}_x}$ is $T_x := \mathrm{par}\big(\partial R(x)\big)^{\perp}$.

Continuity $\partial R : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is continuous along $\mathfrak{M}_x$ near x.

$\mathrm{par}(C)$:sub-space parallel to C, where $C \subset \mathbb{R}^n$ is a non-empty convex set.

Partial smoothness [Lewis '03]

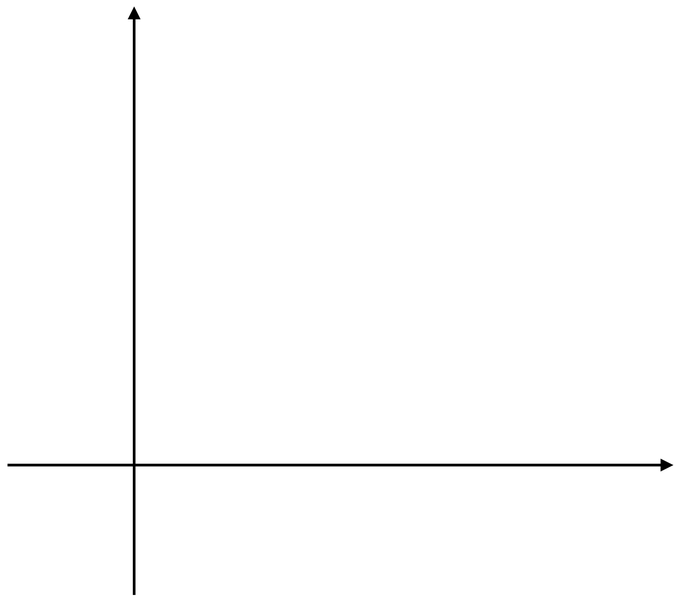
A function R is partly smooth at x relative to a set $\mathfrak{M}_x$ containing x if $\partial R(x) \neq \varnothing$ and

Smoothness $\mathfrak{M}_x$ is a C^2-smooth manifold, $R|_{\mathfrak{M}_x}$ is C^2 near x.

Sharpness Tangent space $T_{\mathfrak{M}_x}$ is $T_x := \mathrm{par}\big(\partial R(x)\big)^{\perp}$.

Continuity $\partial R : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is continuous along $\mathfrak{M}_x$ near x.

$\mathrm{par}(C)$:sub-space parallel to C, where $C \subset \mathbb{R}^n$ is a non-empty convex set.



Partial smoothness [Lewis '03]

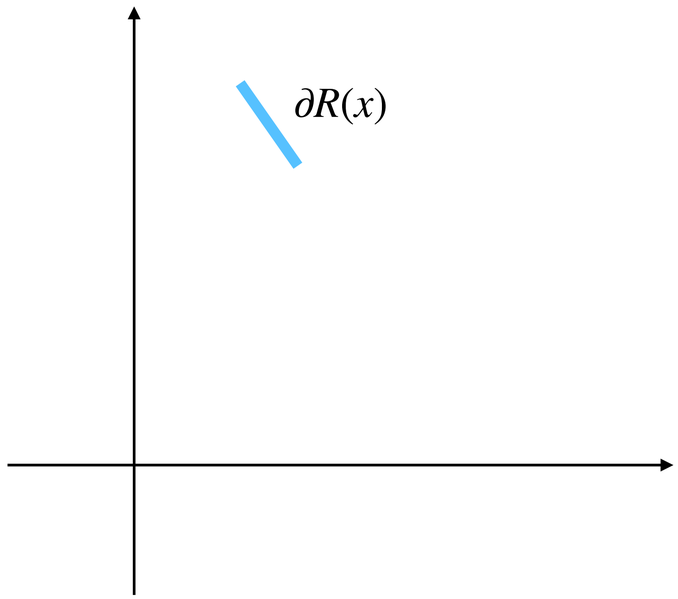
A function R is partly smooth at x relative to a set $\mathfrak{M}_x$ containing x if $\partial R(x) \neq \varnothing$ and

Smoothness $\mathfrak{M}_x$ is a C^2-smooth manifold, $R|_{\mathfrak{M}_x}$ is C^2 near x.

Sharpness Tangent space $T_{\mathfrak{M}_x}$ is $T_x := \mathrm{par}\big(\partial R(x)\big)^{\perp}$.

Continuity $\partial R : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is continuous along $\mathfrak{M}_x$ near x.

par(C):sub-space parallel to C, where $C \subset \mathbb{R}^n$ is a non-empty convex set.



Partial smoothness [Lewis '03]

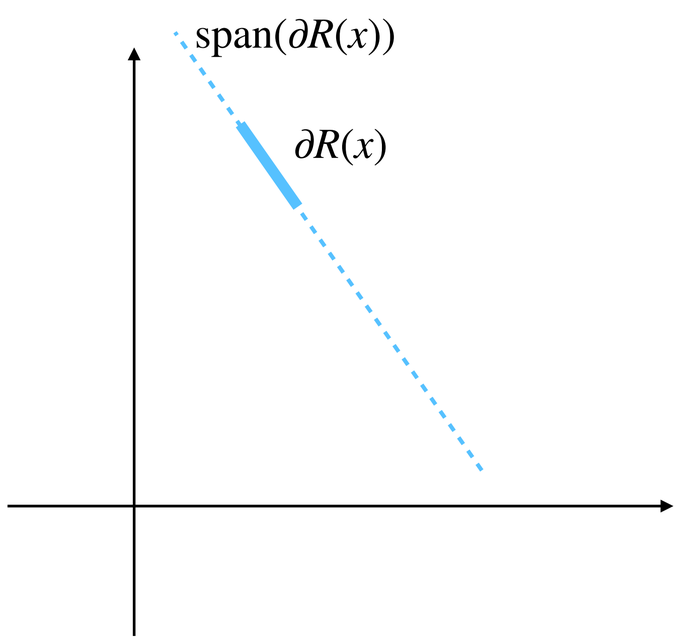
A function R is partly smooth at x relative to a set $\mathfrak{M}_x$ containing x if $\partial R(x) \neq \varnothing$ and

Smoothness $\mathfrak{M}_x$ is a C^2-smooth manifold, $R|_{\mathfrak{M}_x}$ is C^2 near x.

Sharpness Tangent space $T_{\mathfrak{M}_x}$ is $T_x := \mathrm{par}\big(\partial R(x)\big)^{\perp}$.

Continuity $\partial R : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is continuous along $\mathfrak{M}_x$ near x.

par(C):sub-space parallel to C, where $C \subset \mathbb{R}^n$ is a non-empty convex set.



Partial smoothness [Lewis '03]

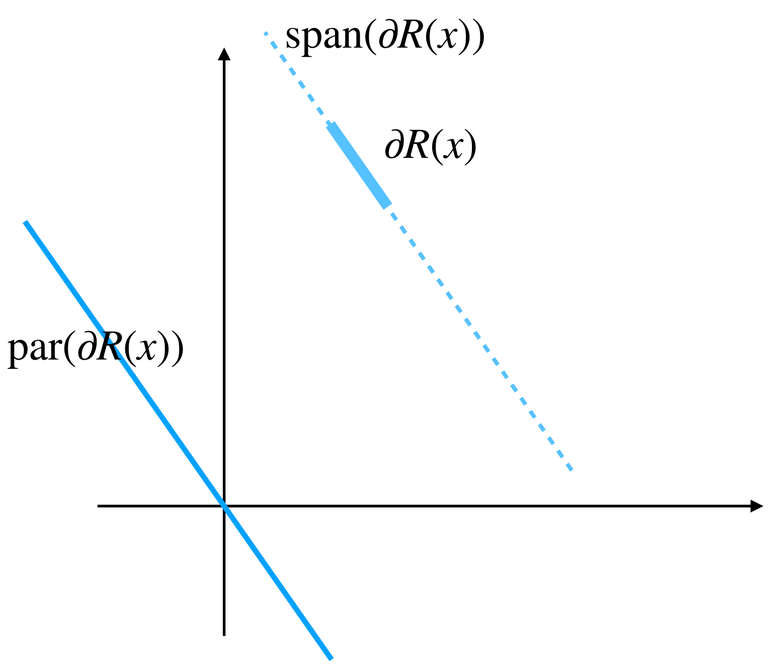
A function R is partly smooth at x relative to a set $\mathfrak{M}_x$ containing x if $\partial R(x) \neq \varnothing$ and

Smoothness $\mathfrak{M}_x$ is a C^2-smooth manifold, $R|_{\mathfrak{M}_x}$ is C^2 near x.

Sharpness Tangent space $T_{\mathfrak{M}_x}$ is $T_x := \mathrm{par}\big(\partial R(x)\big)^{\perp}$.

Continuity $\partial R : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is continuous along $\mathfrak{M}_x$ near x.

par(C):sub-space parallel to C, where $C \subset \mathbb{R}^n$ is a non-empty convex set.



Partial smoothness [Lewis '03]

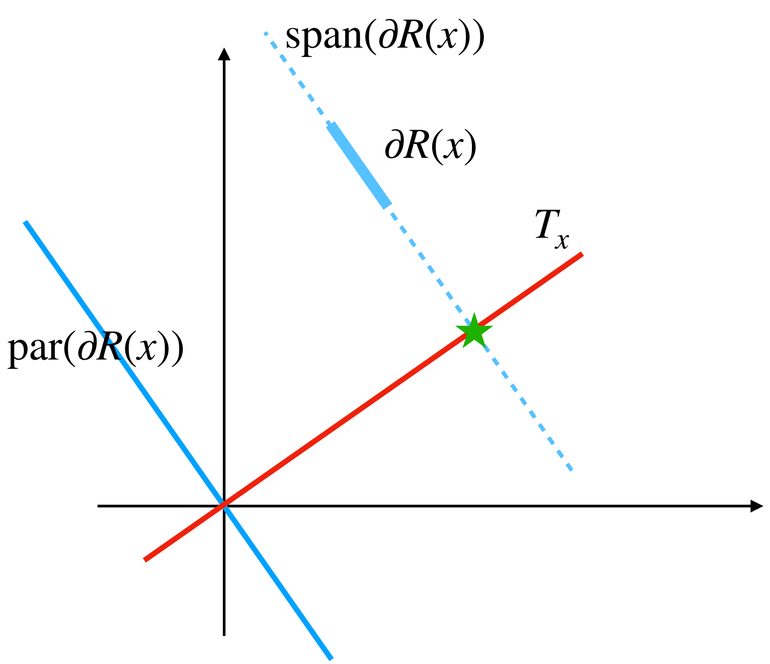
A function R is partly smooth at x relative to a set $\mathfrak{M}_x$ containing x if $\partial R(x) \neq \varnothing$ and

Smoothness $\mathfrak{M}_x$ is a C^2-smooth manifold, $R|_{\mathfrak{M}_x}$ is C^2 near x.

Sharpness Tangent space $T_{\mathfrak{M}_x}$ is $T_x := \mathrm{par}\big(\partial R(x)\big)^{\perp}$.

Continuity $\partial R : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is continuous along $\mathfrak{M}_x$ near x.

par(C):sub-space parallel to C, where $C \subset \mathbb{R}^n$ is a non-empty convex set.



Partial smoothness [Lewis '03]

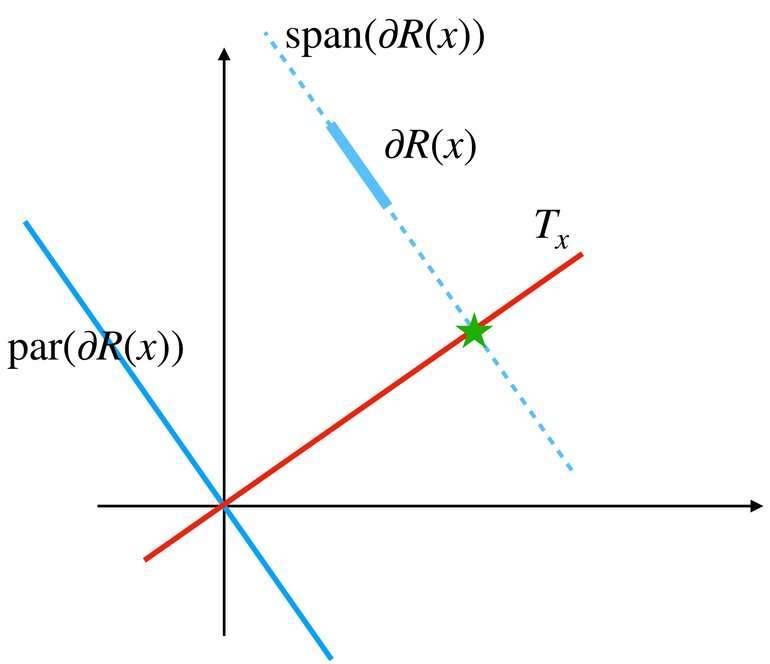
A function R is partly smooth at x relative to a set $\mathfrak{M}_x$ containing x if $\partial R(x) \neq \varnothing$ and

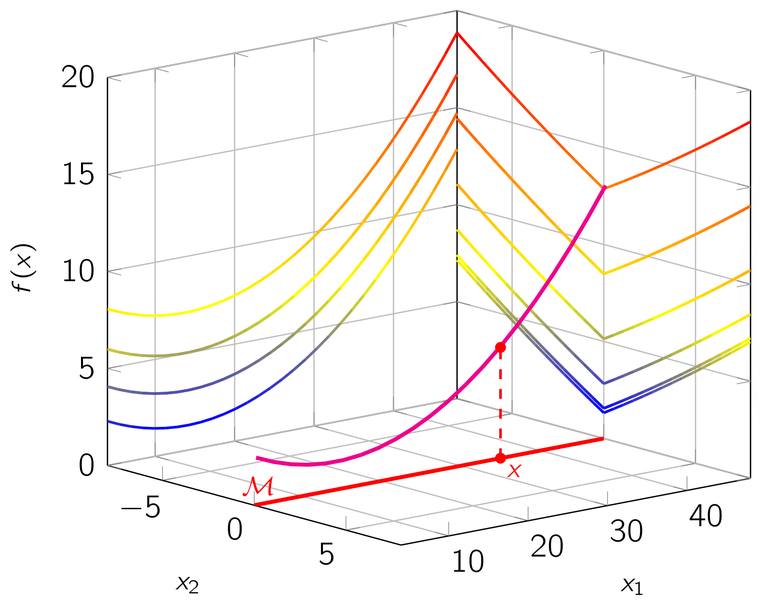
Smoothness $\mathfrak{M}_x$ is a C^2-smooth manifold, $R|_{\mathfrak{M}_x}$ is C^2 near x.

Sharpness Tangent space $T_{\mathfrak{M}_x}$ is $T_x := \mathrm{par}\big(\partial R(x)\big)^{\perp}$.

Continuity $\partial R : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is continuous along $\mathfrak{M}_x$ near x.

par(C):sub-space parallel to C, where $C \subset \mathbb{R}^n$ is a non-empty convex set.





Partial smoothness [Lewis '03]

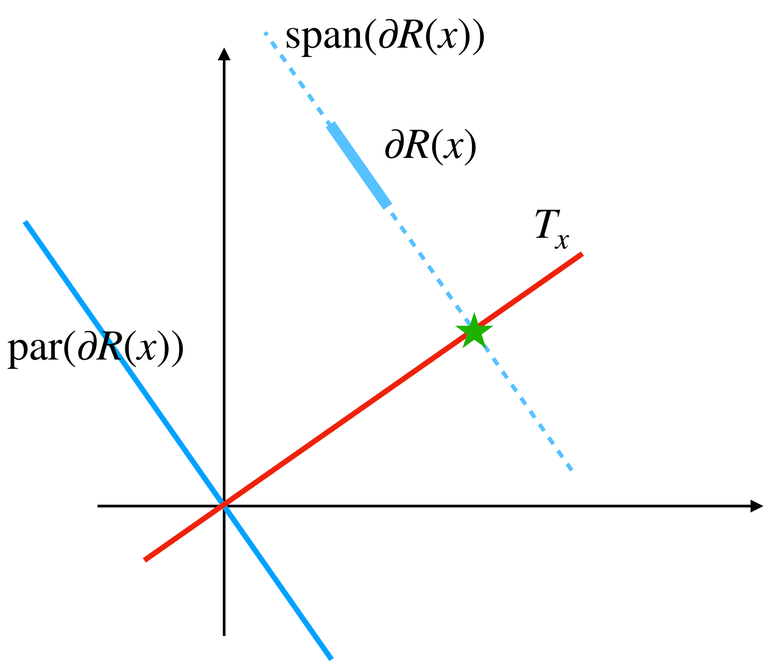
A function R is partly smooth at x relative to a set $\mathfrak{M}_x$ containing x if $\partial R(x) \neq \varnothing$ and

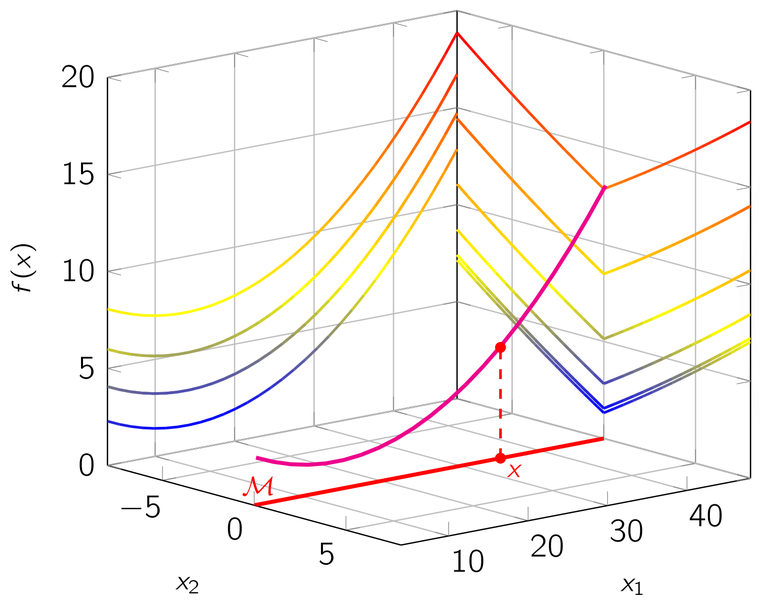
Smoothness $\mathfrak{M}_x$ is a C^2-smooth manifold, $R|_{\mathfrak{M}_x}$ is C^2 near x.

Sharpness Tangent space $T_{\mathfrak{M}_x}$ is $T_x := \mathrm{par}\big(\partial R(x)\big)^{\perp}$.

Continuity $\partial R : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is continuous along $\mathfrak{M}_x$ near x.

par(C):sub-space parallel to C, where $C \subset \mathbb{R}^n$ is a non-empty convex set.





Examples

- $\ell_1, \ell_{1,2}, \ell_\infty$-norms
- Nuclear norm
- Total variation
- Partly smooth sets...

Trajectory of first-order methods

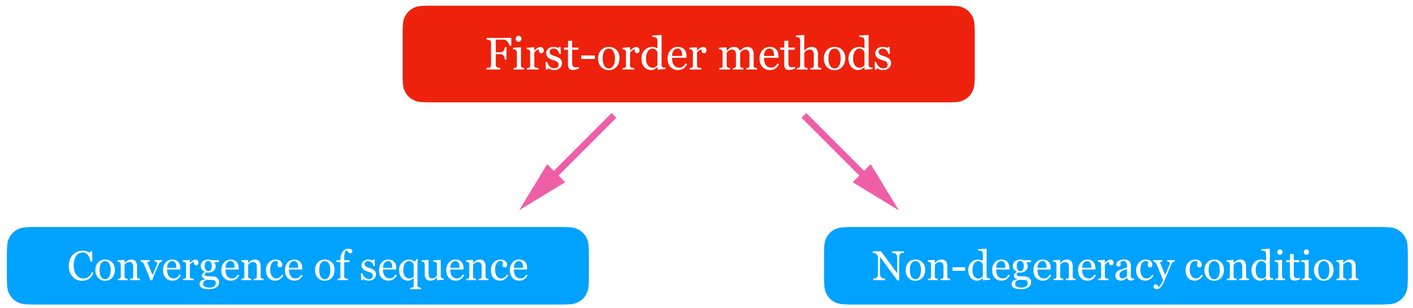
First-order methods

Trajectory of first-order methods

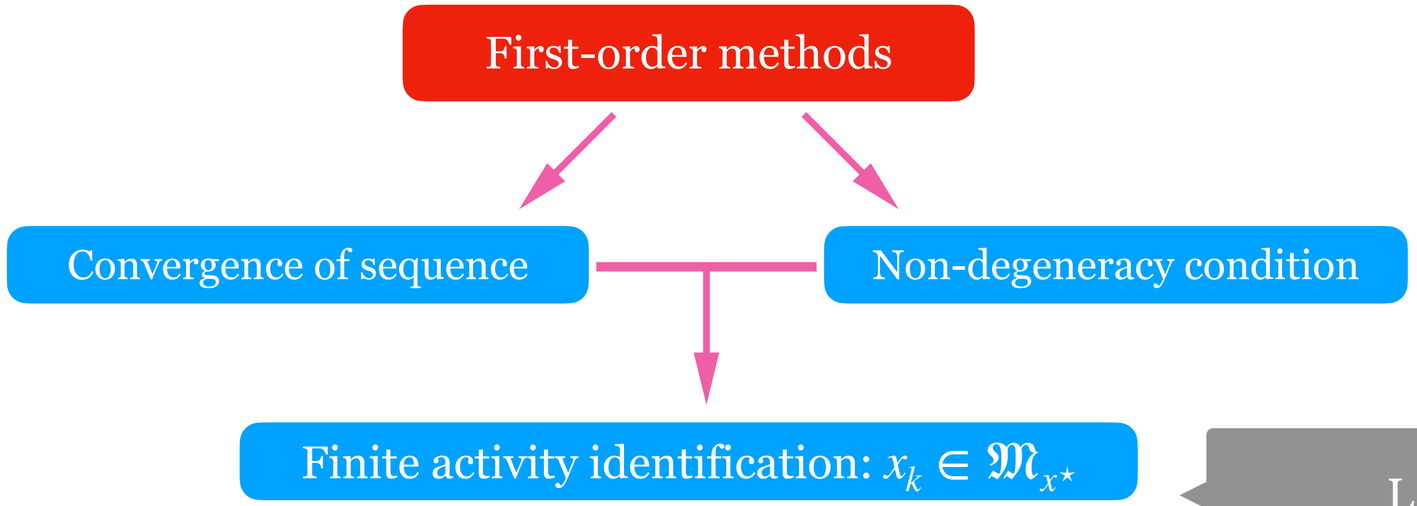
First-order methods

Convergence of sequence

Trajectory of first-order methods

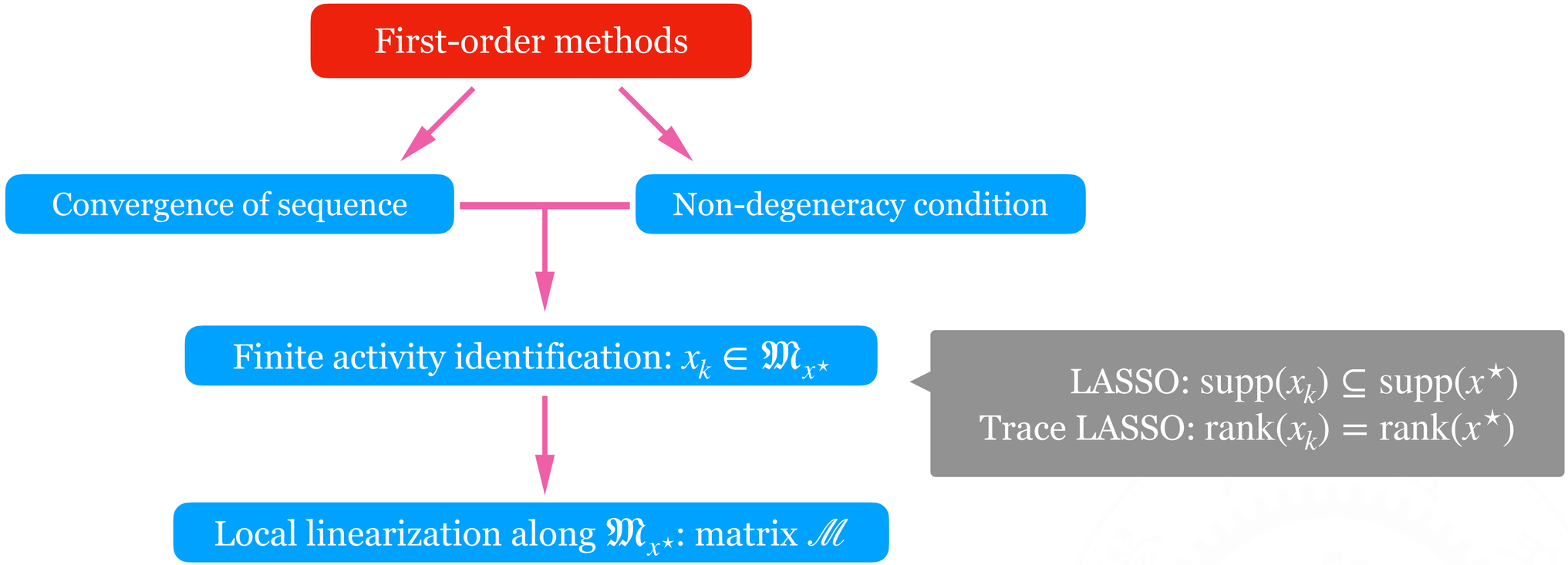


Trajectory of first-order methods

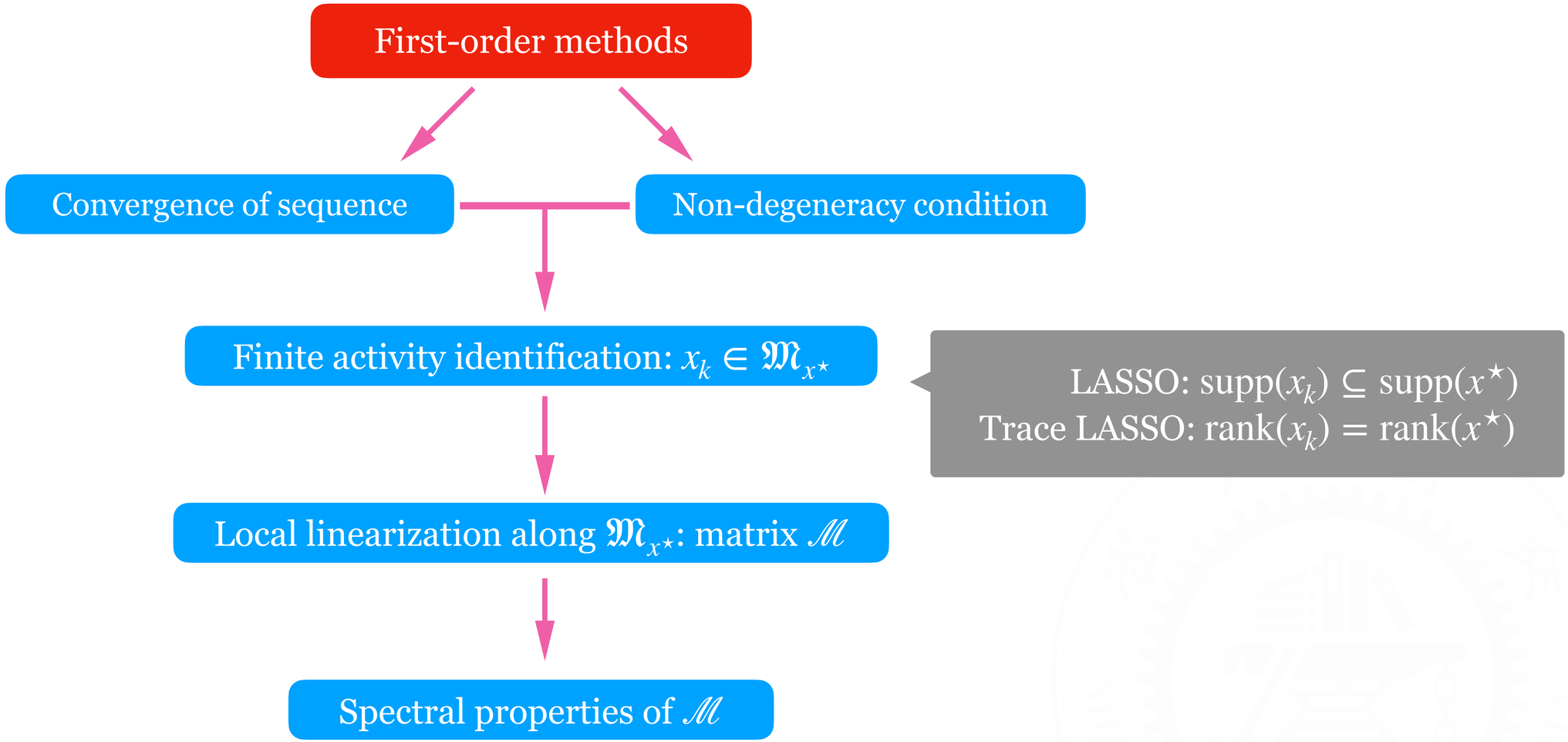


LASSO: $\mathrm{supp}(x_k) \subseteq \mathrm{supp}(x^\star)$
Trace LASSO: $\mathrm{rank}(x_k) = \mathrm{rank}(x^\star)$

Trajectory of first-order methods

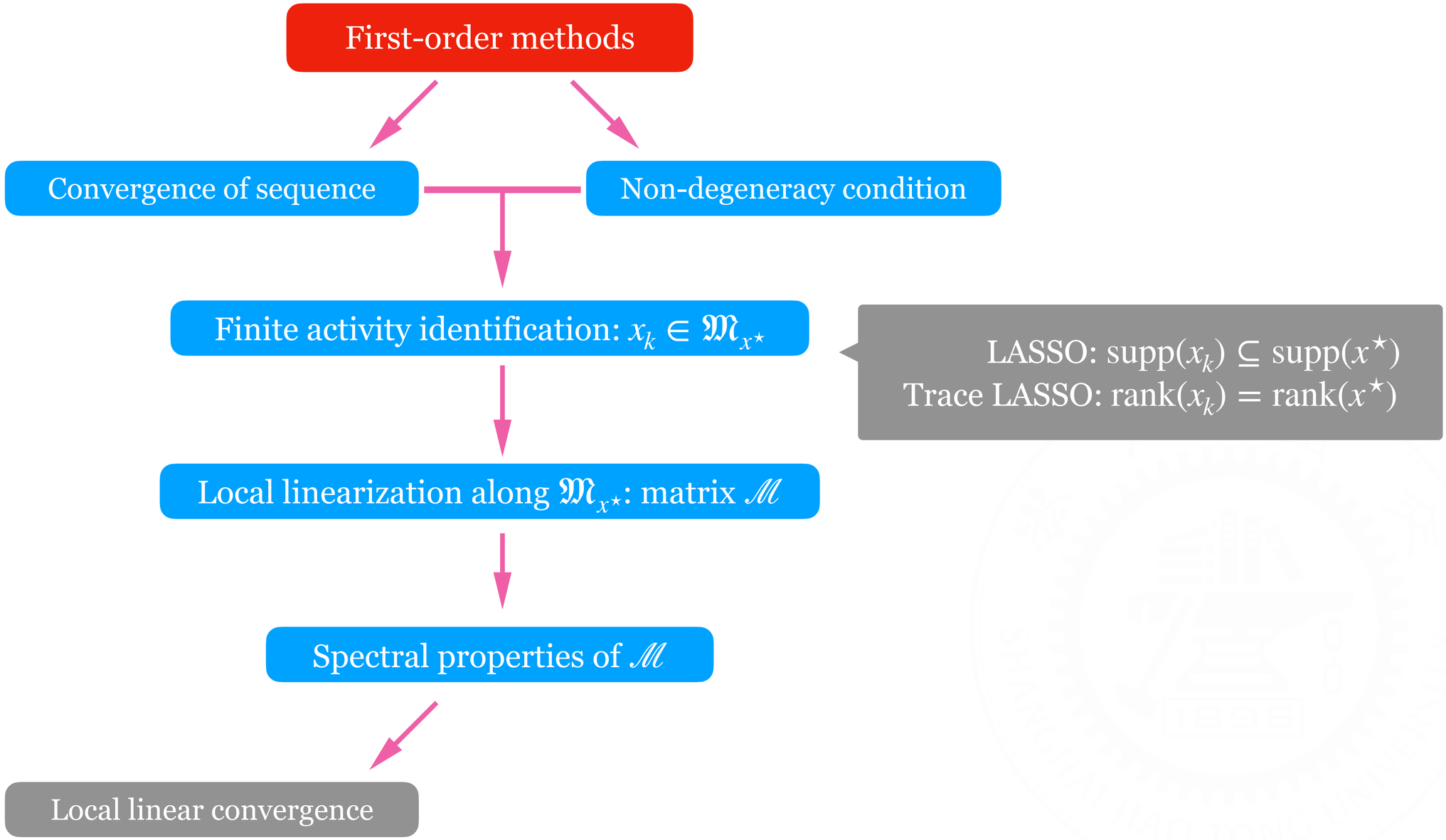


Trajectory of first-order methods

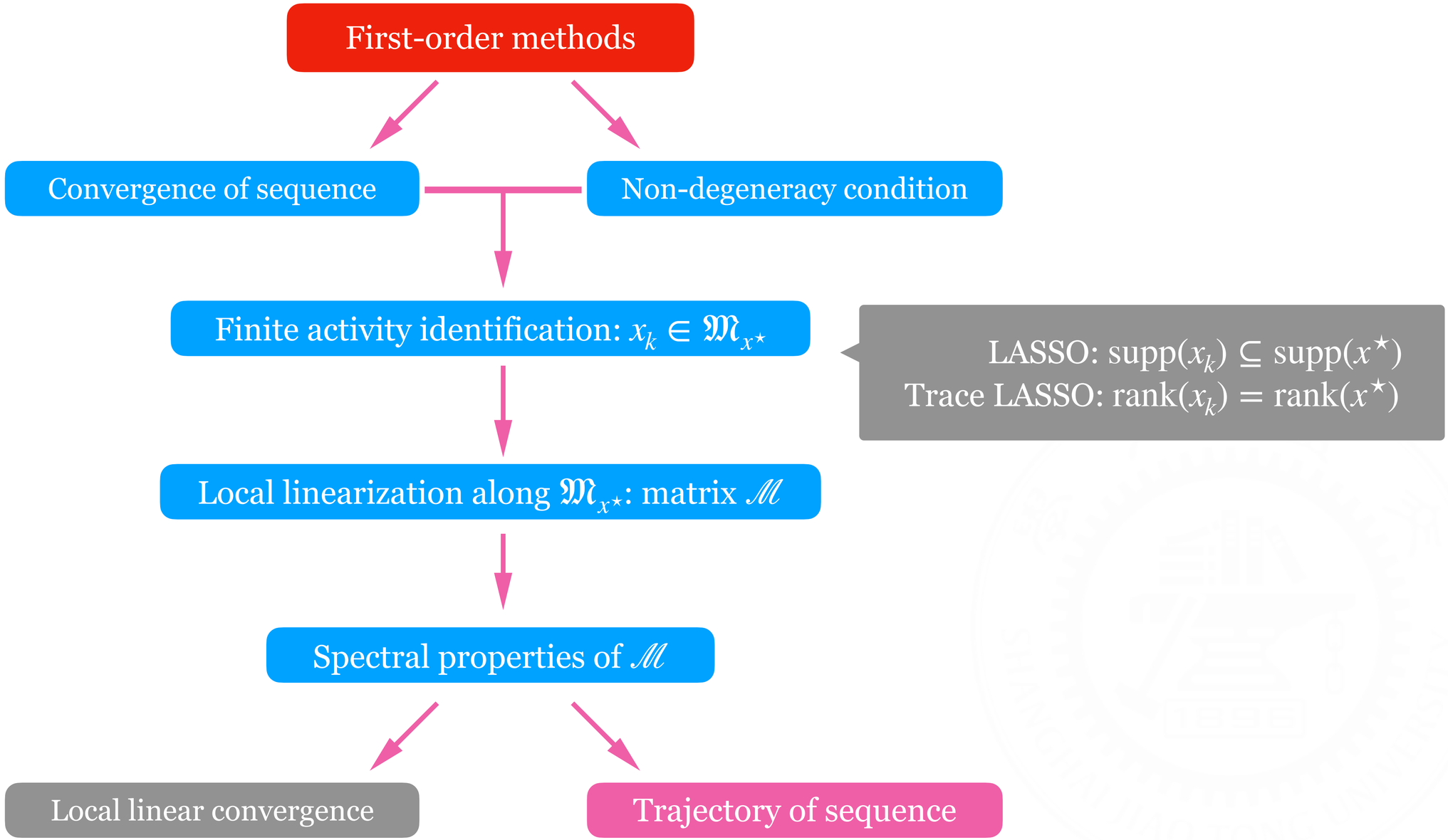


LASSO: $\mathrm{supp}(x_k) \subseteq \mathrm{supp}(x^\star)$
Trace LASSO: $\mathrm{rank}(x_k) = \mathrm{rank}(x^\star)$

Trajectory of first-order methods



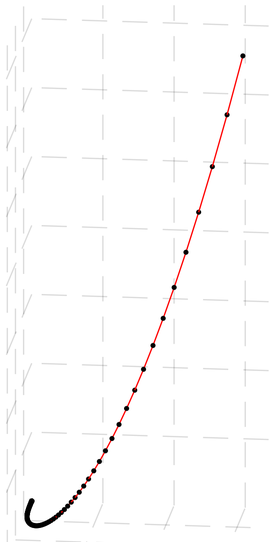
Trajectory of first-order methods



Trajectory of first-order methods

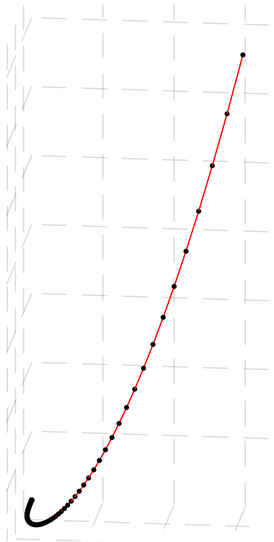
Forward-Backward

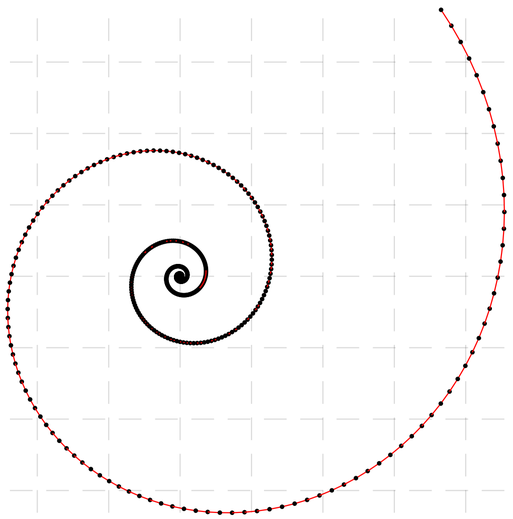
$\mathscr{M}$ is similar to a **symmetric matrix** with real eigenvalues in $]-1,\ +1]$



Trajectory of first-order methods

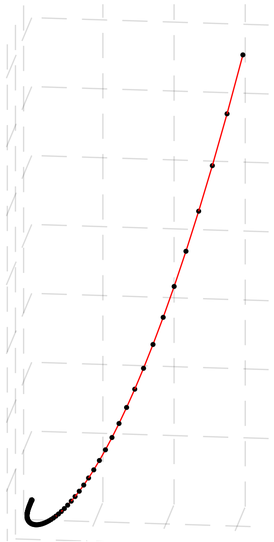
Forward-Backward	Douglas-Rachford/ADMM
$\mathscr{M}$ is similar to a **symmetric matrix** with real eigenvalues in $]-1,\ +1]$	Both functions are **locally polyhedral**, $\mathscr{M}$ is **normal** with complex eigenvalues: $\cos(\theta)e^{\pm \mathrm{i}\theta}$

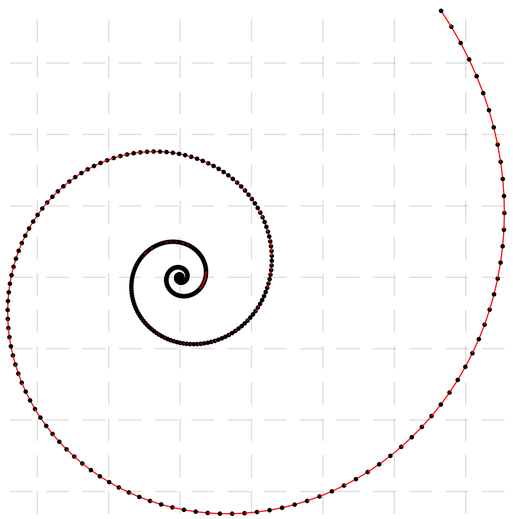


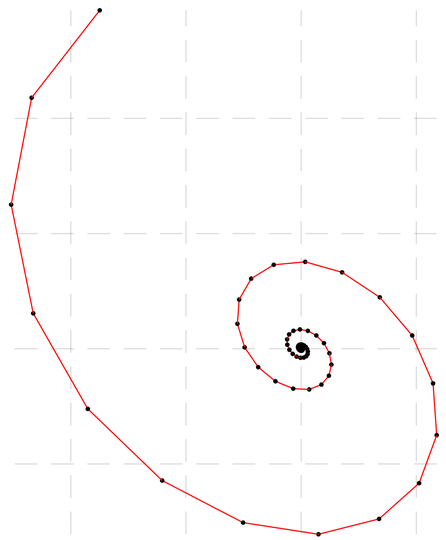


Trajectory of first-order methods

Forward-Backward	Douglas-Rachford/ADMM	Primal-Dual
$\mathscr{M}$ is similar to a **symmetric matrix** with real eigenvalues in $]-1,\ +1]$	Both functions are **locally polyhedral**, $\mathscr{M}$ is **normal** with complex eigenvalues: $\cos(\theta)e^{\pm \mathrm{i}\theta}$	Both functions are **locally polyhedral**, $\mathscr{M}$ is up to orthogonal transform a **block diagonal matrix.**

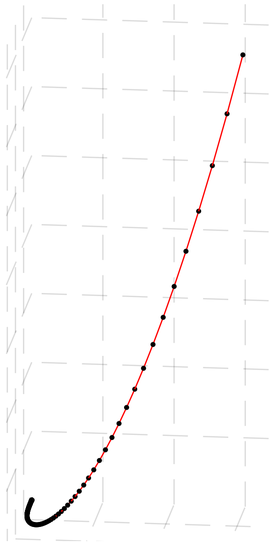


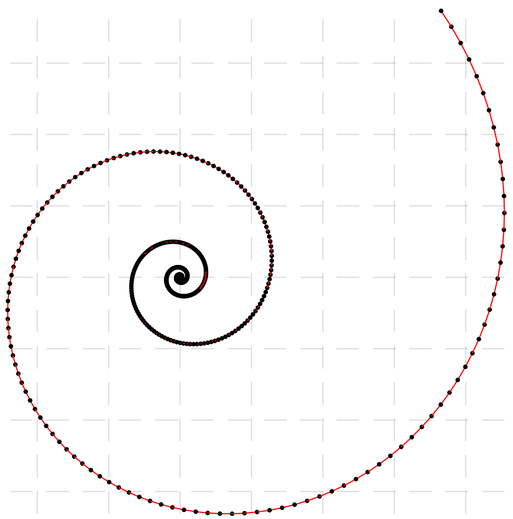


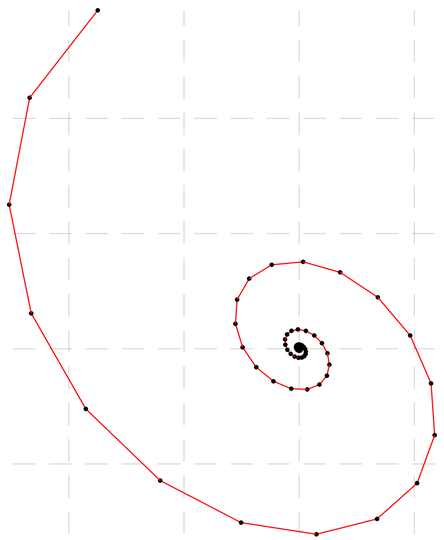


Trajectory of first-order methods

Forward-Backward	Douglas-Rachford/ADMM	Primal-Dual
$\mathscr{M}$ is similar to a **symmetric matrix** with real eigenvalues in $]-1,\ +1]$	Both functions are **locally polyhedral**, $\mathscr{M}$ is **normal** with complex eigenvalues: $\cos(\theta)e^{\pm \mathrm{i}\theta}$	Both functions are **locally polyhedral**, $\mathscr{M}$ is up to orthogonal transform a **block diagonal matrix.**







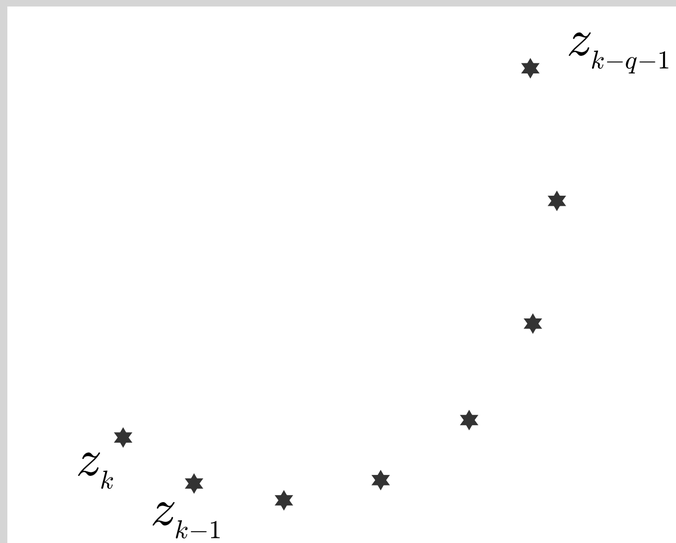
DR/ADMM if smoothness or strong convexity is posed, **straight-line** trajectory can be obtained under proper parameters.

Adaptive acceleration via linear prediction

Linear prediction: illustration

Idea Given past points $\{z_{k-j}\}_{j=0}^{q+1}$, how to predict z_{k+1}?

Define $\{v_{k-j} = z_{k-j} - z_{k-1-j}\}_{j=0}^{q}$.



Linear prediction: illustration

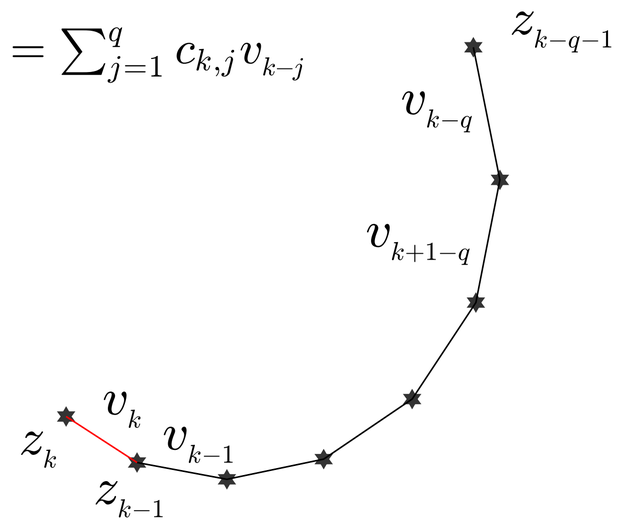
Idea Given past points $\{z_{k-j}\}_{j=0}^{q+1}$, how to predict z_{k+1}?

Define $\{v_{k-j} = z_{k-j} - z_{k-1-j}\}_{j=0}^{q}$.

- Use the directions $\{v_{k-1}, \ldots, v_{k-q}\}$ to fit the current direction v_k:

$$c_k := \arg\min_{c \in \mathbb{R}^q} \| \sum_{j=1}^{q} c_j v_{k-j} - v_k \|.$$

$v_k = \sum_{j=1}^{q} c_{k,j} v_{k-j}$

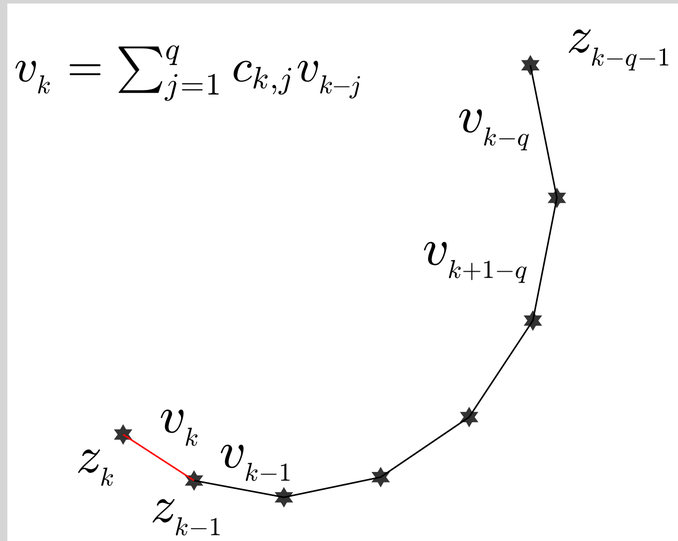


Linear prediction: illustration

Idea Given past points $\{z_{k-j}\}_{j=0}^{q+1}$, how to predict z_{k+1}?

Define $\{v_{k-j} = z_{k-j} - z_{k-1-j}\}_{j=0}^{q}$.

- Use the directions $\{v_{k-1}, \ldots, v_{k-q}\}$ to fit the current direction v_k:
$$c_k := \arg\min_{c \in \mathbb{R}^q} \| \sum_{j=1}^{q} c_j v_{k-j} - v_k \|.$$
- Suppose we know z_{k+1}, then we can compute c_{k+1} as above...

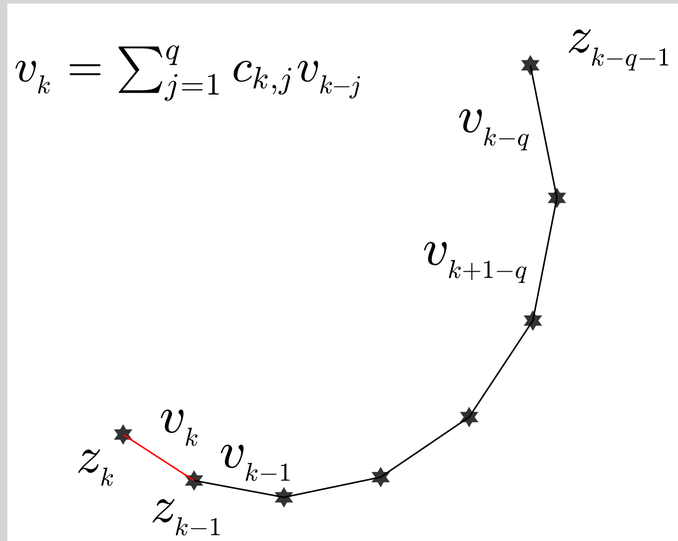


Linear prediction: illustration

Idea Given past points $\{z_{k-j}\}_{j=0}^{q+1}$, how to predict z_{k+1}?

Define $\{v_{k-j} = z_{k-j} - z_{k-1-j}\}_{j=0}^{q}$.

- Use the directions $\{v_{k-1}, \ldots, v_{k-q}\}$ to fit the current direction v_k:
$$c_k := \arg\min_{c \in \mathbb{R}^q} \| \sum_{j=1}^{q} c_j v_{k-j} - v_k \|.$$
- Suppose we know z_{k+1}, then we can compute c_{k+1} as above...
- Approximation
$$v_{k+1} = \sum_j c_{k+1,j} v_{k+1-j} \approx \sum_j c_{k,j} v_{k+1-j}\,.$$



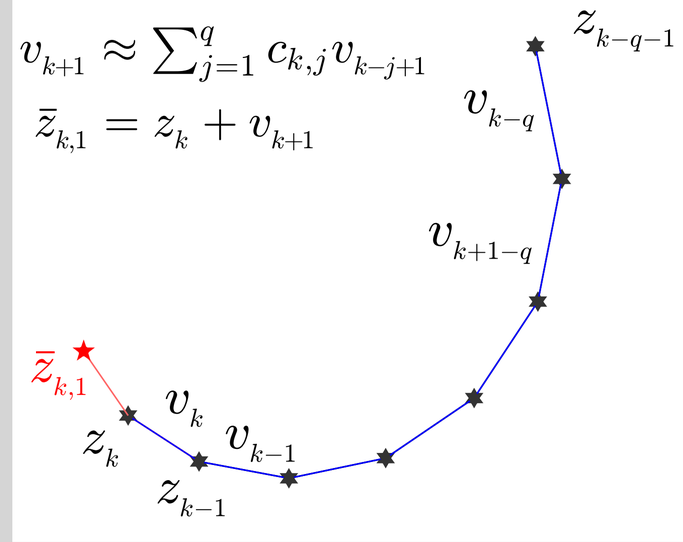
Linear prediction: illustration

Idea Given past points $\{z_{k-j}\}_{j=0}^{q+1}$, how to predict z_{k+1}?

Define $\{v_{k-j} = z_{k-j} - z_{k-1-j}\}_{j=0}^{q}$.

- Use the directions $\{v_{k-1}, \ldots, v_{k-q}\}$ to fit the current direction v_k:
$$c_k := \arg\min_{c \in \mathbb{R}^q} \| \sum_{j=1}^{q} c_j v_{k-j} - v_k \|.$$
- Suppose we know z_{k+1}, then we can compute c_{k+1} as above...
- Approximation
$$v_{k+1} = \sum_j c_{k+1,j} v_{k+1-j} \approx \sum_j c_{k,j} v_{k+1-j}\,.$$
- Let
$$z_{k+1} \approx \bar{z}_{k,1} = z_k + \sum_j c_{k,j} v_{k+1-j}\,.$$

$$v_{k+1} \approx \sum_{j=1}^{q} c_{k,j} v_{k-j+1}$$
$$\bar{z}_{k,1} = z_k + v_{k+1}$$

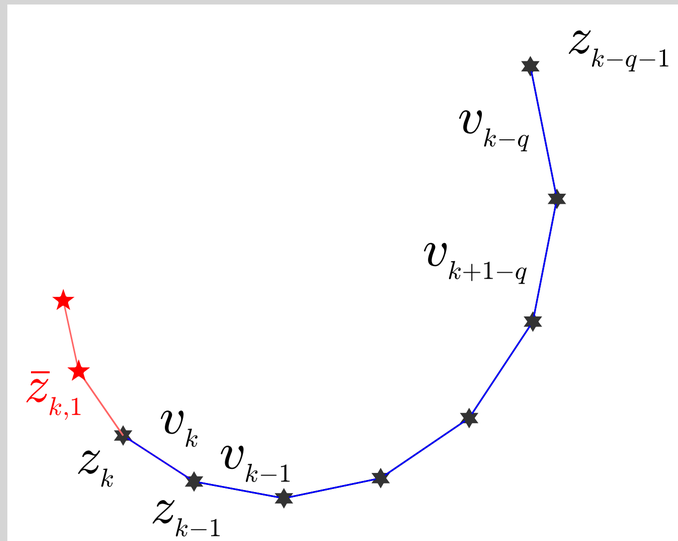


Linear prediction: illustration

Idea Given past points $\{z_{k-j}\}_{j=0}^{q+1}$, how to predict z_{k+1}?

Define $\{v_{k-j} = z_{k-j} - z_{k-1-j}\}_{j=0}^{q}$.

- Repeat on $\{z_{k-j}\}_{j=0}^{q} \cup \{\bar{z}_{k,1}\}$ and so on...

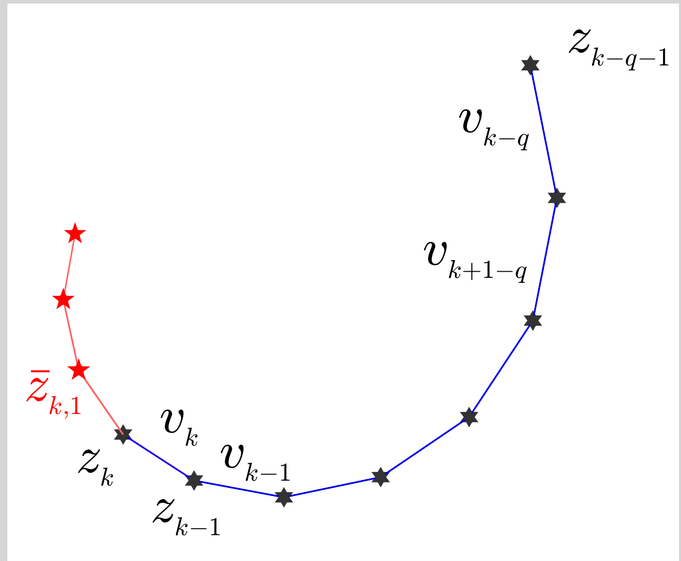


Linear prediction: illustration

Idea Given past points $\{z_{k-j}\}_{j=0}^{q+1}$, how to predict z_{k+1}?

Define $\{v_{k-j} = z_{k-j} - z_{k-1-j}\}_{j=0}^{q}$.

- Repeat on $\{z_{k-j}\}_{j=0}^{q} \cup \{\bar{z}_{k,1}\}$ and so on...



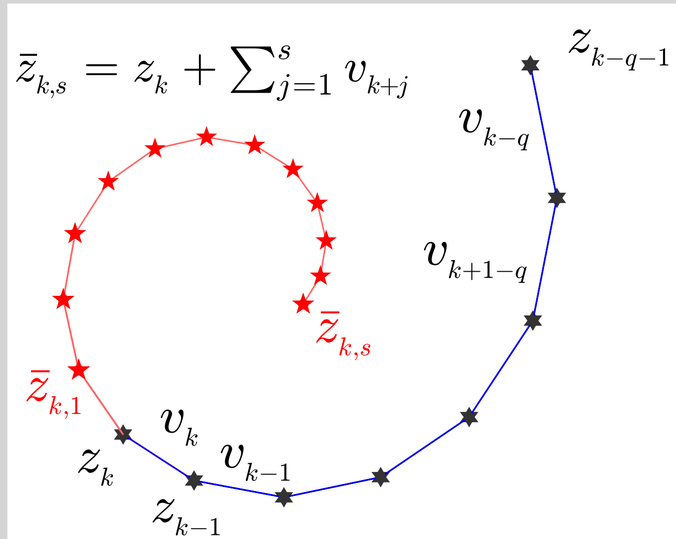
Linear prediction: illustration

Idea Given past points $\{z_{k-j}\}_{j=0}^{q+1}$, how to predict z_{k+1}?

Define $\{v_{k-j} = z_{k-j} - z_{k-1-j}\}_{j=0}^{q}$.

- Repeat on $\{z_{k-j}\}_{j=0}^{q} \cup \{\bar{z}_{k,1}\}$ and so on...
- Repeat s times

$$z_{k+s} \approx \bar{z}_{k,s} = z_k + \sum_{j=1}^{s} v_{k+j}\,.$$



Linear prediction: illustration

Idea Given past points $\{z_{k-j}\}_{j=0}^{q+1}$, how to predict z_{k+1}?

Define $\{v_{k-j} = z_{k-j} - z_{k-1-j}\}_{j=0}^{q}$.

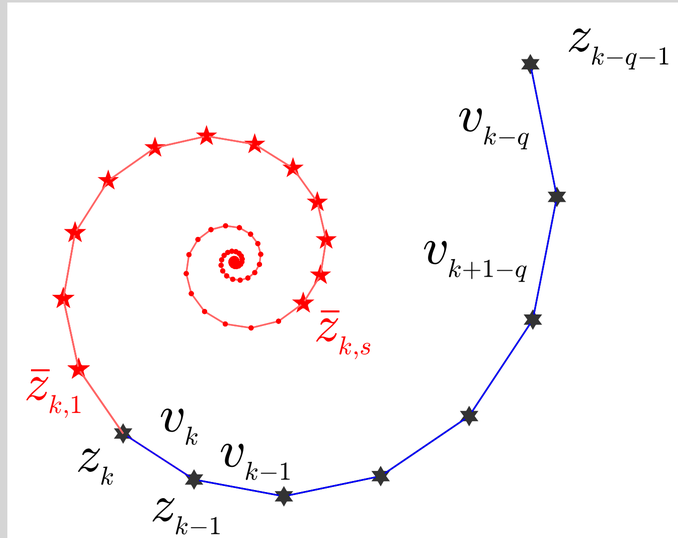
- Repeat on $\{z_{k-j}\}_{j=0}^{q} \cup \{\bar{z}_{k,1}\}$ and so on...

- Repeat s times

$$z_{k+s} \approx \bar{z}_{k,s} = z_k + \sum_{j=1}^{s} v_{k+j}\,.$$

- Let $s = +\infty$, if converges

$$z^\star \approx \bar{z}_{k,+\infty} = z_k + \sum_{j=1}^{+\infty} v_{k+j}\,.$$



Linear prediction: illustration

Idea Given past points $\{z_{k-j}\}_{j=0}^{q+1}$, how to predict z_{k+1}?

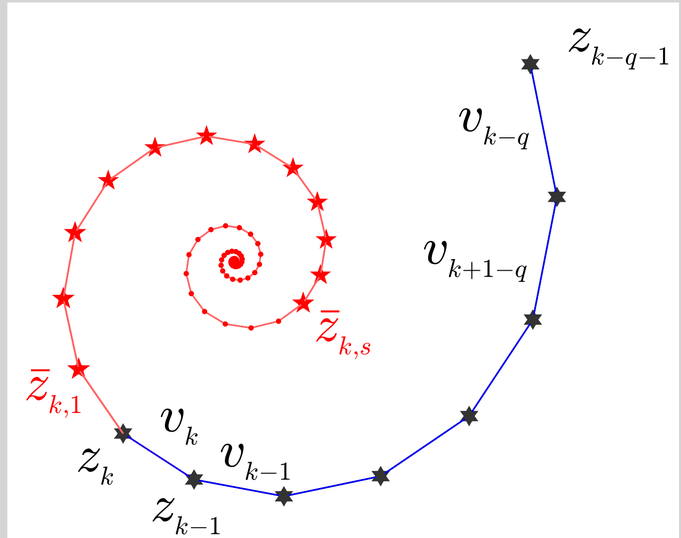
Define $\{v_{k-j} = z_{k-j} - z_{k-1-j}\}_{j=0}^{q}$.

- Repeat on $\{z_{k-j}\}_{j=0}^{q} \cup \{\bar{z}_{k,1}\}$ and so on...
- Repeat s times

$$z_{k+s} \approx \bar{z}_{k,s} = z_k + \sum_{j=1}^{s} v_{k+j}\,.$$

- Let $s = +\infty$, if converges

$$z^\star \approx \bar{z}_{k,+\infty} = z_k + \sum_{j=1}^{+\infty} v_{k+j}\,.$$



The s-step linear prediction is $\bar{z}_{k,s} = z_k + \mathscr{E}_{s,q,k}$, where $\mathscr{E}_{s,q,k} = \sum_{j=1}^{q} \hat{c}_j v_{k+1-j}$ and

$$\hat{c} := \left(\sum_{i=1}^{s} [H(c_k)]^i\right)_{[:,1]} \quad \text{and} \quad H(c) := \begin{bmatrix} c_{[1:q-1]} & \mathrm{Id}_{q-1} \\ c_q & 0 \end{bmatrix}.$$

Adaptive acceleration for FoM

Given first-order method

$$z_{k+1} = \mathscr{F}(z_k)\,.$$

A^{2}FoM via linear prediction

Let $s \geq 1$, $q \geq 1$be integers, $z_0 \in \mathbb{R}^n$ and$\bar{z}_0 = z_0$, set $V_0 = \mathbf{0} \in \mathbb{R}^{n\times(q+1)}$:

- **For** $k \geq 1$

$$z_k = \mathscr{F}(\bar{z}_{k-1}),$$
$$v_k = z_k - z_{k-1},$$
$$V_k = \left[v_k, V_{k-1}[:,1:q]\right].$$

- If $\mathrm{mod}(k, q+2) = 0$, compute c and H_c

$$\text{If } \rho(H_c) < 1\text{: } \bar{z}_k = z_k + V_k\left(\sum\nolimits_{i=1}^{s} H_c^i\right)_{[:,1]};$$
$$\text{else: } \bar{z}_k = z_k;$$

If $\mathrm{mod}(k, q+2) \neq 0$, $\bar{z}_k = z_k$.

Remarks

- Every $(q+2)$-iteration we apply 1-step **LP**.
- Only apply the linear prediction when $\rho(H_c) < 1$.
- Extra memory cost $n \times (q+1)$ (the difference vector matrix). Usually $q \leq 10$.
- Extra computation cost, $q^2 n$ from V_k^+.

Remarks

- Every $(q+2)$-iteration we apply 1-step **LP**.
- Only apply the linear prediction when $\rho(H_c) < 1$.
- Extra memory cost $n \times (q+1)$ (the difference vector matrix). Usually $q \leq 10$.
- Extra computation cost, $q^2 n$ from V_k^+.

- **Conditional convergence:** treat LP as **perturbation error**, *e.g.*

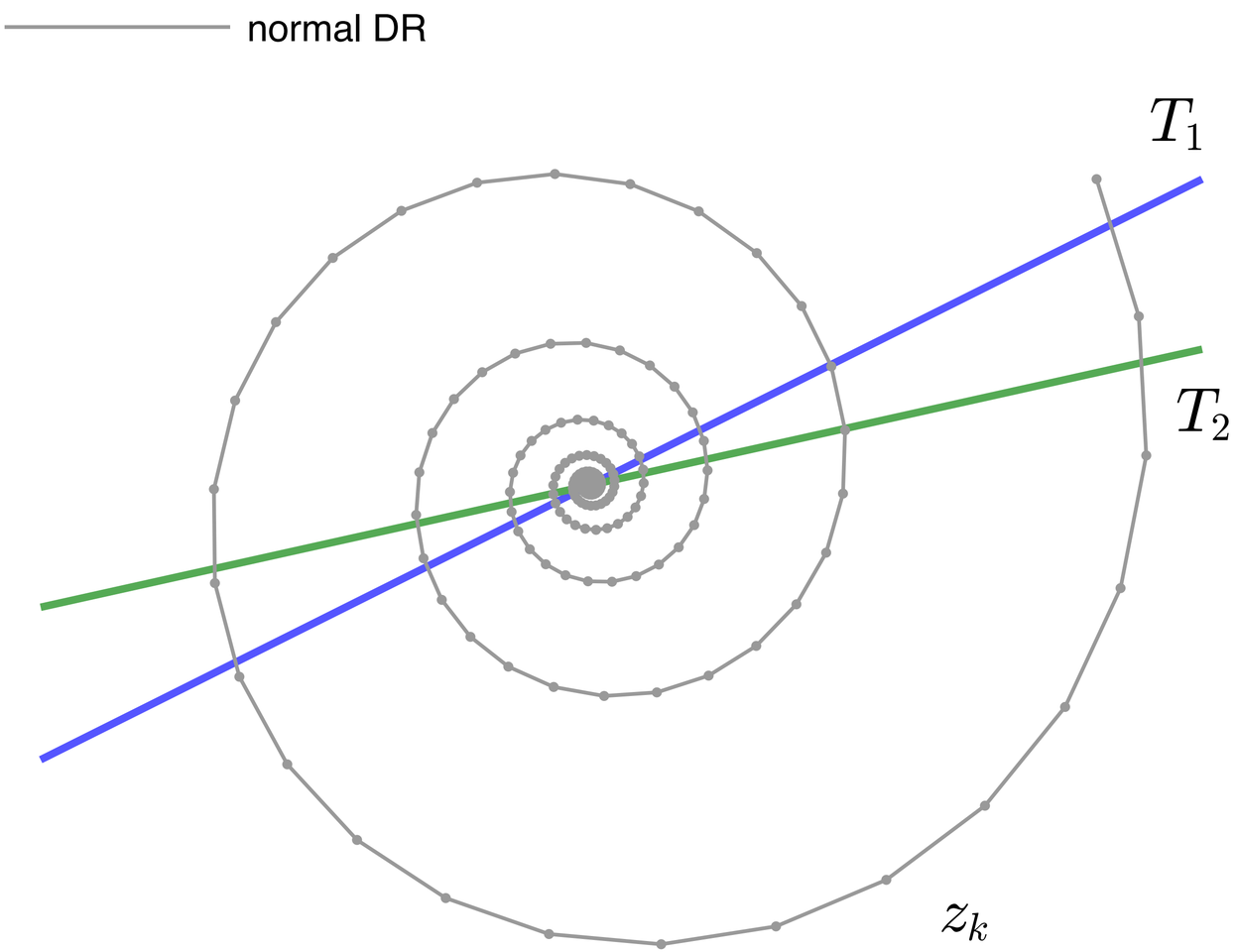
$$z_{k+1} = \mathscr{F}(z_k + \epsilon_k)$$

- **Safeguarded LP**

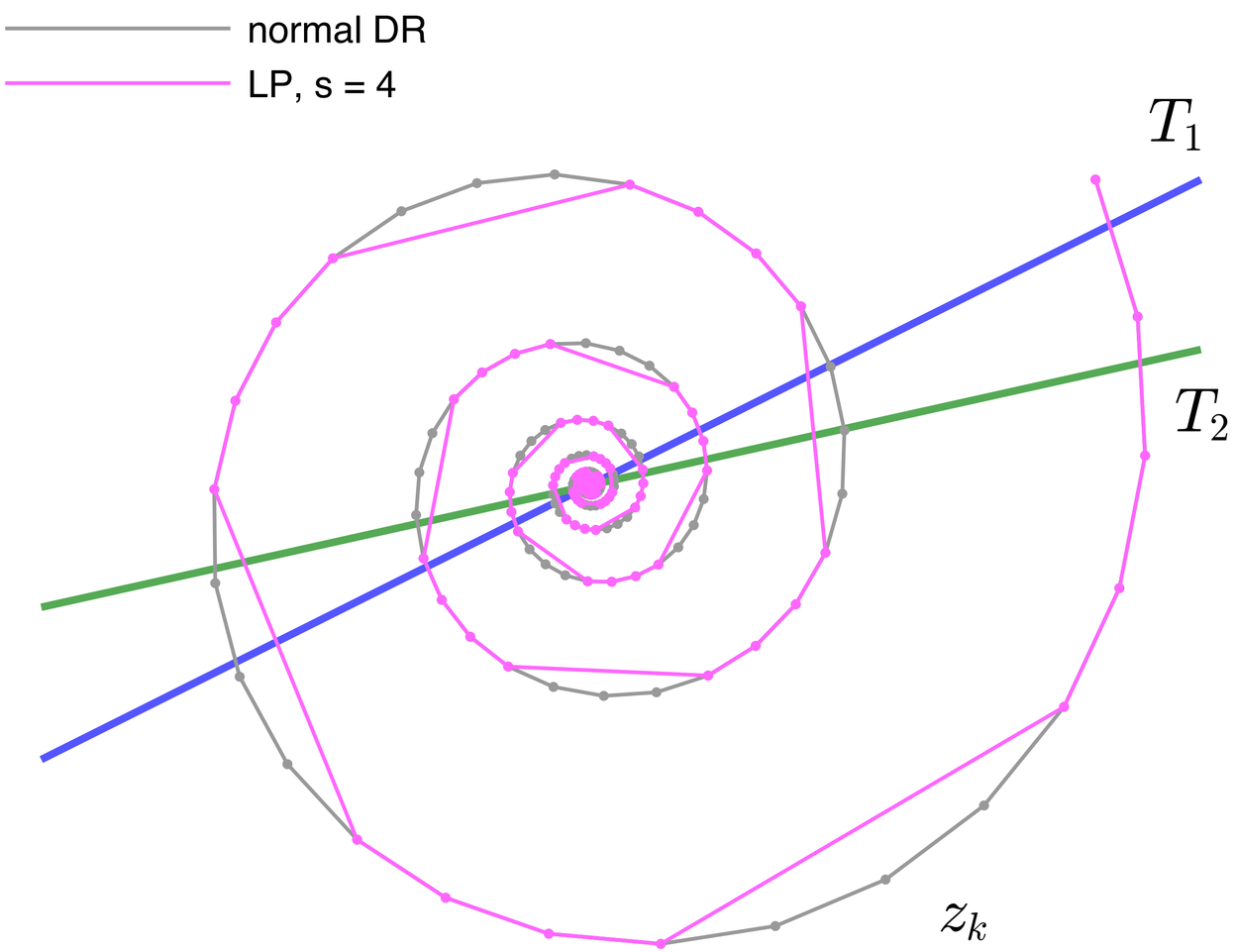
$$\bar{z}_k = z_k + a_k V_k \left(\sum_{i=1}^{s} H_c^i \right)_{[:,1]}$$

with a_k updated online.

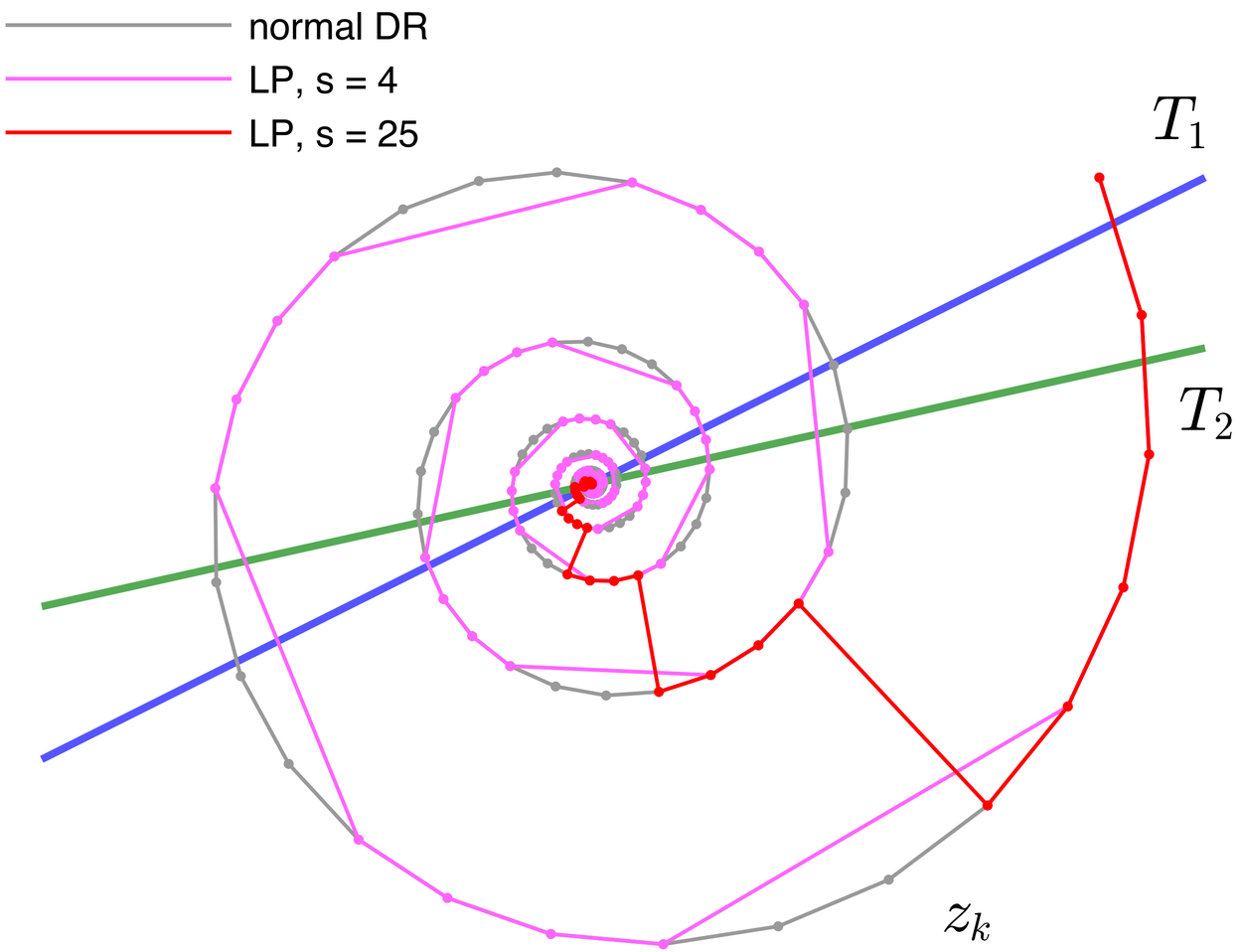
Illustration



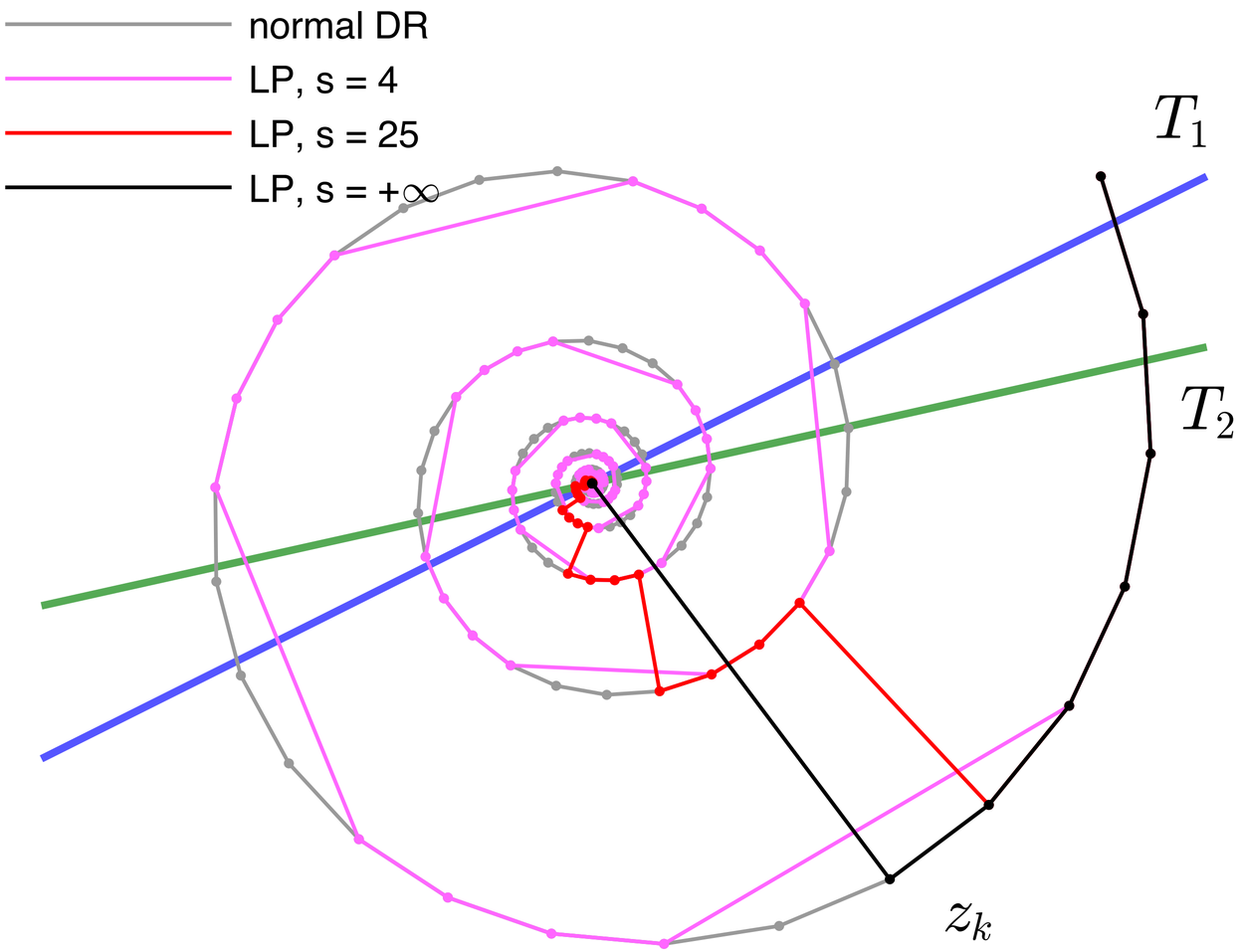
Illustration



Illustration



Illustration



Relation with previous work

Convergence acceleration

Given a sequence $\{z_k\}_k$ which converges to $z^\star$. Can we generate another sequence $\{\bar{z}_k\}_k$ such that $\|\bar{z}_k - z^\star\| = o(\|z_k - z^\star\|)$?

This is called **convergence acceleration** and is well-established in numerical analysis:

1927	Aitkin's Δ-process.
1965	Andersen acceleration.
1970's	Vector extrapolation techniques such as Minimal Polynomial Extrapolation (**MPE**) and Reduced Rank Extrapolation (**RRE**) [Sidi '17].
Renaissance	❖ Regularized Non-linear Acceleration (**RNA**) is a regularized version of RRE introduced by [Scieur, D'Aspremont, Bach '16]. ❖ Anderson acceleration for proximal methods [Zhang et al '18; Fu et al '20...].

Polynomial extrapolation

Polynomial extrapolation [Cabay & Jackson '76]

Consider $z_k = Mz_{k-1} + d$ with $\rho(M) < 1$ such that $z_k \to z^\star$

- $z_k - z^\star = M(z_{k-1} - z^\star) = M^k(z_0 - z^\star)$.

Polynomial extrapolation

Polynomial extrapolation [Cabay & Jackson '76]

Consider $z_k = Mz_{k-1} + d$ with $\rho(M) < 1$ such that $z_k \to z^\star$

- $z_k - z^\star = M(z_{k-1} - z^\star) = M^k(z_0 - z^\star)$.
- If $\mathscr{P}(\lambda) = \sum_{j=0}^{q} c_j \lambda^j$ is the minimal polynomial of M w.r.t. $z_o - z^\star$, that is

$$\mathscr{P}(M)(z_0 - z^\star) = \sum_{j=0}^{q} c_j M^j (z_0 - z^\star) = \sum_{j=0}^{q} c_j (z_j - z^\star)$$

then $z^\star = \dfrac{\sum_j c_j z_j}{\sum_j c_j}$

Polynomial extrapolation

Polynomial extrapolation [Cabay & Jackson '76]

Consider $z_k = Mz_{k-1} + d$ with $\rho(M) < 1$ such that $z_k \to z^\star$

- $z_k - z^\star = M(z_{k-1} - z^\star) = M^k(z_0 - z^\star)$.
- If $\mathscr{P}(\lambda) = \sum_{j=0}^{q} c_j \lambda^j$ is the minimal polynomial of M w.r.t. $z_o - z^\star$, that is

$$\mathscr{P}(M)(z_0 - z^\star) = \sum_{j=0}^{q} c_j M^j (z_0 - z^\star) = \sum_{j=0}^{q} c_j (z_j - z^\star)$$

then $z^\star = \dfrac{\sum_j c_j z_j}{\sum_j c_j}$

- The coefficient c can be computed w.o. the knowledge of $z^\star$: $c_1 = 1$

$$V_q c_{[1:q-1]} = -v_{q+1}$$

with $V_q = \left[v_1 \,|\, v_2 \,|\, \ldots \,|\, v_q\right]$ and $v_j = z_j - z_{j-1}$

Vector extrapolation technique

Vector extrapolation methods with applications (SIAM, 2017) by Avram Sidi.

Given a sequence generated by $z_{k+1} = \mathscr{F}(z_k)$.

Minimal polynomial extrapolation []

Let $z_0 = \bar{z}$.

S.1 Generating $\{z_j\}_{j=0}^{q}$ and let $v_j = z_j - z_{j-1}$.

Vector extrapolation technique

Vector extrapolation methods with applications (SIAM, 2017) by Avram Sidi.

Given a sequence generated by $z_{k+1} = \mathscr{F}(z_k)$.

Minimal polynomial extrapolation []

Let $z_0 = \bar{z}$.

S.1 Generating $\{z_j\}_{j=0}^{q}$ and let $v_j = z_j - z_{j-1}$.

S.2 Let $c \in \mathbb{R}^{q+1}$ such that $c_q = 1$ and

$$V_q c_{[1:q-1]} = -v_{q+1}.$$

For $j = 0,\ldots,q-1$, let $\hat{c}_j = c_j / \sum_{j=0}^{q} c_j$

Vector extrapolation technique

Vector extrapolation methods with applications (SIAM, 2017) by Avram Sidi.

Given a sequence generated by $z_{k+1} = \mathscr{F}(z_k)$.

Minimal polynomial extrapolation []

Let $z_0 = \bar{z}$.

S.1 Generating $\{z_j\}_{j=0}^{q}$ and let $v_j = z_j - z_{j-1}$.

S.2 Let $c \in \mathbb{R}^{q+1}$ such that $c_q = 1$ and

$$V_q c_{[1:q-1]} = -v_{q+1}.$$

For $j = 0,\ldots,q-1$, let $\hat{c}_j = c_j / \sum_{j=0}^{q} c_j$

S.3 $\bar{z} = \sum_j \hat{c}_j z_j$

Vector extrapolation technique

Vector extrapolation methods with applications (SIAM, 2017) by Avram Sidi.

Given a sequence generated by $z_{k+1} = \mathscr{F}(z_k)$.

Minimal polynomial extrapolation []

Let $z_0 = \bar{z}$.

S.1 Generating $\{z_j\}_{j=0}^{q}$ and let $v_j = z_j - z_{j-1}$.

S.2 Let $c \in \mathbb{R}^{q+1}$ such that $c_q = 1$ and

$$V_q c_{[1:q-1]} = -v_{q+1}.$$

For $j = 0,\ldots,q-1$, let $\hat{c}_j = c_j / \sum_{j=0}^{q} c_j$

S.3 $\bar{z} = \sum_j \hat{c}_j z_j$

Reduced rank extrapolation [Andersen '65; Kaniel & Stein '74; Eddy '79; Mesina '77]

Vector extrapolation technique

Vector extrapolation methods with applications (SIAM, 2017) by Avram Sidi.

Given a sequence generated by $z_{k+1} = \mathscr{F}(z_k)$.

Minimal polynomial extrapolation []

Let $z_0 = \bar{z}$.

S.1 Generating $\{z_j\}_{j=0}^{q}$ and let $v_j = z_j - z_{j-1}$.

S.2 Let $c \in \mathbb{R}^{q+1}$ such that $c_q = 1$ and

$$V_q c_{[1:q-1]} = -v_{q+1}.$$

For $j = 0,\ldots,q-1$, let $\hat{c}_j = c_j / \sum_{j=0}^{q} c_j$

S.3 $\bar{z} = \sum_j \hat{c}_j z_j$

Reduced rank extrapolation [Andersen '65; Kaniel & Stein '74; Eddy '79; Mesina '77]

Replace step **S.2** by

$$V_q c_{[1:q-1]} = -v_{q+1} \quad \text{subject to} \quad \mathbf{1}^T c = 1.$$

Vector extrapolation technique

Vector extrapolation methods with applications (SIAM, 2017) by Avram Sidi.

Given a sequence generated by $z_{k+1} = \mathscr{F}(z_k)$.

Minimal polynomial extrapolation []

Let $z_0 = \bar{z}$.

S.1 Generating $\{z_j\}_{j=0}^{q}$ and let $v_j = z_j - z_{j-1}$.

S.2 Let $c \in \mathbb{R}^{q+1}$ such that $c_q = 1$ and

$$V_q c_{[1:q-1]} = -v_{q+1}.$$

For $j = 0,\ldots,q-1$, let $\hat{c}_j = c_j / \sum_{j=0}^{q} c_j$

S.3 $\bar{z} = \sum_j \hat{c}_j z_j$

Reduced rank extrapolation [Andersen '65; Kaniel & Stein '74; Eddy '79; Mesina '77]

Replace step **S.2** by

$$V_q c_{[1:q-1]} = -v_{q+1} \quad \text{subject to} \quad \mathbf{1}^T c = 1.$$

LP is equivalent to **MPE** with **S.3** replaced by $\bar{z} := \sum_{j=0}^{q} \tilde{c}_j z_{j+1}$.

Relation between MPE and LP

Our derivation

- is based on the sequence **trajectory**.
- motivates checking $\rho(H_c) < 1$.

Relation between MPE and LP

Our derivation

- is based on the sequence **trajectory**.
- motivates checking $\rho(H_c) < 1$.

Example: solve with DR

$$\min_x \; R(x) \quad \text{such that} \quad Ax = b\,.$$

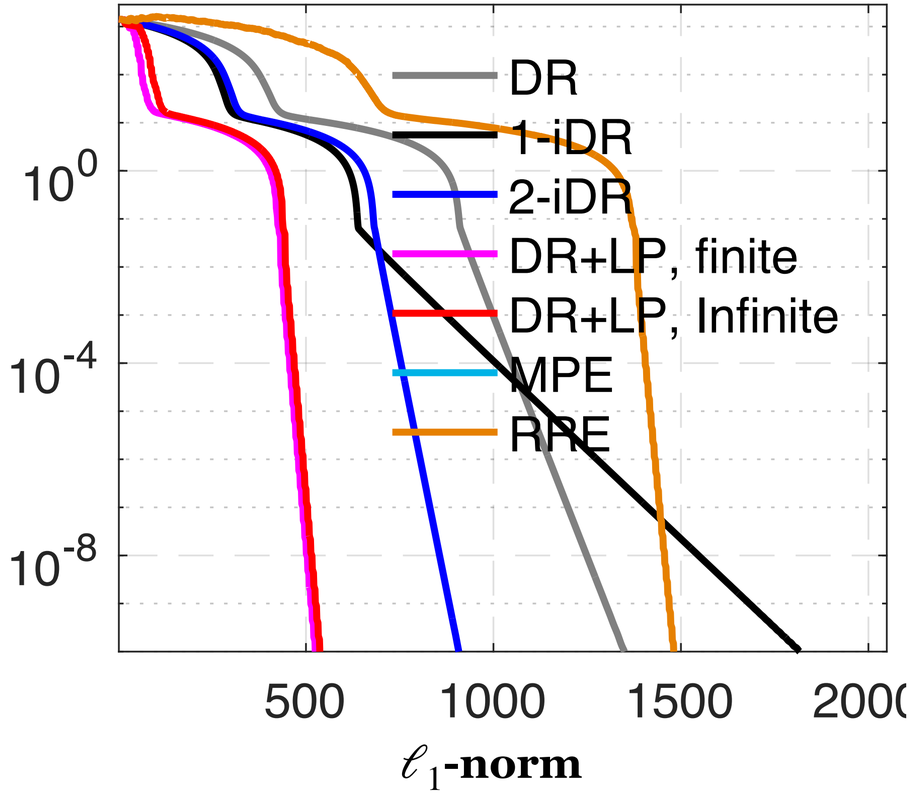
Relation between MPE and LP

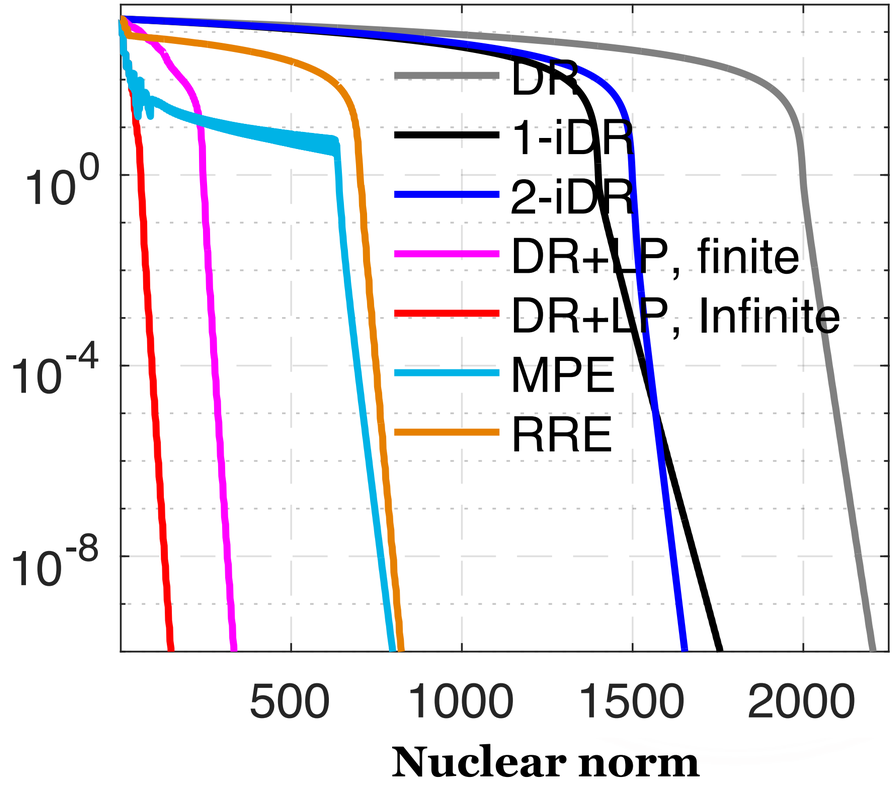
Our derivation

- is based on the sequence **trajectory**.
- motivates checking $\rho(H_c) < 1$.

Example: solve with DR

$$\min_x \; R(x) \quad \text{such that} \quad Ax = b\,.$$





Acceleration guarantees

When $z_{k+1} - z_k = M(z_k - z_{k-1})$

$$\|\bar{z}_{k,s} - z^\star\| \leq \|z_{k+s} - z^\star\| + B\epsilon_k$$

where $\epsilon_k = \|V_{k-1}c - v_k\|$ and $B = \sum_{\ell=1}^{s} \|M^\ell\| |\sum_{i=0}^{s-\ell} (H_c^i)_{[1,1]}|$.

Acceleration guarantees

When $z_{k+1} - z_k = M(z_k - z_{k-1})$

$$\|\bar{z}_{k,s} - z^\star\| \leq \|z_{k+s} - z^\star\| + B\epsilon_k$$

where $\epsilon_k = \|V_{k-1}c - v_k\|$ and $B = \sum_{\ell=1}^{s} \|M^\ell\| |\sum_{i=0}^{s-\ell} (H_c^i)_{[1,1]}|$.

Asymptotic bound ($k \to +\infty$)

$$\epsilon_k = O(|\lambda_{q+1}|^k)$$

where λ_{q+1} is the $(q+1)^{th}$ largest eigenvalue. Without extrapolation, we just have $O(|\lambda_1|^k)$.

Acceleration guarantees

When $z_{k+1} - z_k = M(z_k - z_{k-1})$

$$\|\bar{z}_{k,s} - z^\star\| \leq \|z_{k+s} - z^\star\| + B\epsilon_k$$

where $\epsilon_k = \|V_{k-1}c - v_k\|$ and $B = \sum_{\ell=1}^{s} \|M^\ell\| \left|\sum_{i=0}^{s-\ell} (H_c^i)_{[1,1]}\right|$.

Asymptotic bound ($k \to +\infty$)

$$\epsilon_k = O(|\lambda_{q+1}|^k)$$

where λ_{q+1} is the $(q+1)^{th}$ largest eigenvalue. Without extrapolation, we just have $O(|\lambda_1|^k)$.

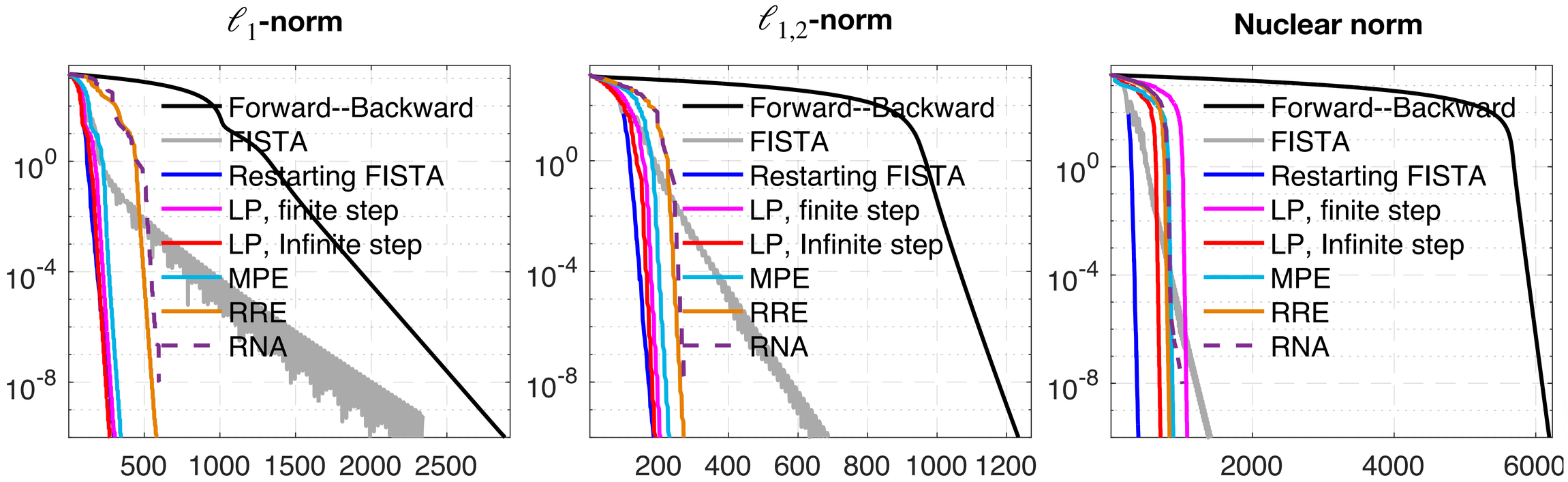
Non-asymptotic bound if $\Sigma(M) \subset [\alpha, \beta]$ with $-1 < \alpha < \beta < 1$

$$B\epsilon_k \leq K\beta^{k-q}\left(\frac{\sqrt{\eta}-1}{\sqrt{\eta}+1}\right)^q \qquad \text{where} \qquad \eta = \frac{1-\alpha}{1-\beta}$$

- Similar error bounds also hold for MPE and RRE [Sidi '98].
- For DR & PD with polyhedral functions, guaranteed acceleration for $q = 2$.

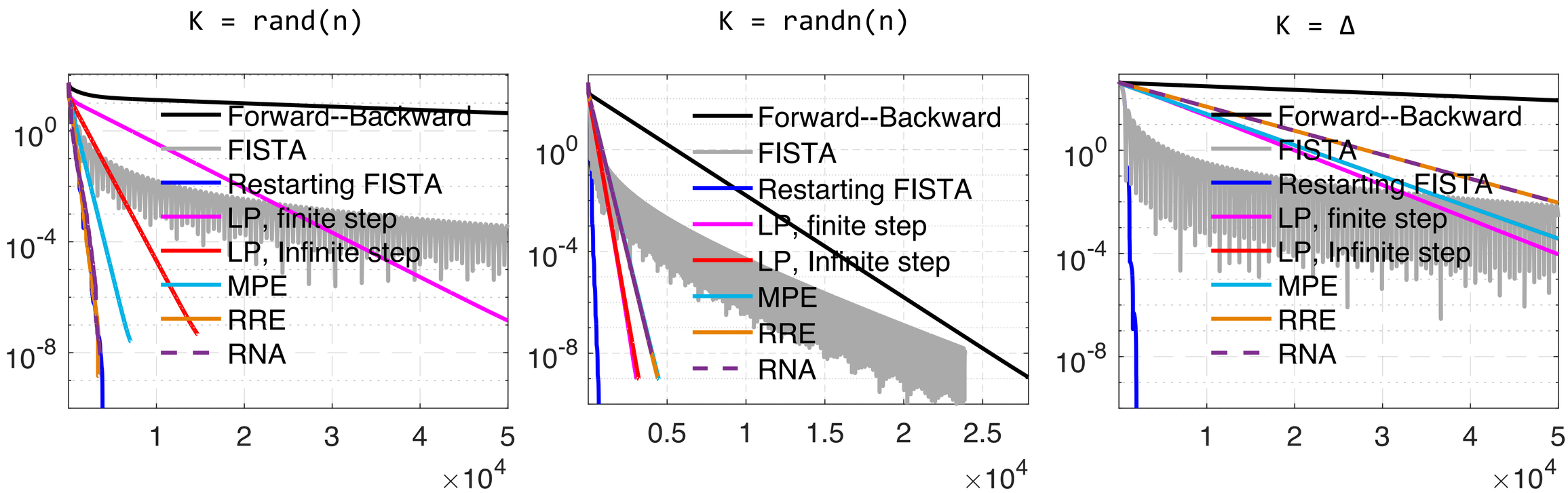
LASSO-type problems with FB

NB recall the straight-line trajectory of FB.



Note the performance of restarted-FISTA [O'Donoghue & Candès '12]!

Least squares with gradient descent



Note the performance of restarted-FISTA [O'Donoghue & Candès '12]!

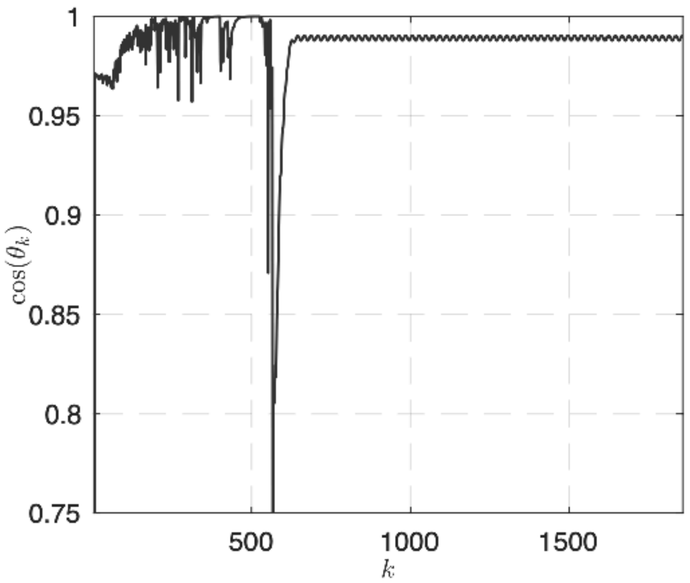
Numerical experiments

TV-based image inpainting

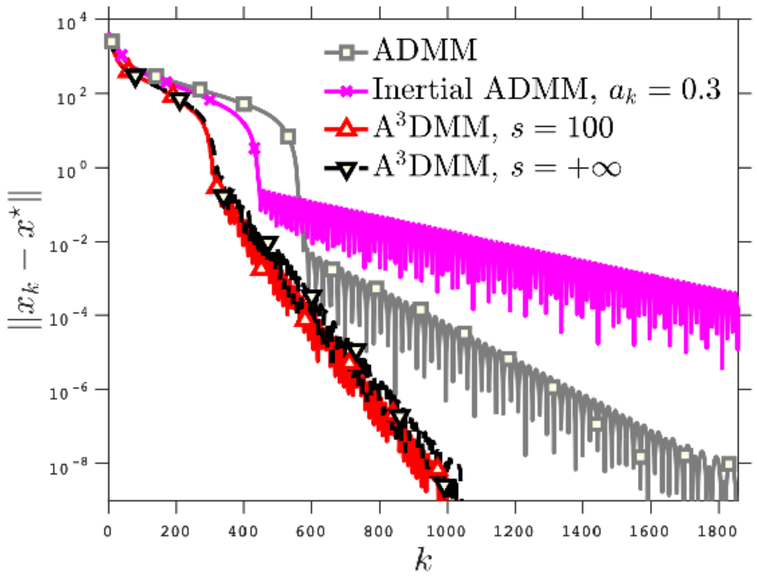
Let $\Omega \overset{\text{def}}{=} \{x \in \mathbb{R}^{n\times n} : P_{\mathcal{D}}(x) = f\}$, $P_{\mathcal{D}}$ randomly sets 50% pixels to zero and consider

$$\min_{x\in\mathbb{R}^{n\times n}} \|y\|_1 + \iota_\Omega(x) \quad \text{such that} \quad \nabla x - y = 0.$$

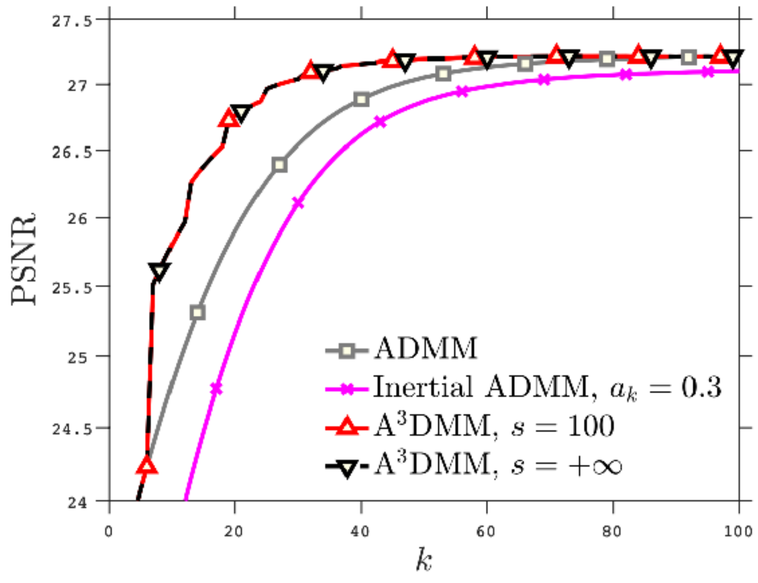
Angle $\{\theta_k\}$



$\|x_k - x^\star\|$



PSNR



- Both functions are polyhedral, trajectory is a spiral.
- Inertial ADMM is **slower** than ADMM.

TV-based image inpainting

Original image

ADMM, PSNR = 26.5448

Inertial ADMM, PSNR = 26.1096

Corrupted image

A^{3}DMM $s = 100$, PSNR = 27.0402

A^{3}DMM $s = +\infty$, PSNR = 27.0402

Take-away messages

Trajectory of FoM

- Local linearization of FoM.
- Different FoMs demonstrate different trajectories: straight line and spirals.

Take-away messages

Trajectory of FoM

- Local linearization of FoM.
- Different FoMs demonstrate different trajectories: straight line and spirals.

An adaptive acceleration for FoM

- Though motivated by local trajectory, works **globally**.
- For polyhedral functions, guaranteed acceleration for DR/PD using 4 points.
- For FB, guaranteed acceleration using 3 points, but better using **restarted-FISTA**.

Take-away messages

Trajectory of FoM

- Local linearization of FoM.
- Different FoMs demonstrate different trajectories: straight line and spirals.

An adaptive acceleration for FoM

- Though motivated by local trajectory, works **globally**.
- For polyhedral functions, guaranteed acceleration for DR/PD using 4 points.
- For FB, guaranteed acceleration using 3 points, but better using **restarted-FISTA**.

Outlook

- Better geometric descriptions.
- Adaptive acceleration for non-convex optimization, stochastic optimization.

Take-away messages

Trajectory of FoM

- Local linearization of FoM.
- Different FoMs demonstrate different trajectories: straight line and spirals.

An adaptive acceleration for FoM

- Though motivated by local trajectory, works **globally**.
- For polyhedral functions, guaranteed acceleration for DR/PD using 4 points.
- For FB, guaranteed acceleration using 3 points, but better using **restarted-FISTA**.

Outlook

- Better geometric descriptions.
- Adaptive acceleration for non-convex optimization, stochastic optimization.

Thank you very much for your attention!

https://jliang993.github.io/