Less Pessimistic Analysis of First-Order Methods

Exploiting Structure and Data

Bartolomeo Stellato

Department of Operations Research and Financial Engineering

PRINCETON UNIVERSITY

Sierra Seminar — June 17 2025




Most applications require fast and effective decisions in real-time

First-order methods are now widely popular…

Parametric Convex Optimization

decisions

$$\begin{array}{ll} \text{minimize} & f(z, x) \\ \text{subject to} & z \in C(x) \end{array}$$

parameters

same problem with
varying parameters

$$x \sim \mathbf{P}$$

(unknown distribution)

First-order methods are now widely popular…

Parametric Convex Optimization

decisions

$$\begin{array}{ll} \text{minimize} & f(z, x) \\ \text{subject to} & z \in C(x) \end{array}$$

parameters

same problem with
varying parameters

$$x \sim \mathbf{P}$$

(unknown distribution)

example
projected gradient descent

$$z^{k+1} = \underbrace{\Pi_{C(x)}}_{\text{projection}}\big(\underbrace{z^k - \theta \nabla f(z^k, x)}_{\text{gradient step}}\big)$$

First-order methods are now widely popular…

Parametric Convex Optimization

decisions

$$\begin{array}{ll} \text{minimize} & f(z, x) \\ \text{subject to} & z \in C(x) \end{array}$$

parameters

same problem with varying parameters

$$x \sim \mathbf{P}$$

(unknown distribution)

example
projected gradient descent

$$z^{k+1} = \underbrace{\Pi_{C(x)}}_{\text{projection}}\big(\underbrace{z^k - \theta \nabla f(z^k, x)}_{\text{gradient step}}\big)$$

benefits of first-order methods

✓ cheap iterations

✓ easy to warm-start

embedded optimization

large-scale optimization

…and they can solve many constrained convex problems!

Linear Programs

PDLP

Applegate, Díaz, Hinder, Lu, Lubin, O'Donoghue, Schudy (2021)

NVIDIA cuOPT

…and they can solve many constrained convex problems!

Linear Programs

PDLP

Applegate, Díaz, Hinder, Lu, Lubin, O'Donoghue, Schudy (2021)

NVIDIA cuOPT

Quadratic Programs

OSQP

Stellato, Banjac, Goulart, Bemporad, Boyd (2020)

ProxSuite

THE ADVANCED PROXIMAL OPTIMIZATION TOOLBOX

Bambade, Schramm, El Kazdadi, Caron, Taylor, Carpentier (2025)

…and they can solve many constrained convex problems!

Linear Programs

PDLP

Applegate, Díaz, Hinder, Lu, Lubin, O'Donoghue, Schudy (2021)

NVIDIA cuOPT

Quadratic Programs

OSQP

Stellato, Banjac, Goulart, Bemporad, Boyd (2020)

ProxSuite — The Advanced Proximal Optimization Toolbox

Bambade, Schramm, El Kazdadi, Caron, Taylor, Carpentier (2025)

Conic Programs

SCS — Splitting Conic Solver

O'Donoghue, Chu, Parikh, Boyd (2016)

…and they can solve many constrained convex problems!

Linear Programs

PDLP

Applegate, Díaz, Hinder, Lu, Lubin, O'Donoghue, Schudy (2021)

NVIDIA cuOPT

Quadratic Programs

OSQP

Stellato, Banjac, Goulart, Bemporad, Boyd (2020)

ProxSuite
THE ADVANCED PROXIMAL OPTIMIZATION TOOLBOX

Bambade, Schramm, El Kazdadi, Caron, Taylor, Carpentier (2025)

Conic Programs

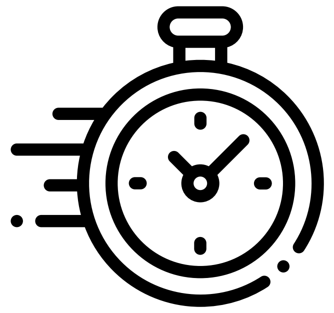
SCS
SPLITTING CONIC SOLVER

O'Donoghue, Chu, Parikh, Boyd (2016)

but they can sometime converge slowly!

major issue in safety-critical applications with

real-time requirements

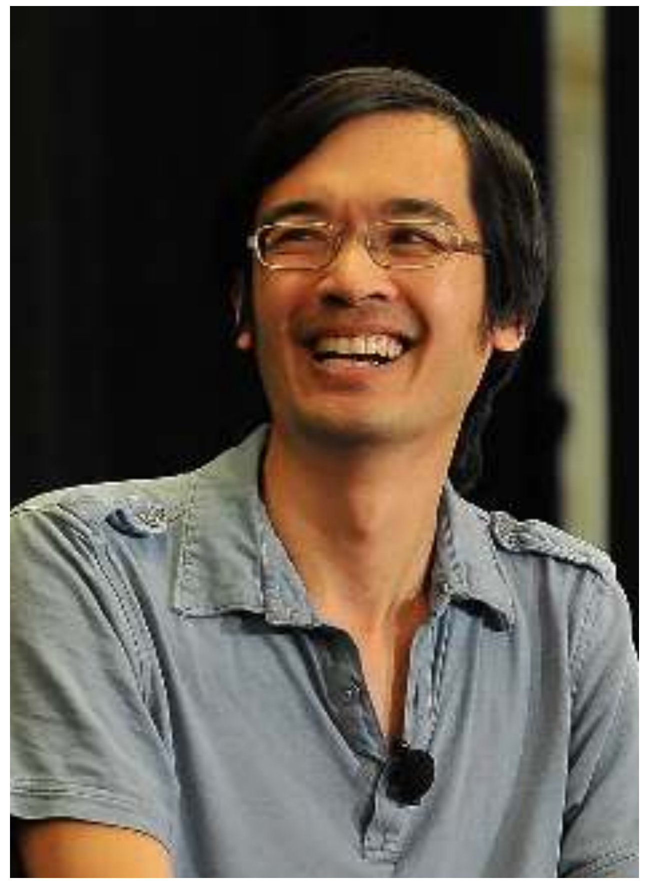


limited computing power

How to analyze first-order methods?

How to analyze first-order methods?

Computers can help us
formalize and verify proofs!



How to analyze first-order methods?

Me:

How can I get performance bounds for first-order methods with the help of a computer?

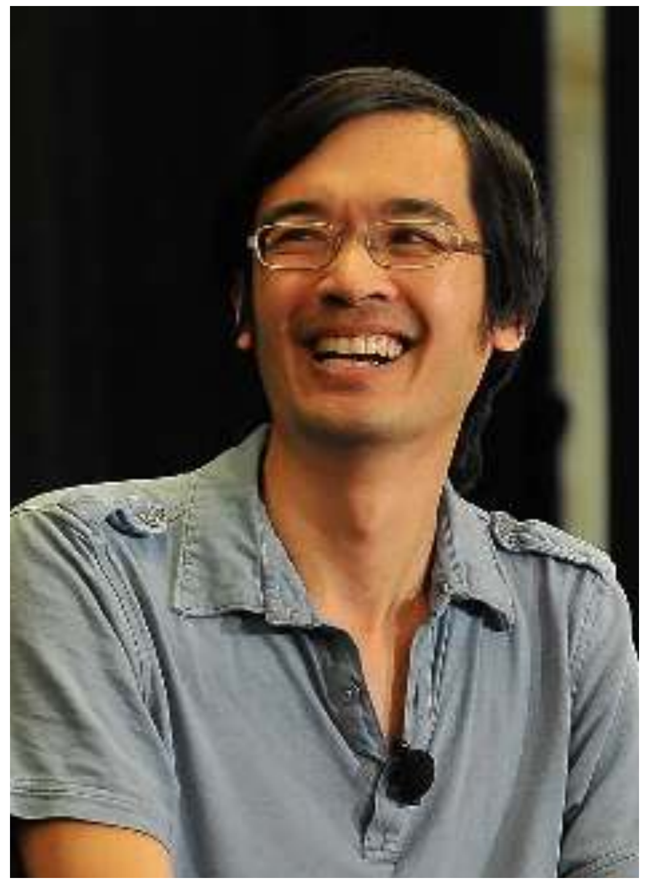
openai/gpt-4.1 Today at 5:10 PM

You can get **performance bounds for first-order methods** (like gradient descent) using **performance estimation problems** (PEPs). These are optimization problems that compute the worst-case performance of your algorithm over a class of functions (e.g., L-smooth convex).

Here's how you can do it with a computer:

1. **Formulate the PEP:**
   - Define the class of functions (e.g., convex, L-smooth).
   - Encode the algorithm (the iteration steps).
   - Specify the performance metric (e.g., function value gap after N steps).
2. **Translate to a Semidefinite Program (SDP):**
   - Typically, you can express the worst-case bound as an SDP.
   - For many standard settings, tools and scripts already exist.
3. **Use Existing Software:**
   - **PEPit (Python):** A user-friendly, actively maintained toolbox for many first-order methods. See: PEPit on GitHub .
   - **CVXPY or YALMIP:** For custom formulations, you can build and solve SDPs using SDP solvers (MOSEK, SCS, etc).

Computers can help us formalize and verify proofs!



How to analyze first-order methods?

Me:

How can I get performance bounds for first-order methods with the help of a computer?

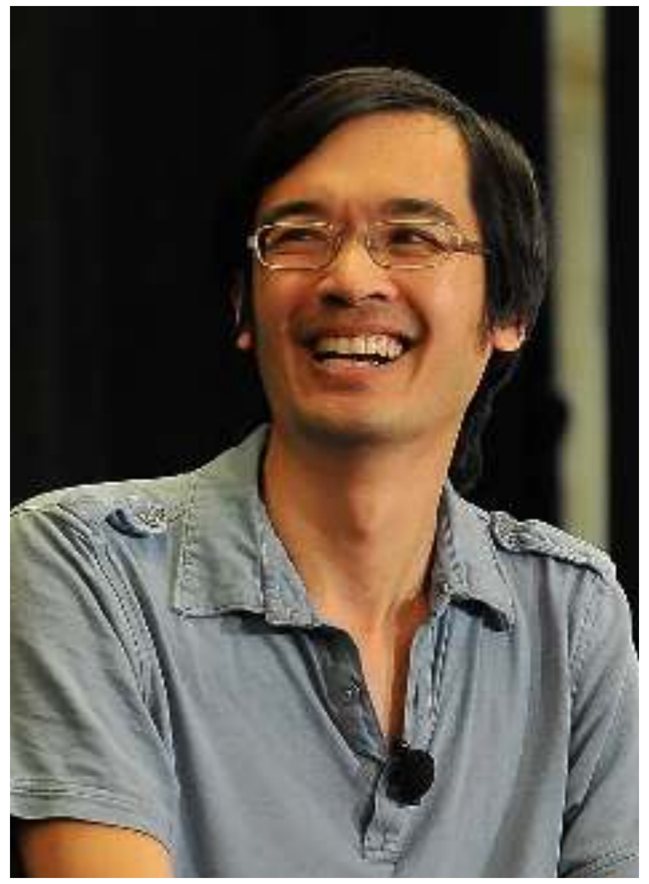
openai/gpt-4.1 Today at 5:10 PM

You can get **performance bounds for first-order methods** (like gradient descent) using **performance estimation problems** (PEPs). These are optimization problems that compute the worst-case performance of your algorithm over a class of functions (e.g., L-smooth convex).

Here's how you can do it with a computer:

1. **Formulate the PEP:**
   - Define the class of functions (e.g., convex, L-smooth).
   - Encode the algorithm (the iteration steps).
   - Specify the performance metric (e.g., function value gap after N steps).
2. **Translate to a Semidefinite Program (SDP):**
   - Typically, you can express the worst-case bound as an SDP.
   - For many standard settings, tools and scripts already exist.
3. **Use Existing Software:**
   - **PEPit (Python):** A user-friendly, actively maintained toolbox for many first-order methods. See: PEPit on GitHub .
   - **CVXPY or YALMIP:** For custom formulations, you can build and solve SDPs using SDP solvers (MOSEK, SCS, etc).

Computers can help us formalize and verify proofs!



Best known bounds via computer assisted analysis

problem

minimize $\quad f(z)$

algorithm

$$z^{k+1} = T(z^k)$$

Best known bounds via computer assisted analysis

problem

minimize $\quad f(z)$

algorithm

$$z^{k+1} = T(z^k)$$

→

Performance Estimation Problem

$$\begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\| \\ \text{subject to} & f \in \mathcal{F} \\ & z^0, z^1, \dots, z^K \quad \text{iterates of algorithm } T \text{ on } f \\ & \|z^1 - z^0\| \le 1 \end{array}$$

Best known bounds via computer assisted analysis

problem

minimize $\quad f(z)$

algorithm

$$z^{k+1} = T(z^k)$$

Performance Estimation Problem

$$\begin{array}{ll} \text{maximize} & \boxed{\|z^K - z^{K-1}\|} \\ \text{subject to} & f \in \mathcal{F} \\ & z^0, z^1, \ldots, z^K \quad \text{iterates of algorithm } T \text{ on } f \\ & \|z^1 - z^0\| \leq 1 \end{array}$$

performance metric

Best known bounds via computer assisted analysis

problem

minimize $\quad f(z)$

algorithm

$$z^{k+1} = T(z^k)$$

Performance Estimation Problem

$$\begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\| \\ \text{subject to} & f \in \mathcal{F} \\ & z^0, z^1, \ldots, z^K \quad \text{iterates of algorithm } T \text{ on } f \\ & \|z^1 - z^0\| \le 1 \end{array}$$

performance metric

function class
(e.g., strongly-convex, smooth functions)

Best known bounds via computer assisted analysis

problem

minimize $\quad f(z)$

algorithm

$$z^{k+1} = T(z^k)$$

Performance Estimation Problem

$$
\begin{array}{ll}
\text{maximize} & \|z^K - z^{K-1}\| \\
\text{subject to} & f \in \mathcal{F} \\
 & z^0, z^1, \dots, z^K \quad \text{iterates of algorithm } T \text{ on } f \\
 & \|z^1 - z^0\| \leq 1
\end{array}
$$

performance metric

function class
(e.g., strongly-convex, smooth functions)

initial condition

Best known bounds via computer assisted analysis

problem

minimize $f(z)$

algorithm

$z^{k+1} = T(z^k)$

Performance Estimation Problem

$$\begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\| \\ \text{subject to} & f \in \mathcal{F} \\ & z^0, z^1, \ldots, z^K \quad \text{iterates of algorithm } T \text{ on } f \\ & \|z^1 - z^0\| \le 1 \end{array}$$

performance metric

function class
(e.g., strongly-convex, smooth functions)

initial condition

⚠️
infinite dimensional problem

Best known bounds via computer assisted analysis

problem

minimize $\quad f(z)$

algorithm

$$z^{k+1} = T(z^k)$$

Performance Estimation Problem

$$
\begin{array}{ll}
\text{maximize} & \|z^K - z^{K-1}\| \\
\text{subject to} & f \in \mathcal{F} \\
& z^0, z^1, \dots, z^K \quad \text{iterates of algorithm } T \text{ on } f \\
& \|z^1 - z^0\| \leq 1
\end{array}
$$

performance metric

function class *(e.g., strongly-convex, smooth functions)*

initial condition

⚠️ **infinite dimensional problem**

interpolation conditions

$$\exists f \in \mathcal{F} \text{ such that } \begin{cases} f^i = f(x^i) \\ g^i = \nabla f(x^i) \end{cases}$$

Drori and Teboulle (2014), Taylor, Hendrickx, Gilneur (2017)

Best known bounds via computer assisted analysis

problem

minimize $f(z)$

algorithm

$z^{k+1} = T(z^k)$

Performance Estimation Problem

$$\begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\| \\ \text{subject to} & f \in \mathcal{F} \\ & z^0, z^1, \dots, z^K \quad \text{iterates of algorithm } T \text{ on } f \\ & \|z^1 - z^0\| \leq 1 \end{array}$$

performance metric

function class
(e.g., strongly-convex, smooth functions)

initial condition

⚠️
infinite dimensional problem

interpolation conditions

$$\exists f \in \mathcal{F} \text{ such that } \begin{cases} f^i = f(x^i) \\ g^i = \nabla f(x^i) \end{cases}$$

example

convex functions $\mathcal{F}_{0,\infty}$

$$f^j \geq f^i + (g^i)^T (z^j - z^i)$$

Drori and Teboulle (2014), Taylor, Hendrickx, Gilneur (2017)

PEP SDP formulation

still a nonconvex QCQP

$$f^j \geq f^i + (g^i)^T(z^j - z^i)$$

$$\|z^1 - z^0\| \leq 1 \quad \iff \quad (z^1 - z^0)^T(z^1 - z^0) \leq 1$$

PEP SDP formulation

still a nonconvex QCQP

convex SDP
Gram matrix reformulation

$$f^j \geq f^i + (g^i)^T(z^j - z^i)$$

$$\|z^1 - z^0\| \leq 1 \iff (z^1 - z^0)^T(z^1 - z^0) \leq 1$$

$$G = P^T P \quad \text{with} \quad P = \begin{bmatrix} | & & | & | & & | \\ z^0 & \dots & z^K & g^0 & \dots & g^K \\ | & & | & | & & | \end{bmatrix}$$

$$F = (f^0, \dots, f^K)$$

PEP SDP formulation

still a nonconvex QCQP

convex SDP
Gram matrix reformulation

$$f^j \geq f^i + (g^i)^T(z^j - z^i)$$

$$\|z^1 - z^0\| \leq 1 \quad \Longleftrightarrow \quad (z^1 - z^0)^T(z^1 - z^0) \leq 1$$

$$G = P^T P \quad \text{with} \quad P = \begin{bmatrix} | & & | & | & & | \\ z^0 & \dots & z^K & g^0 & \dots & g^K \\ | & & | & | & & | \end{bmatrix}$$

$$F = (f^0, \dots, f^K)$$

Features

- Tight SDP formulation
- Proofs via dual problem
- Useful for algorithm design

PEP SDP formulation

still a nonconvex QCQP

convex SDP
Gram matrix reformulation

$$f^j \geq f^i + (g^i)^T(z^j - z^i)$$

$$\|z^1 - z^0\| \leq 1 \iff (z^1 - z^0)^T(z^1 - z^0) \leq 1$$

$$G = P^T P \quad \text{with} \quad P = \begin{bmatrix} | & & | & | & & | \\ z^0 & \dots & z^K & g^0 & \dots & g^K \\ | & & | & | & & | \end{bmatrix}$$

$$F = (f^0, \dots, f^K)$$

Features

- Tight SDP formulation
- Proofs via dual problem
- Useful for algorithm design

Several papers on the topic

early methodological

Drori and Teboulle (2014), Taylor, Hendrickx, Gilneur (2017), Lesssard, Recht, Packard (2016)

more recent works

Taylor and Bach (2019) Ryu, Taylor, Bergeling, Giselsson (2020) Park and Ryu (2022), Das Gupta, Freund, Sun, Taylor (2024) Luner, Grimmer (2024), Upadhyaya, Banert, Taylor, Giselsson (2025)…

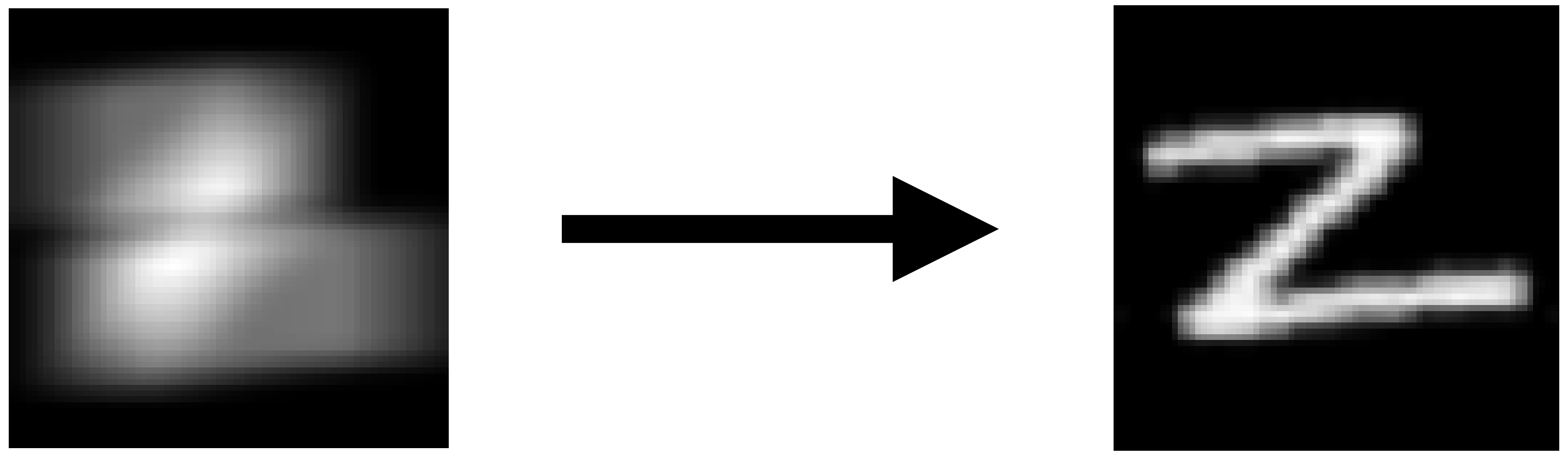
algorithm design

Kim and Fessler (2016) Jang, Das Gupta, Ryu (2023) Das Gupta, Van Parys, Ryu (2024)…

We still do not fully understand their convergence!

image deblurring problem

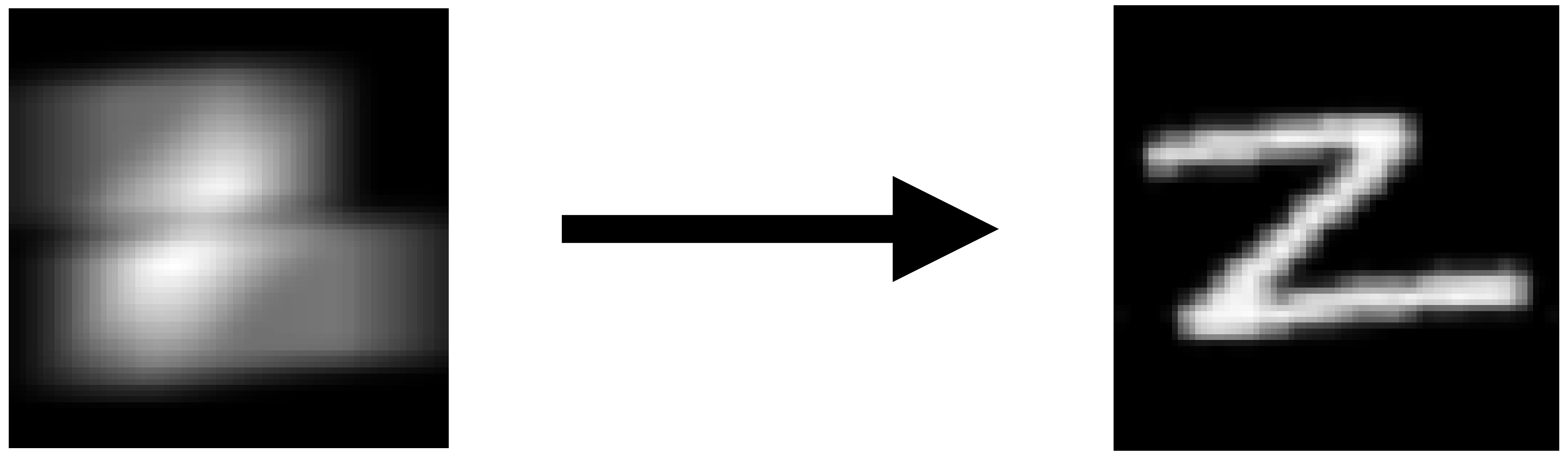
emnist dataset



Theoretical bounds from Performance Estimation Problem (PEP)

We still do not fully understand their convergence!

image deblurring problem
emnist dataset



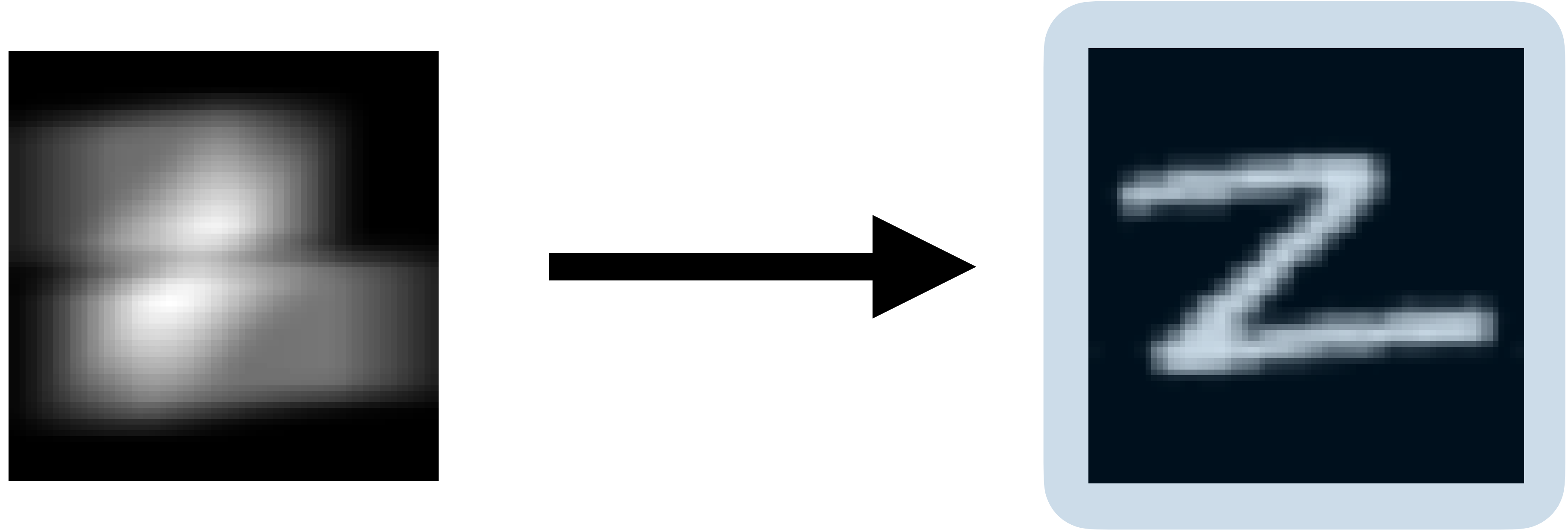
minimize $\quad \|Az - x\|_2^2 + \lambda \|z\|_1$

subject to $\quad 0 \leq z \leq 1$

Theoretical bounds from Performance Estimation Problem (PEP)

We still do not fully understand their convergence!

image deblurring problem
emnist dataset



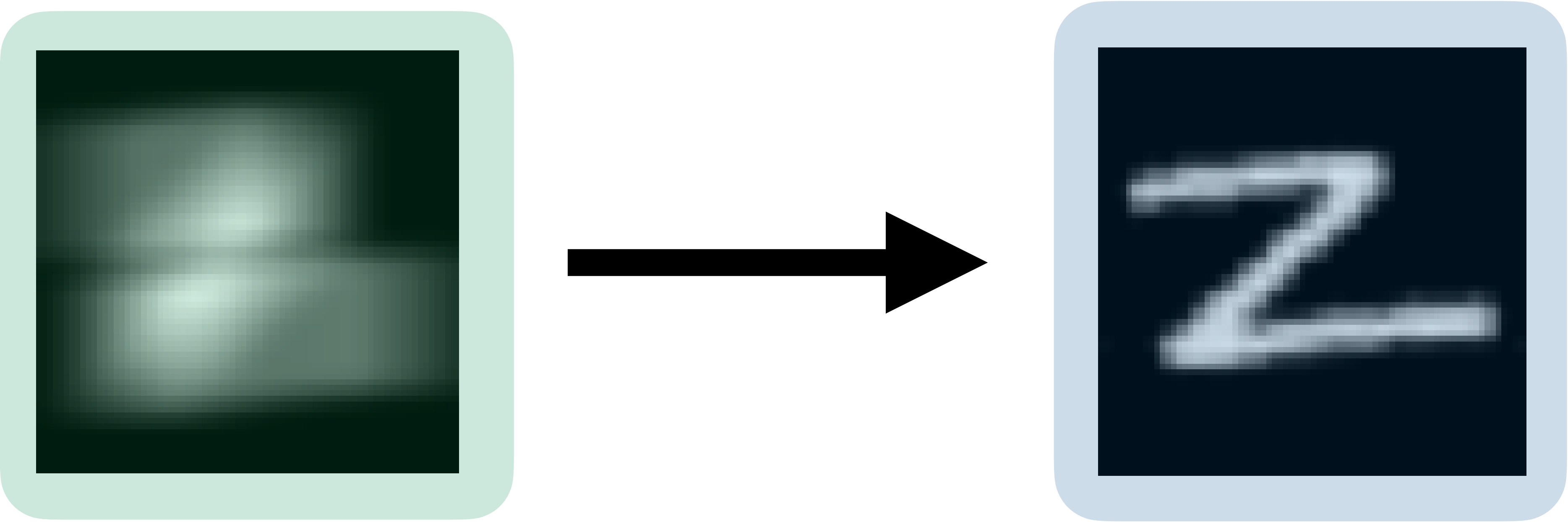
$$\begin{array}{ll} \text{minimize} & \|Az - x\|_2^2 + \lambda\|z\|_1 \\ \text{subject to} & 0 \leq z \leq 1 \end{array}$$

deblurred image

Theoretical bounds from Performance Estimation Problem (PEP)

We still do not fully understand their convergence!

image deblurring problem
emnist dataset



$$\begin{array}{ll} \text{minimize} & \|Az - x\|_2^2 + \lambda\|z\|_1 \\ \text{subject to} & 0 \leq z \leq 1 \end{array}$$

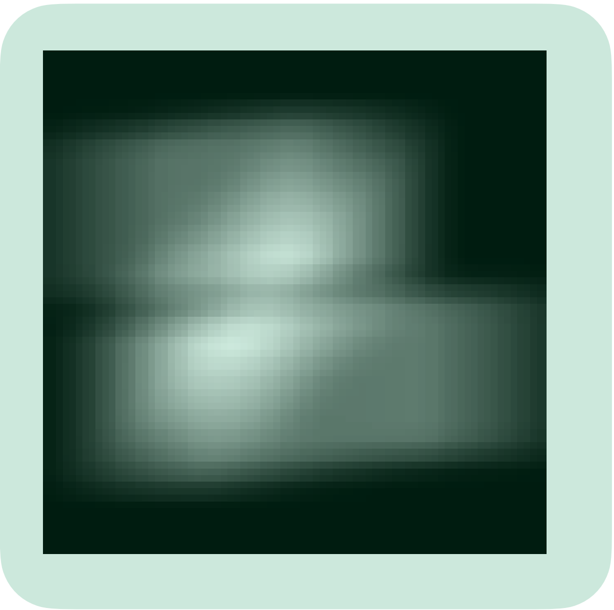
blurred image

deblurred image

Theoretical bounds from Performance Estimation Problem (PEP)

We still do not fully understand their convergence!

image deblurring problem
emnist dataset

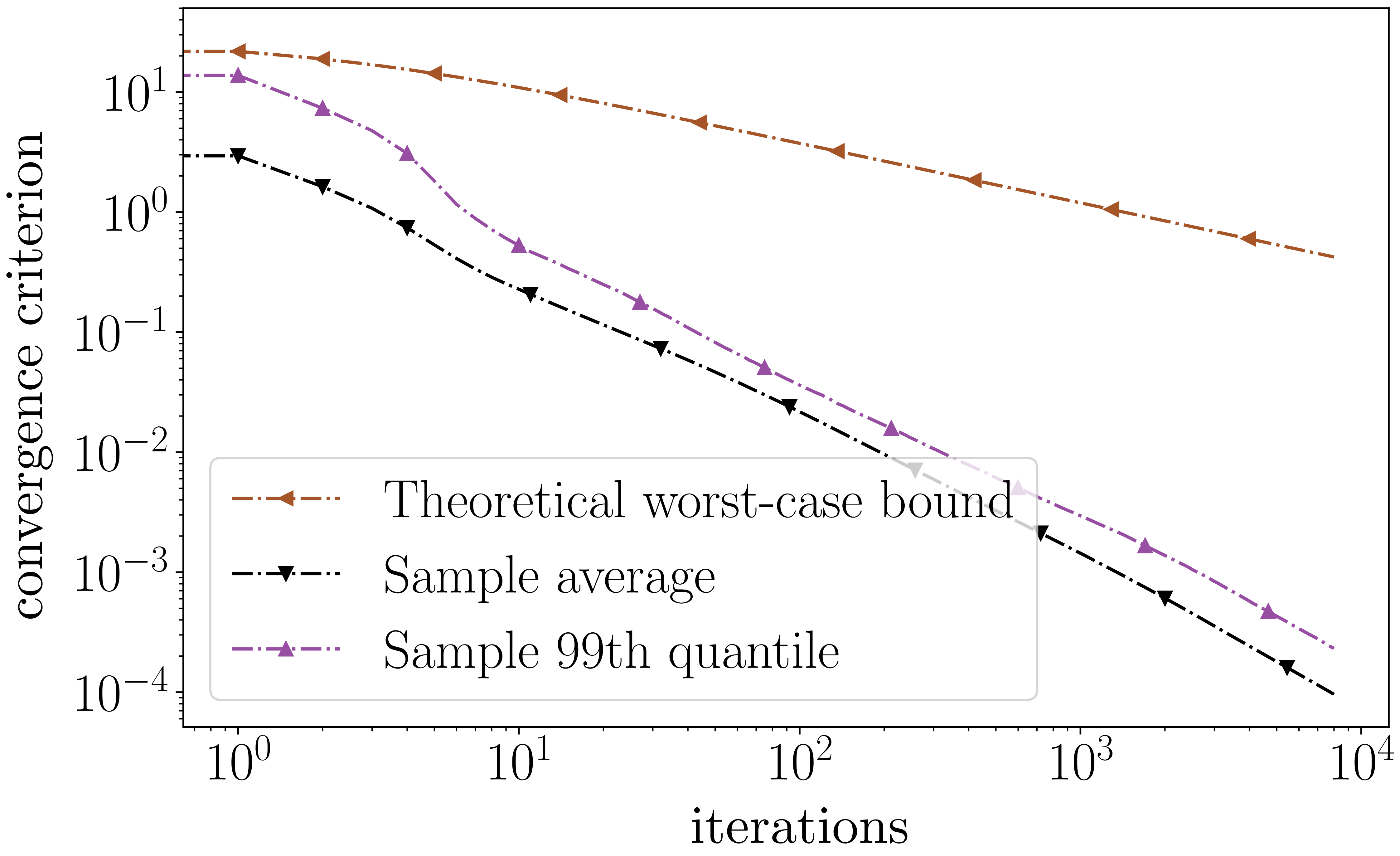


minimize $\quad \|Az - x\|_2^2 + \lambda\|z\|_1$

subject to $\quad 0 \leq z \leq 1$

blurred image

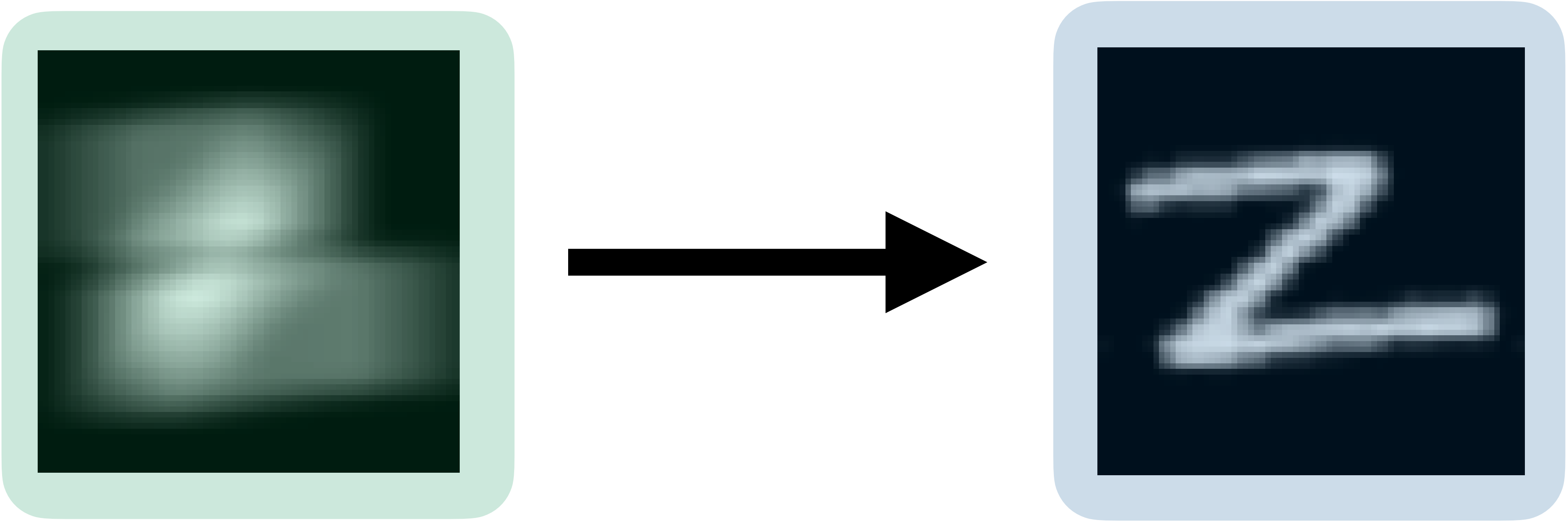
deblurred image



Theoretical bounds from Performance Estimation Problem (PEP)

We still do not fully understand their convergence!

image deblurring problem
emnist dataset

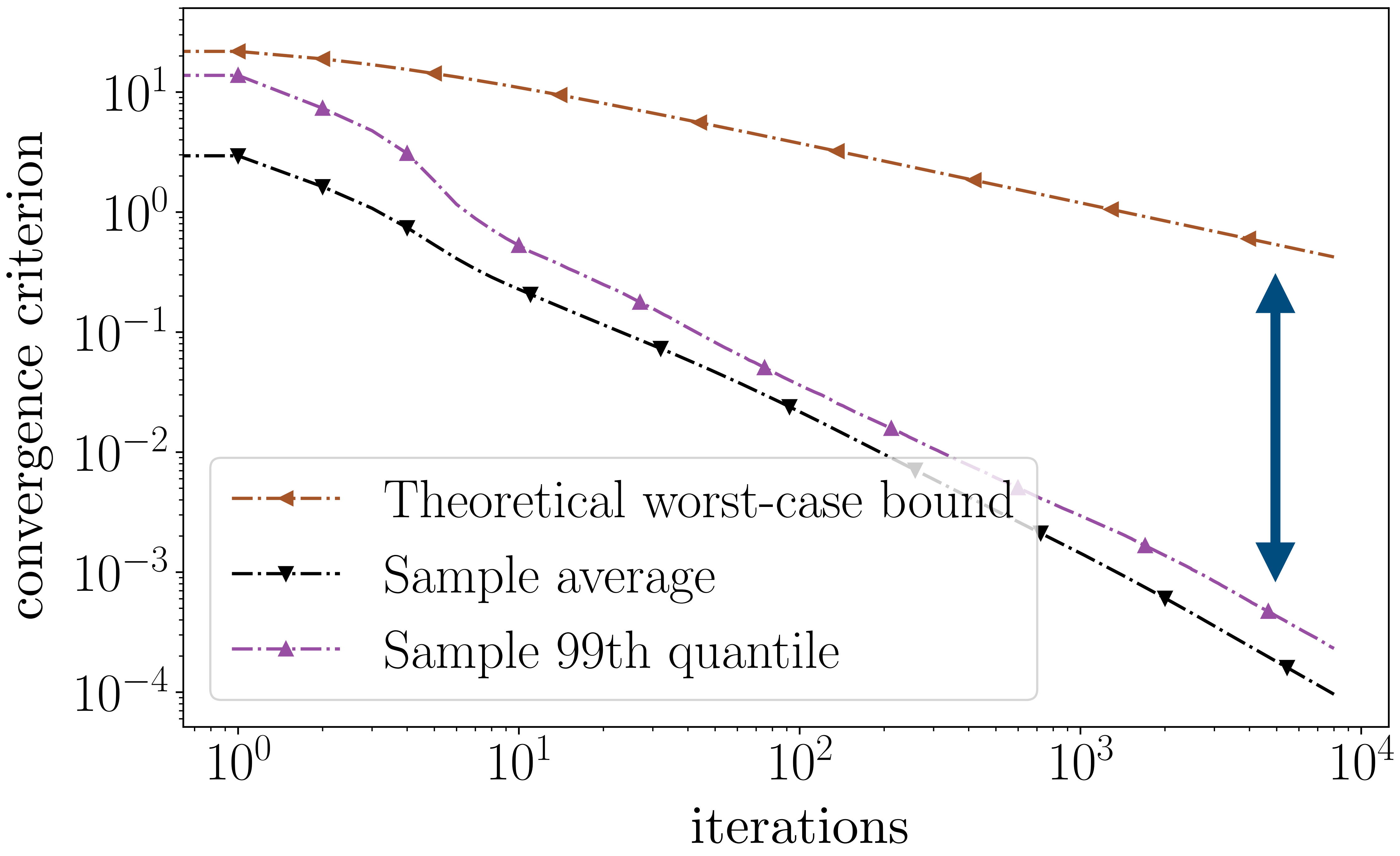


minimize $\quad \|Az - x\|_2^2 + \lambda \|z\|_1$

subject to $\quad 0 \leq z \leq 1$

blurred image

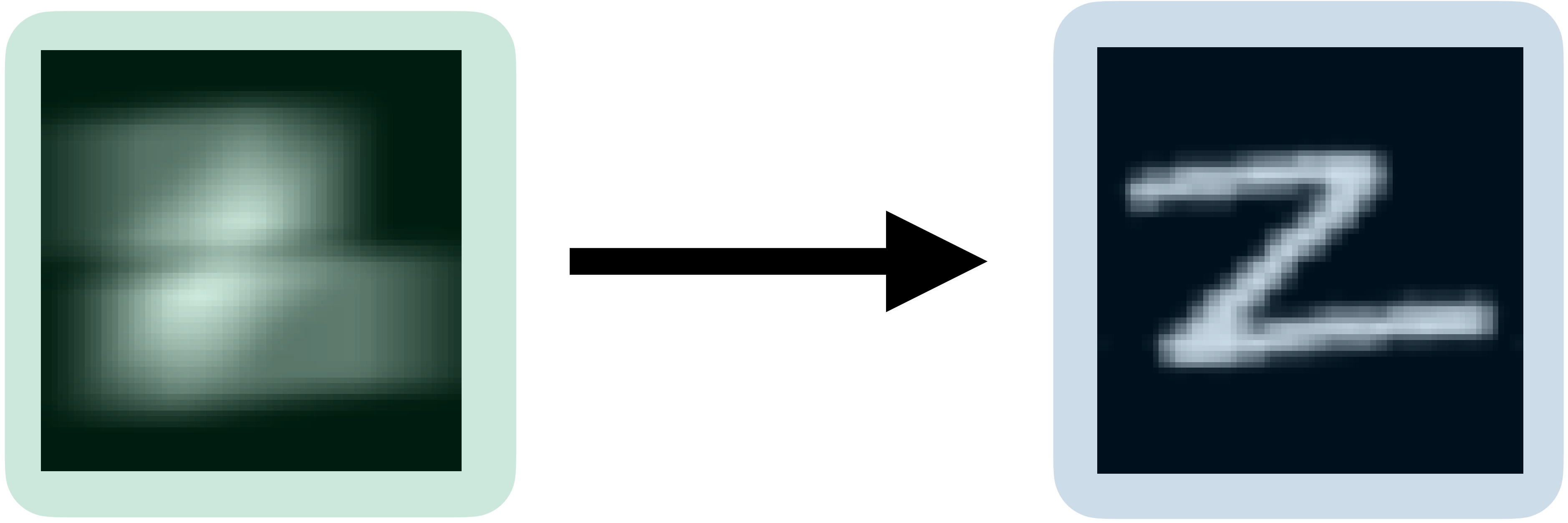
deblurred image



Theoretical bounds from Performance Estimation Problem (PEP)

We still do not fully understand their convergence!

image deblurring problem
emnist dataset

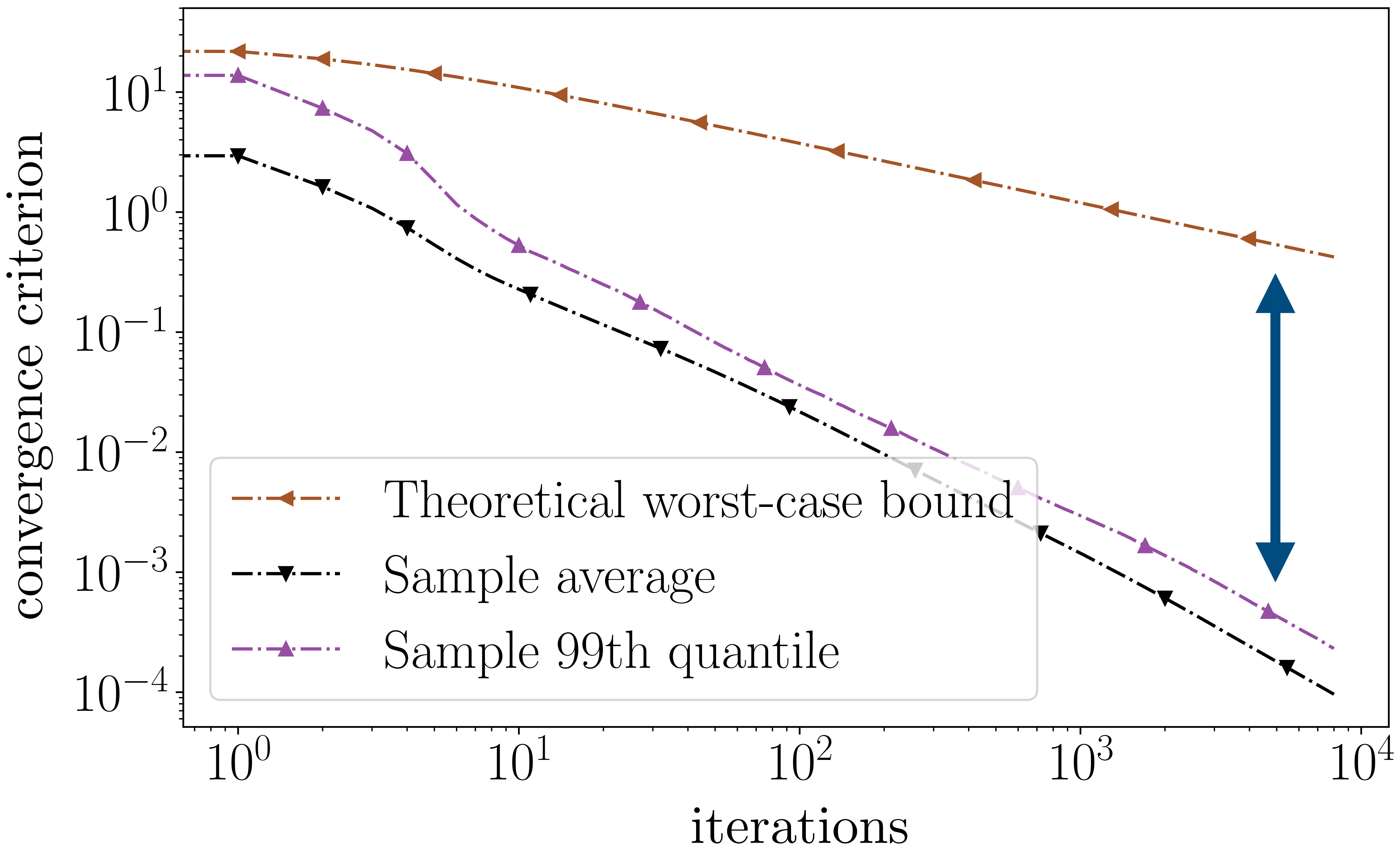


minimize $\|Az - x\|_2^2 + \lambda \|z\|_1$

subject to $0 \leq z \leq 1$

blurred image

deblurred image

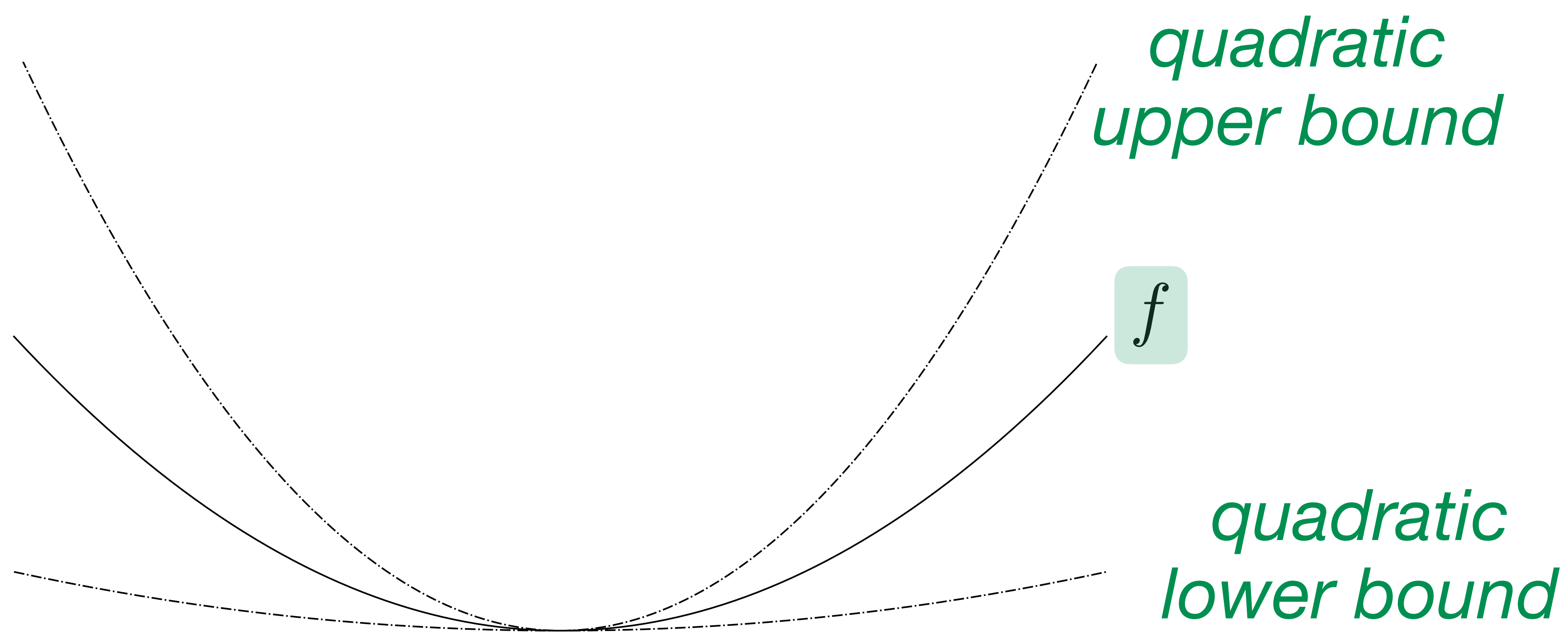


Why are the bounds so pessimistic?

Theoretical bounds from Performance Estimation Problem (PEP)

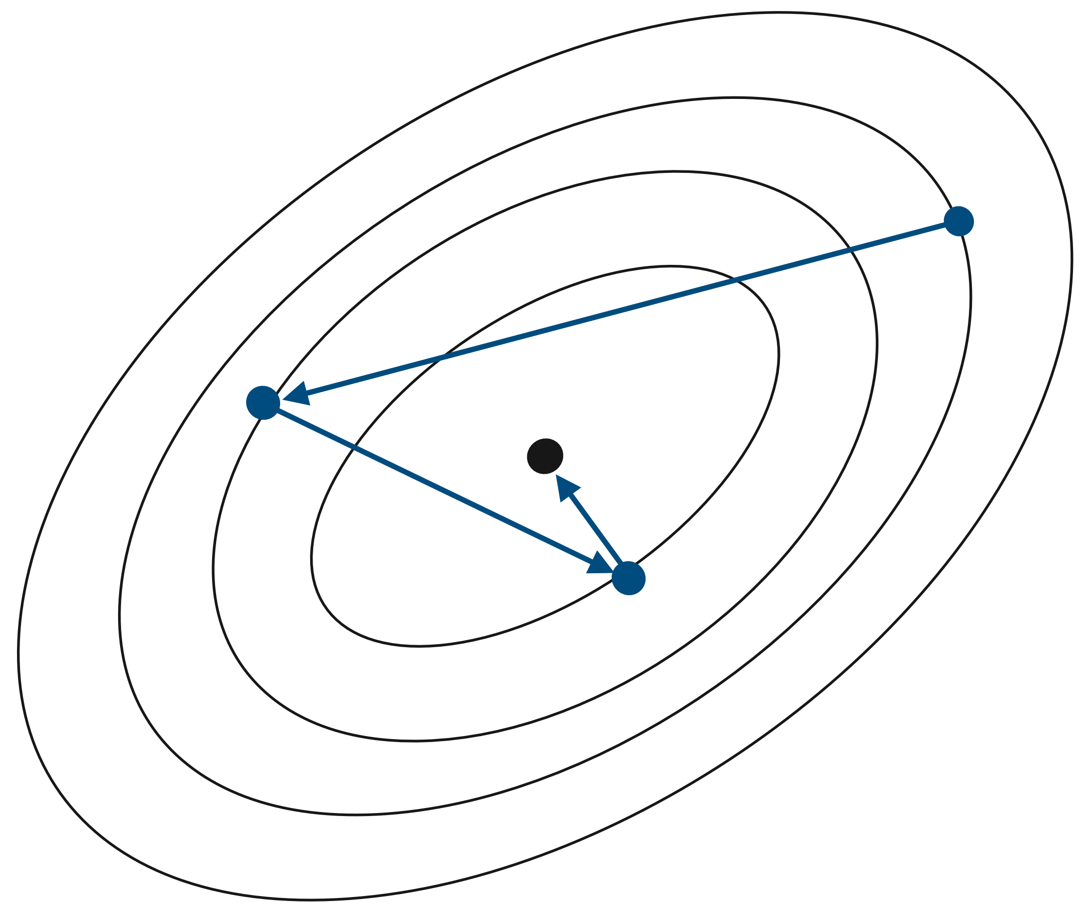
sources of pessimism in classical convergence analysis

general function classes
(f is strongly convex and smooth...)



we may never encounter that function

***a priori* analysis**
never runs the algorithm



running the algorithm may highlight key structural properties

💡 main idea

in most applications we repeatedly solve the same problem with varying parameters

$$\begin{array}{ll} \text{minimize} & f(z, x) \\ \text{subject to} & z \in C(x) \end{array}$$

💡 main idea

in most applications we repeatedly solve the same problem with varying parameters

$$\begin{array}{ll} \text{minimize} & f(z, x) \\ \text{subject to} & z \in C(x) \end{array}$$

structured problems
(e.g., parameters -> solutions)

large amount of data
(e.g., instances, iterate trajectories)

We will model first-order methods with operators

iterations

$$z^{k+1} = \boxed{T}(z^k, x) \quad \text{for } k = 0, 1, \dots$$

operator
(e.g., contractive, averaged)

goal: find fixed-points

$$\boxed{z^\star} = T(\boxed{z^\star}, x)$$

We will model first-order methods with operators

iterations

$$z^{k+1} = \boxed{T}(z^k, x) \quad \text{for } k = 0, 1, \ldots$$

operator

(e.g., contractive, averaged)

goal: find fixed-points

$$\boxed{z^\star} = T(\boxed{z^\star}, x)$$

example
gradient descent

problem

minimize $\quad f(z, x) \longrightarrow$

optimality conditions

$$\nabla f(z^\star, x) = 0$$

iterations

$$z^{k+1} = z^k - \theta \nabla f(z^k, x)$$

fixed-points

$$z^\star = z^\star - \theta \nabla f(z^\star, x)$$

same as

We will model first-order methods with operators

iterations

$$z^{k+1} = T(z^k, x) \quad \text{for } k = 0, 1, \ldots$$

operator

(e.g., contractive, averaged)

goal: find fixed-points

$$z^\star = T(z^\star, x)$$

example
gradient descent

problem

minimize $\quad f(z, x)$ $\longrightarrow$

optimality conditions

$$\nabla f(z^\star, x) = 0$$

iterations

$$z^{k+1} = z^k - \theta \nabla f(z^k, x)$$

fixed-points

$$z^\star = z^\star - \theta \nabla f(z^\star, x)$$

same as

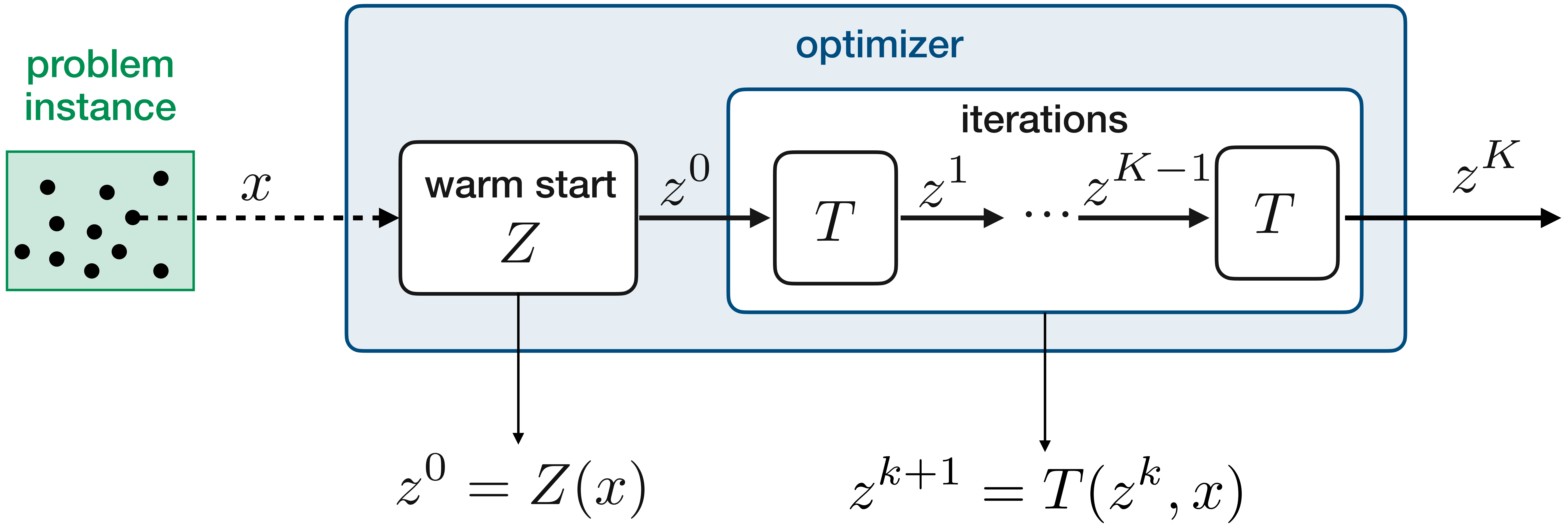
performance metric

$$r^k(x) = \|T(z^{k-1}) - z^{k-1}\| = \|z^k - z^{k-1}\|$$

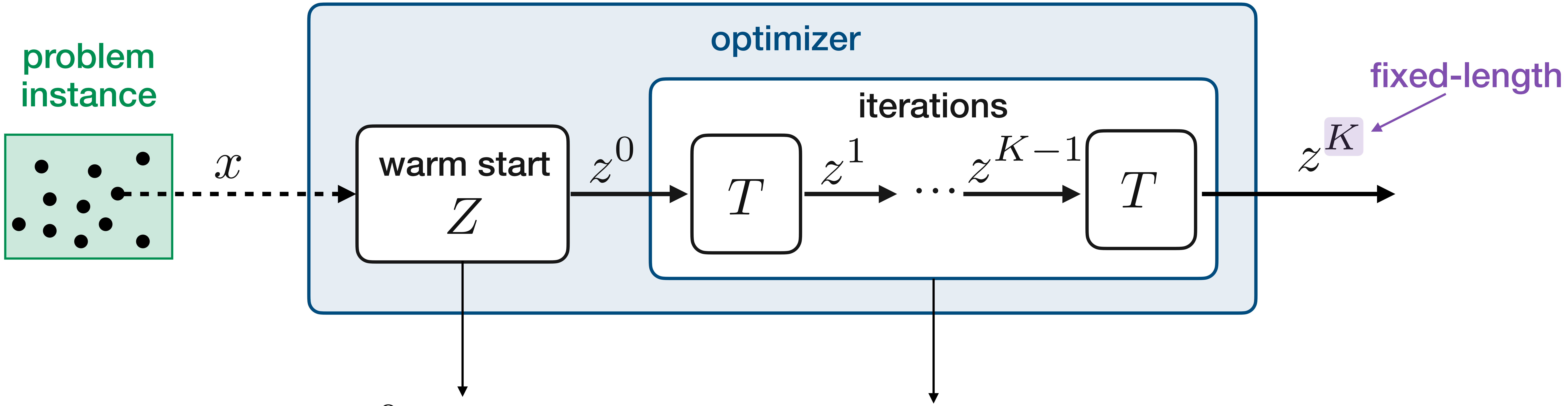
fixed-point residual

(under suitable assumptions it converges to 0)

Algorithms as fixed-length computational graphs



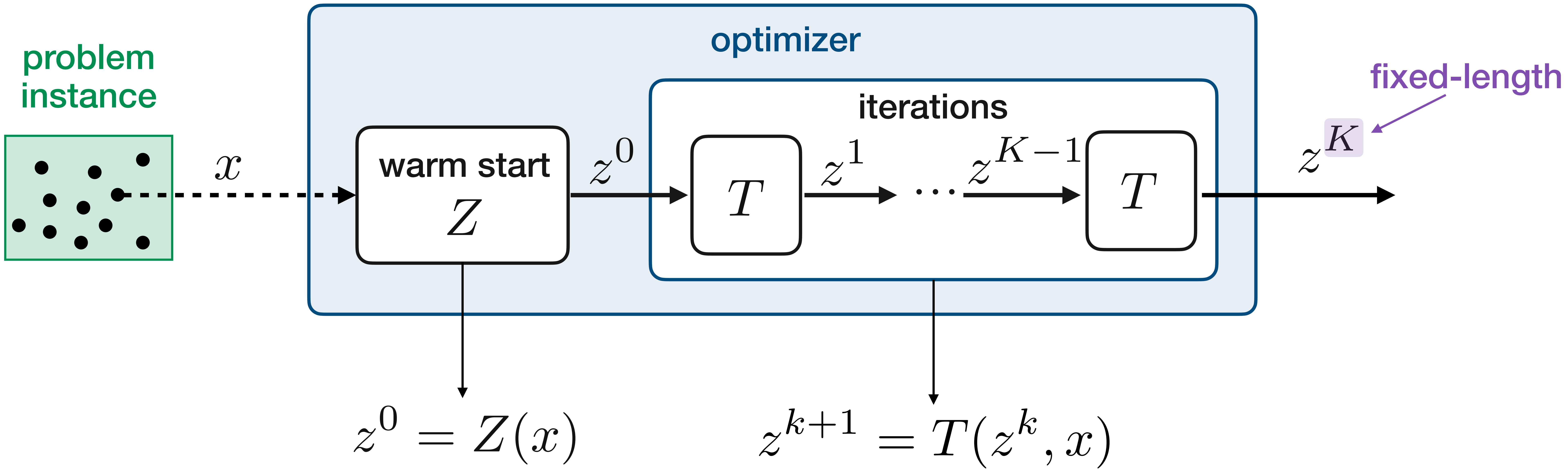
Algorithms as fixed-length computational graphs



$$z^0 = Z(x)$$

$$z^{k+1} = T(z^k, x)$$

Algorithms as fixed-length computational graphs



example

projected gradient descent

$$z^{k+1} = \Pi_{C(x)}(z^k - \theta \nabla_z f(z^k, x))$$

Analyzing the algorithm performance after K iterations

goal

estimate norm of fixed-point residual

$$r^K(x) = \|z^K - z^{K-1}\|$$

Analyzing the algorithm performance after K iterations

goal

estimate norm of fixed-point residual

$$r^K(x) = \|z^K - z^{K-1}\|$$

worst-case

$$\max_{\boxed{x \in \mathcal{X}}} r^K(x) \leq \boxed{\epsilon}$$

problem instances

convergence tolerance

Analyzing the algorithm performance after K iterations

goal

estimate norm of fixed-point residual

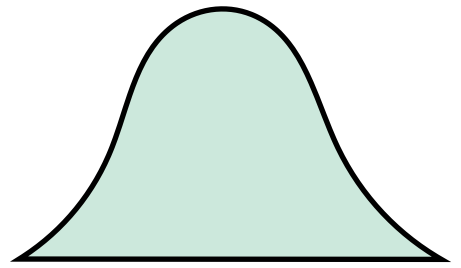
$$r^K(x) = \|z^K - z^{K-1}\|$$

worst-case

$$\max_{x \in \mathcal{X}} r^K(x) \leq \epsilon$$

problem instances

convergence tolerance



probabilistic

$$\mathbf{P}\left(r^K(x) > \epsilon\right) \leq \eta$$

probability bound

problem instances

convergence tolerance

Analyzing the algorithm performance after K iterations

goal

estimate norm of fixed-point residual

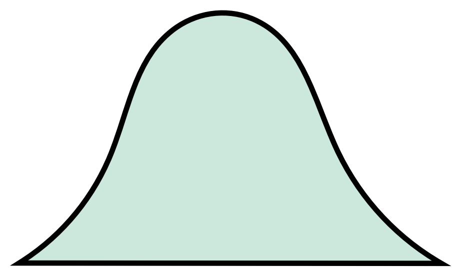
$$r^K(x) = \|z^K - z^{K-1}\|$$

worst-case

$$\max_{x \in \mathcal{X}} r^K(x) \leq \epsilon$$

problem instances

convergence tolerance



probabilistic

$$\mathbf{P}\left(r^K(x) > \epsilon\right) \leq \eta$$

probability bound

problem instances

convergence tolerance

Worst-case analysis

Worst-case algorithm verification for parametric quadratic optimization

$$\begin{array}{ll} \text{minimize} & (1/2)z^T P z + q(x)^T z \\ \text{subject to} & Az \leq b(x) \end{array}$$

Worst-case algorithm verification for parametric quadratic optimization

$$\begin{array}{ll} \text{minimize} & (1/2)z^T P z + q(x)^T z \\ \text{subject to} & A\underbrace{z}_{\text{decisions}} \leq b(x) \end{array}$$

Worst-case algorithm verification for parametric quadratic optimization

positive semidefinite

$$
\begin{array}{ll}
\text{minimize} & (1/2)z^T P z + q(x)^T z \\
\text{subject to} & Az \le b(x)
\end{array}
$$

decisions

Worst-case algorithm verification for parametric quadratic optimization

$$\begin{array}{ll} \text{minimize} & (1/2)z^T \overset{\substack{\text{positive} \\ \text{semidefinite} \\ \downarrow}}{P} z + q(x)^T z \\ \text{subject to} & A\underset{\substack{\uparrow \\ \text{decisions}}}{z} \leq b(x) \end{array}$$

problem instances

Worst-case algorithm verification for parametric quadratic optimization

positive semidefinite

$$\begin{array}{ll} \text{minimize} & (1/2)z^T P z + q(x)^T z \\ \text{subject to} & Az \leq b(x) \end{array}$$

decisions

problem instances

solve with your favorite first-order method

$$z^{k+1} = T(z^k, x)$$

Worst-case algorithm verification for parametric quadratic optimization

positive semidefinite

minimize	$(1/2)z^T P z + q(x)^T z$
subject to	$Az \leq b(x)$

decisions

problem instances

solve with your favorite first-order method

$$z^{k+1} = T(z^k, x)$$

verification problem

$$\max_{x \in \mathcal{X}} r^K(x) = \begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\| \\ \text{subject to} & z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1 \\ & z^0 = Z(x), \quad x \in \mathcal{X} \end{array}$$

Worst-case algorithm verification for parametric quadratic optimization

positive semidefinite

$$\begin{array}{ll} \text{minimize} & (1/2)z^T P z + q(x)^T z \\ \text{subject to} & Az \leq b(x) \end{array}$$

decisions

problem instances

solve with your favorite first-order method

$$z^{k+1} = T(z^k, x)$$

verification problem

$$\max_{x \in \mathcal{X}} r^K(x) = \begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\| \\ \text{subject to} & z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1 \\ & z^0 = Z(x), \quad x \in \mathcal{X} \end{array}$$

problem instances

Worst-case algorithm verification for parametric quadratic optimization

positive semidefinite

$$\begin{array}{ll} \text{minimize} & (1/2)z^T P z + q(x)^T z \\ \text{subject to} & Az \le b(x) \end{array}$$

decisions

problem instances

solve with your favorite first-order method

$$z^{k+1} = T(z^k, x)$$

verification problem

performance metric

$$\max_{x \in \mathcal{X}} r^K(x) = \begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\| \\ \text{subject to} & z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1 \\ & z^0 = Z(x), \quad x \in \mathcal{X} \end{array}$$

problem instances

Worst-case algorithm verification for parametric quadratic optimization

positive semidefinite

$$\begin{array}{ll} \text{minimize} & (1/2) z^T P z + q(x)^T z \\ \text{subject to} & Az \leq b(x) \end{array}$$

decisions

problem instances

solve with your favorite first-order method

$$z^{k+1} = T(z^k, x)$$

verification problem

performance metric

$$\max_{x \in \mathcal{X}} r^K(x) = \begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\| \\ \text{subject to} & z^{k+1} = T(z^k, x), \quad k = 0, \ldots, K-1 \\ & z^0 = Z(x), \quad x \in \mathcal{X} \end{array}$$

problem instances

NP-hard problem!

reduction from
0-1 integer programming

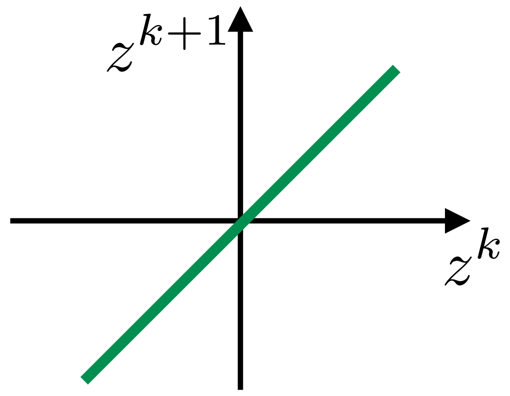
Verification of First-Order Methods for Parametric Quadratic Optimization
V. Ranjan and B. Stellato
arXiv e-prints:2403.03331 (2025)
github.com/stellatogrp/sdp_algo_verify

Two types of algorithm steps

Linear steps → Linear constraints

e.g., gradient, momentum, restarts, anchors, prox of quadratic functions…

$$Mz^{k+1} = Az^k + Bx$$

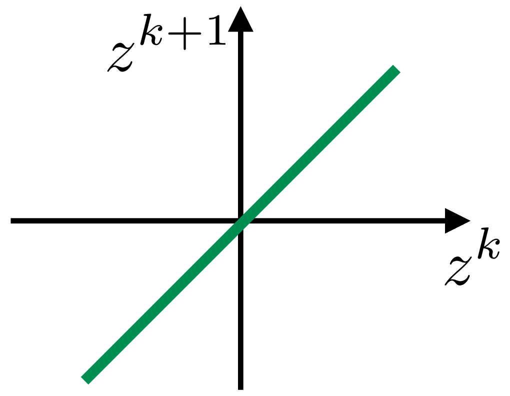


Two types of algorithm steps

Linear steps → Linear constraints

e.g., gradient, momentum, restarts, anchors, prox of quadratic functions…

$$Mz^{k+1} = Az^k + Bx$$

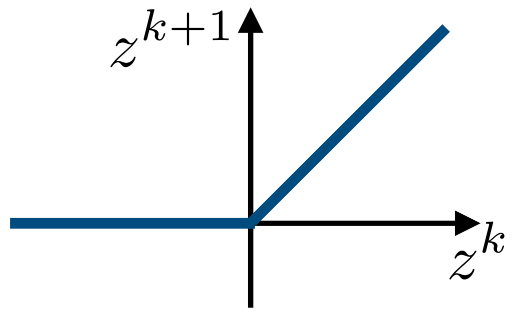


Piecewise affine steps → Bilinear constraints

Elementwise maximum (ReLU)

e.g., one-sided projections

$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$

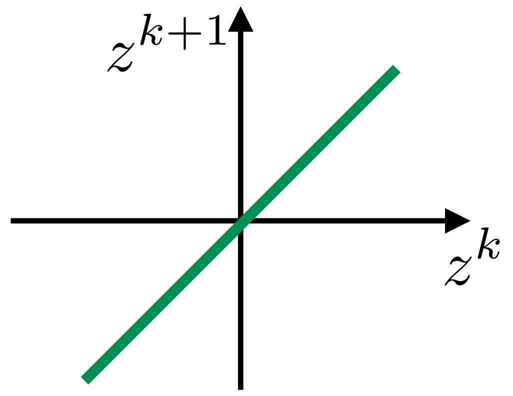


Two types of algorithm steps

Linear steps → Linear constraints

e.g., gradient, momentum, restarts, anchors, prox of quadratic functions…

$$Mz^{k+1} = Az^k + Bx$$

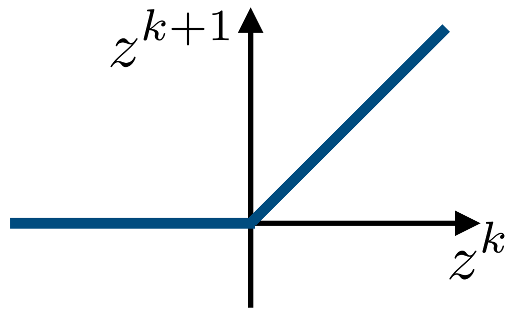


Piecewise affine steps → Bilinear constraints

Elementwise maximum (ReLU)

e.g., one-sided projections

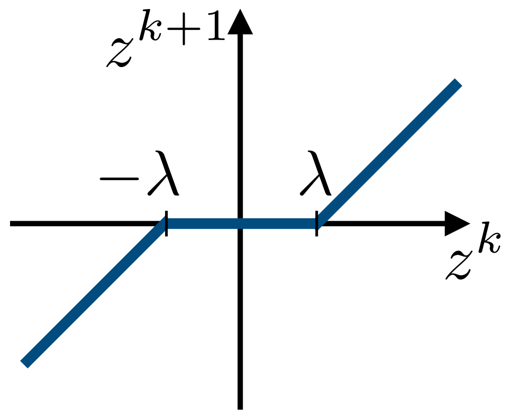
$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$



Soft-thresholding

e.g., prox of 1-norm function

$$z^{k+1} = \phi_\lambda(z^k) = \max\{z^k, \lambda\} - \max\{-z^k, \lambda\}$$

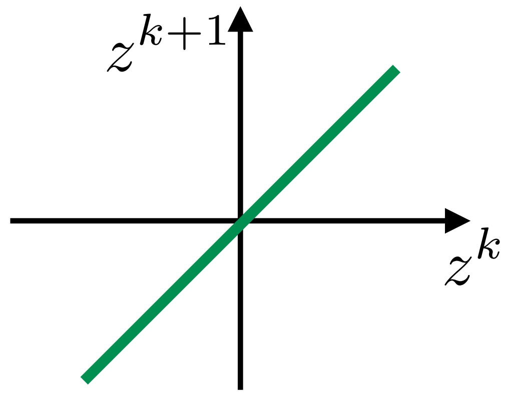


Two types of algorithm steps

Linear steps → Linear constraints

e.g., gradient, momentum, restarts, anchors, prox of quadratic functions…

$$Mz^{k+1} = Az^k + Bx$$

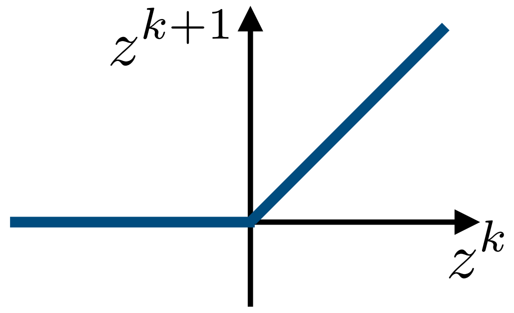


Piecewise affine steps → Bilinear constraints

Elementwise maximum (ReLU)

e.g., one-sided projections

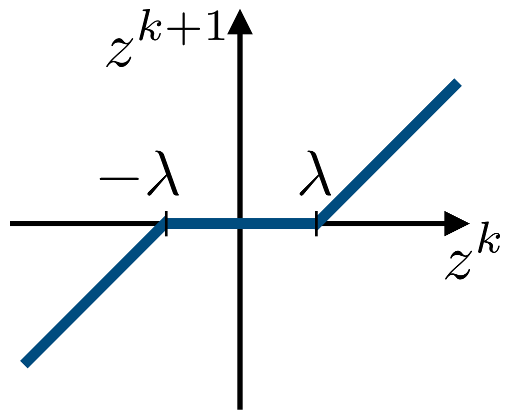
$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$



Soft-thresholding

e.g., prox of 1-norm function

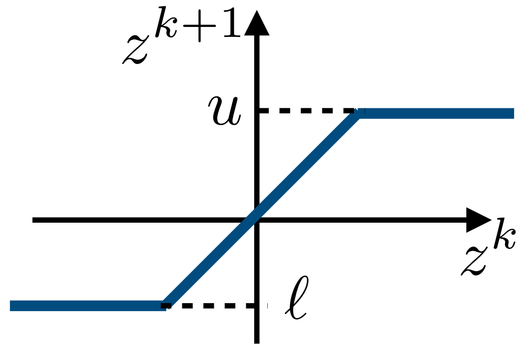
$$z^{k+1} = \phi_\lambda(z^k) = \max\{z^k, \lambda\} - \max\{-z^k, \lambda\}$$



Saturated linear unit (SatLin)

e.g., box projections

$$z^{k+1} = \mathcal{S}_{[\ell,u]}(z^k) = \min\{\max\{z^k, \ell\}, u\}$$

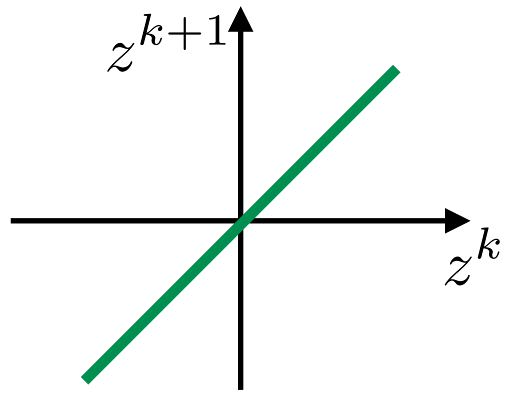


Two types of algorithm steps

Linear steps → Linear constraints

e.g., gradient, momentum, restarts, anchors, prox of quadratic functions…

$$Mz^{k+1} = Az^k + Bx$$

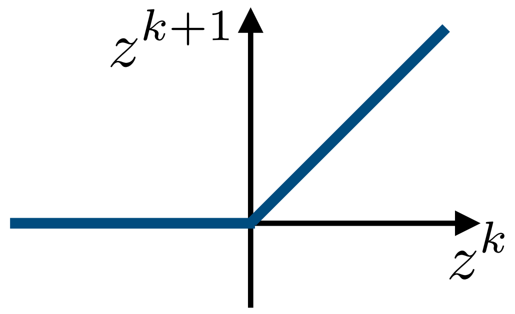


Piecewise affine steps → Bilinear constraints

Elementwise maximum (ReLU)

e.g., one-sided projections

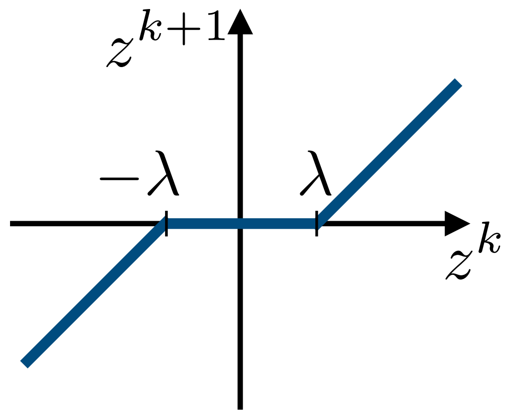
$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$



Soft-thresholding

e.g., prox of 1-norm function

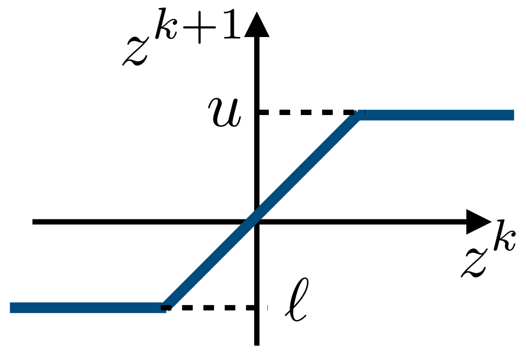
$$z^{k+1} = \phi_\lambda(z^k) = \max\{z^k, \lambda\} - \max\{-z^k, \lambda\}$$



Saturated linear unit (SatLin)

e.g., box projections

$$z^{k+1} = \mathcal{S}_{[\ell,u]}(z^k) = \min\{\max\{z^k, \ell\}, u\}$$



Example:
Nonnegative least squares

minimize $\quad (1/2)\|Dz - x\|_2^2$

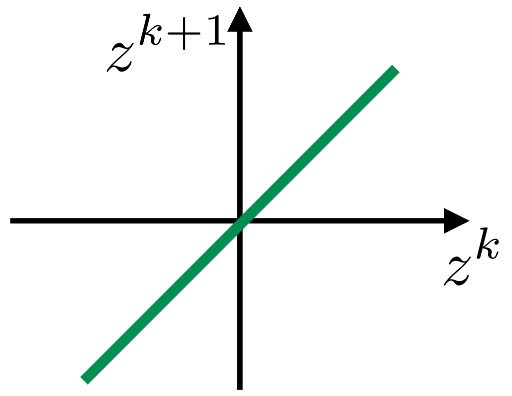
subject to $\quad z \geq 0$

Two types of algorithm steps

Linear steps → Linear constraints

e.g., gradient, momentum, restarts, anchors, prox of quadratic functions…

$$Mz^{k+1} = Az^k + Bx$$

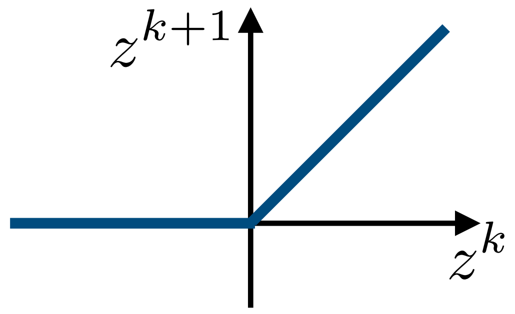


Piecewise affine steps → Bilinear constraints

Elementwise maximum (ReLU)

e.g., one-sided projections

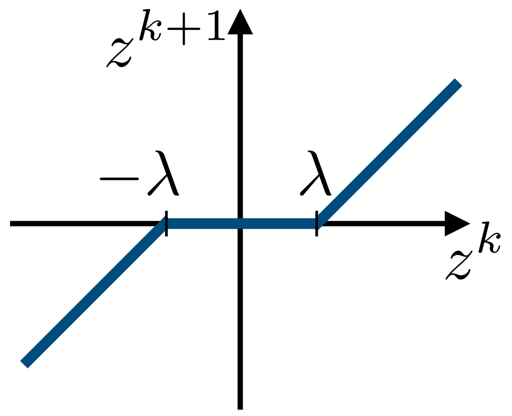
$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$



Soft-thresholding

e.g., prox of 1-norm function

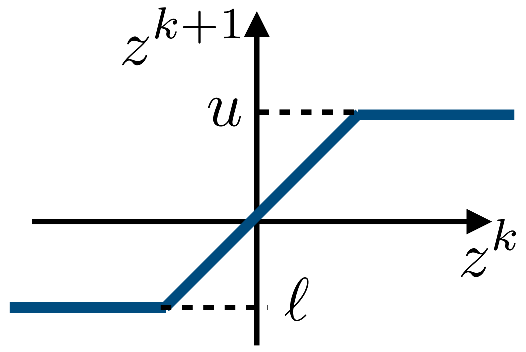
$$z^{k+1} = \phi_\lambda(z^k) = \max\{z^k, \lambda\} - \max\{-z^k, \lambda\}$$



Saturated linear unit (SatLin)

e.g., box projections

$$z^{k+1} = \mathcal{S}_{[\ell,u]}(z^k) = \min\{\max\{z^k, \ell\}, u\}$$



Example:
Nonnegative least squares

minimize	$(1/2)\|Dz - x\|_2^2$
subject to	$z \geq 0$

Projected Gradient Descent

$$w^{k+1} = (I - \theta D^T D)z^k + \theta D^T x$$

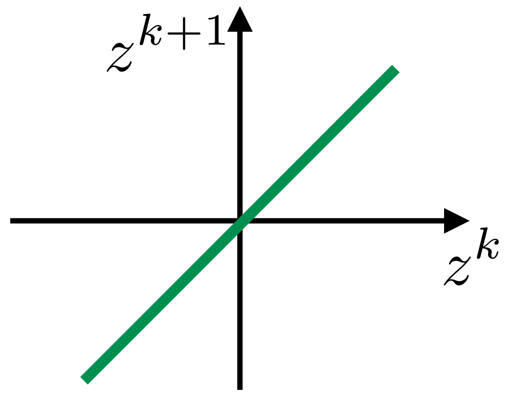
$$z^{k+1} = \max\{w^{k+1}, 0\}$$

Two types of algorithm steps

Linear steps → Linear constraints

e.g., gradient, momentum, restarts, anchors, prox of quadratic functions…

$$Mz^{k+1} = Az^k + Bx$$

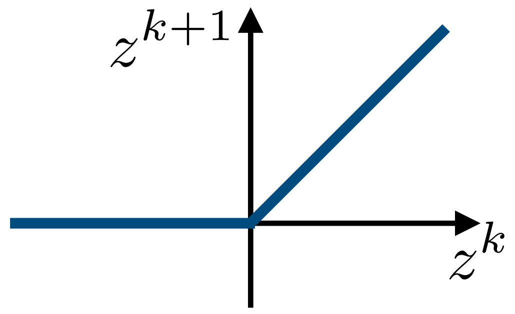


Piecewise affine steps → Bilinear constraints

Elementwise maximum (ReLU)

e.g., one-sided projections

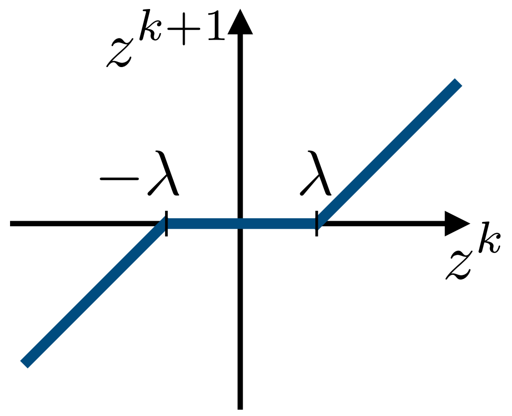
$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$



Soft-thresholding

e.g., prox of 1-norm function

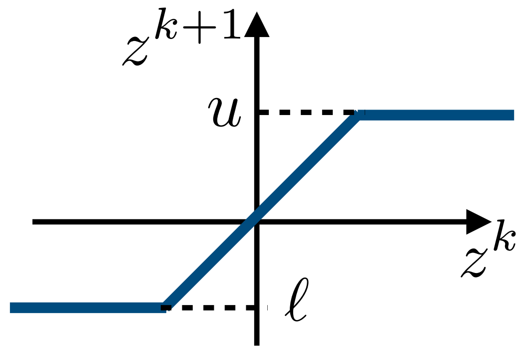
$$z^{k+1} = \phi_\lambda(z^k) = \max\{z^k, \lambda\} - \max\{-z^k, \lambda\}$$



Saturated linear unit (SatLin)

e.g., box projections

$$z^{k+1} = \mathcal{S}_{[\ell,u]}(z^k) = \min\{\max\{z^k, \ell\}, u\}$$



Example:
Nonnegative least squares

minimize $\quad (1/2)\|Dz - x\|_2^2$

subject to $\quad z \geq 0$

gradient step

Projected Gradient Descent

$$w^{k+1} = (I - \theta D^T D)z^k + \theta D^T x$$

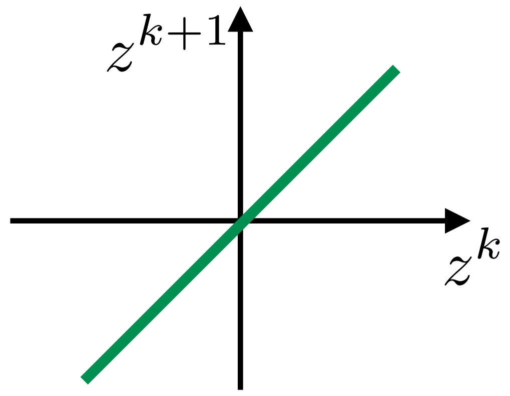
$$z^{k+1} = \max\{w^{k+1}, 0\}$$

Two types of algorithm steps

Linear steps → Linear constraints

e.g., gradient, momentum, restarts, anchors, prox of quadratic functions…

$$Mz^{k+1} = Az^k + Bx$$

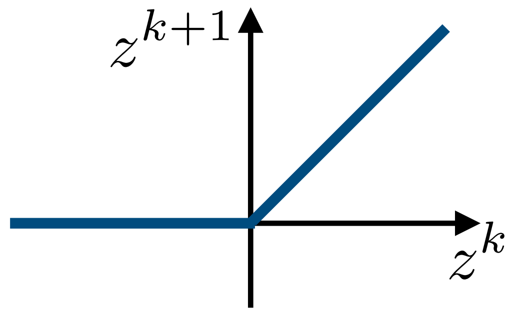


Piecewise affine steps → Bilinear constraints

Elementwise maximum (ReLU)

e.g., one-sided projections

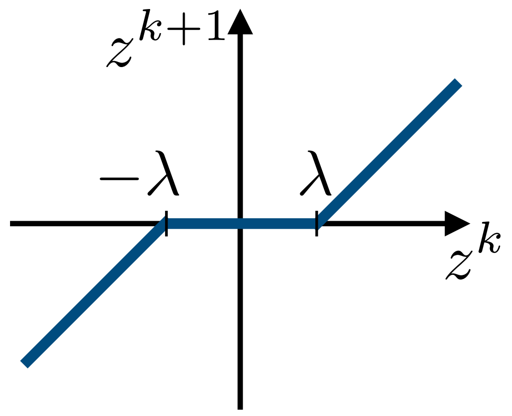
$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$



Soft-thresholding

e.g., prox of 1-norm function

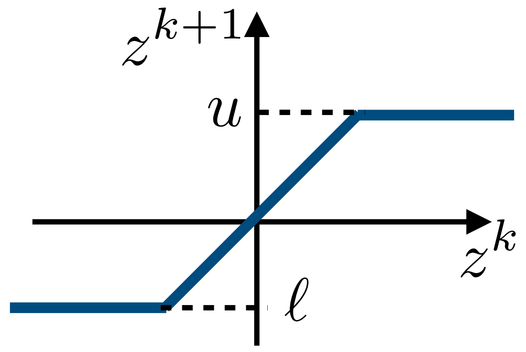
$$z^{k+1} = \phi_\lambda(z^k) = \max\{z^k, \lambda\} - \max\{-z^k, \lambda\}$$



Saturated linear unit (SatLin)

e.g., box projections

$$z^{k+1} = \mathcal{S}_{[\ell,u]}(z^k) = \min\{\max\{z^k, \ell\}, u\}$$



Example:
Nonnegative least squares

minimize $\quad (1/2)\|Dz - x\|_2^2$

subject to $\quad z \geq 0$

gradient step

Projected Gradient Descent

$$w^{k+1} = (I - \theta D^T D)z^k + \theta D^T x$$

$$z^{k+1} = \max\{w^{k+1}, 0\}$$

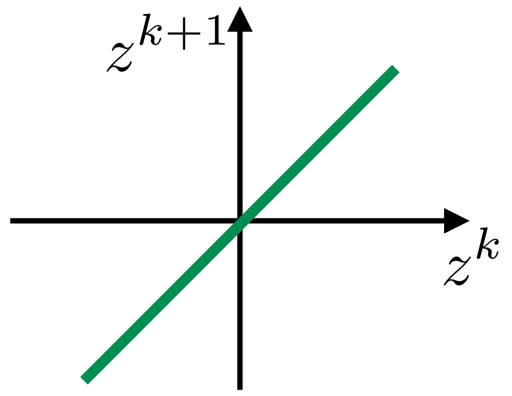
projection step

Two types of algorithm steps

Linear steps → Linear constraints

e.g., gradient, momentum, restarts, anchors, prox of quadratic functions…

$$Mz^{k+1} = Az^k + Bx$$

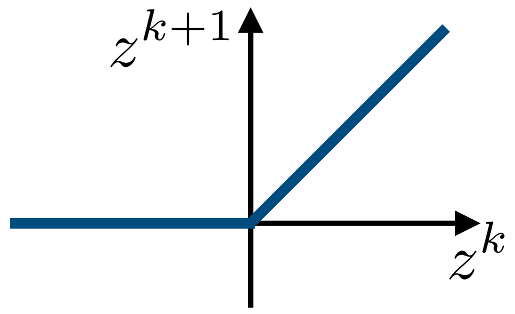


Piecewise affine steps → Bilinear constraints

Elementwise maximum (ReLU)

e.g., one-sided projections

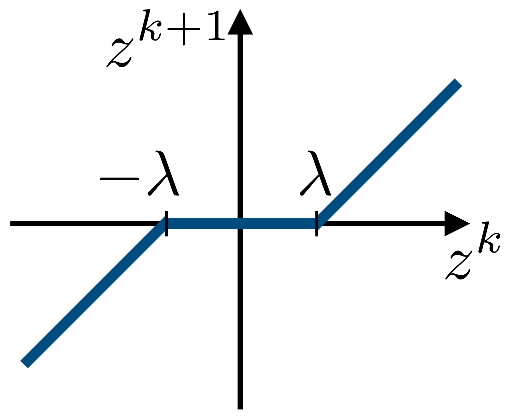
$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$



Soft-thresholding

e.g., prox of 1-norm function

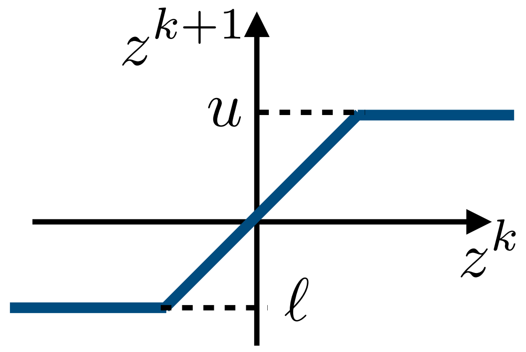
$$z^{k+1} = \phi_\lambda(z^k) = \max\{z^k, \lambda\} - \max\{-z^k, \lambda\}$$



Saturated linear unit (SatLin)

e.g., box projections

$$z^{k+1} = \mathcal{S}_{[\ell,u]}(z^k) = \min\{\max\{z^k, \ell\}, u\}$$



Example:
Nonnegative least squares

minimize $\quad (1/2)\|Dz - x\|_2^2$

subject to $\quad z \geq 0$

Projected Gradient Descent

gradient step

$$w^{k+1} = (I - \theta D^T D)z^k + \theta D^T x$$

projection step

$$z^{k+1} = \max\{w^{k+1}, 0\}$$

similar structure as in neural network verification

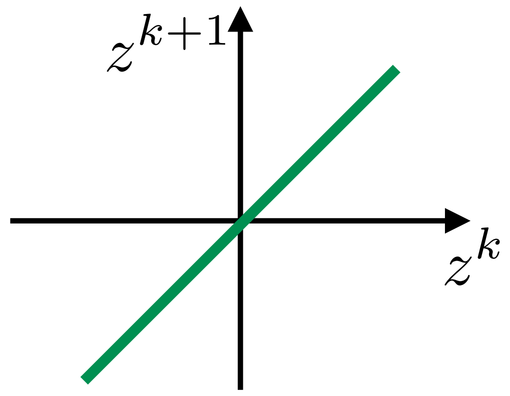
Liu et al. (2021), Albarghouthi (2021), Ceccon et al. (2022), Fischetti and Jo (2018), Tjeng et al. (2019)

Two types of algorithm steps

Linear steps → Linear constraints

e.g., gradient, momentum, restarts, anchors, prox of quadratic functions…

$$Mz^{k+1} = Az^k + Bx$$

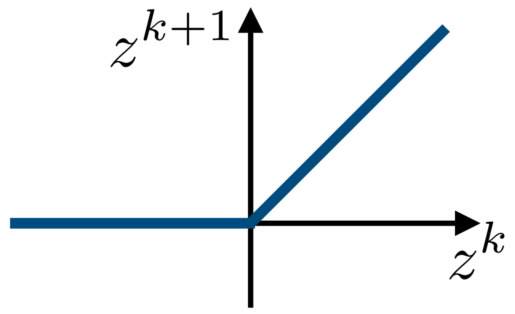


Piecewise affine steps → Bilinear constraints

Elementwise maximum (ReLU)

e.g., one-sided projections

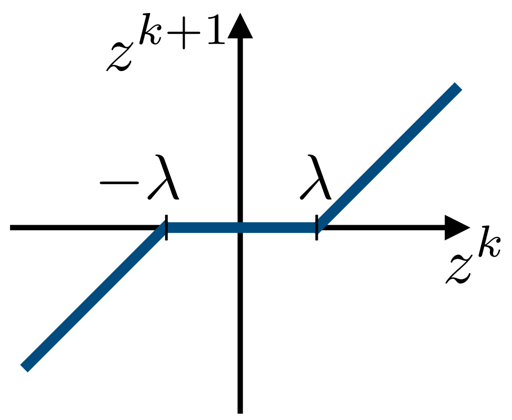
$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$



Soft-thresholding

e.g., prox of 1-norm function

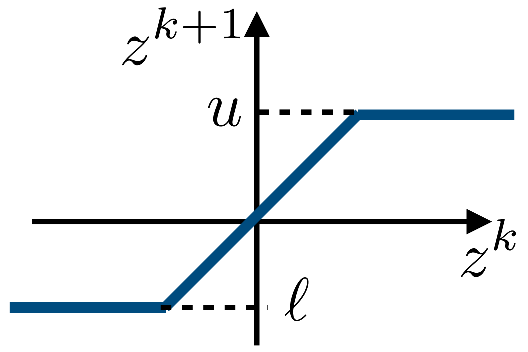
$$z^{k+1} = \phi_\lambda(z^k) = \max\{z^k, \lambda\} - \max\{-z^k, \lambda\}$$



Saturated linear unit (SatLin)

e.g., box projections

$$z^{k+1} = \mathcal{S}_{[\ell,u]}(z^k) = \min\{\max\{z^k, \ell\}, u\}$$



Example:
Nonnegative least squares

minimize $\quad (1/2)\|Dz - x\|_2^2$

subject to $\quad z \geq 0$

gradient step

Projected Gradient Descent

$$w^{k+1} = (I - \theta D^T D)z^k + \theta D^T x$$

$$z^{k+1} = \max\{w^{k+1}, 0\}$$

projection step

similar structure as in neural network verification

Liu et al. (2021), Albarghouthi (2021), Ceccon et al. (2022), Fischetti and Jo (2018), Tjeng et al. (2019)

How can we solve the verification problem?

SDP relaxation of the Verification Problem

$$\max_{x \in \mathcal{X}} r^K(x) = \quad \text{maximize} \quad \boxed{\|z^K - z^{K-1}\|_2^2}$$

$$\text{subject to} \quad z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1$$

$$z^0 = Z(x), \quad x \in \mathcal{X}$$

performance metric
ℓ_2-norm

SDP relaxation of the Verification Problem

$$\max_{x \in \mathcal{X}} r^K(x) = \quad \text{maximize} \quad \|z^K - z^{K-1}\|_2^2$$

$$\text{subject to} \quad z^{k+1} = T(z^k, x), \quad k = 0, \ldots, K-1$$

$$z^0 = Z(x), \quad x \in \mathcal{X}$$

performance metric
ℓ_2-norm

piecewise affine steps → relaxations

Elementwise maximum (ReLU)
e.g., one-sided projections

$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$

SDP relaxation of the Verification Problem

$$\max_{x\in\mathcal{X}} r^K(x) = \quad \text{maximize} \quad \boxed{\|z^K - z^{K-1}\|_2^2}$$

$$\text{subject to} \quad z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1$$

$$z^0 = Z(x), \quad x \in \mathcal{X}$$

performance metric
ℓ_2-norm

piecewise affine steps → relaxations

Elementwise maximum (ReLU)
e.g., one-sided projections

$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\} \longrightarrow$$

bilinear constraints

$$z^{k+1} \geq 0, \quad z^{k+1} \geq z^k$$

$$(z^{k+1})^T(z^{k+1} - z^k) = 0$$

SDP relaxation of the Verification Problem

$$\max_{x \in \mathcal{X}} r^K(x) = \quad \text{maximize} \quad \|z^K - z^{K-1}\|_2^2$$
$$\text{subject to} \quad z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1$$
$$z^0 = Z(x), \quad x \in \mathcal{X}$$

performance metric
ℓ_2-norm

piecewise affine steps → relaxations

Elementwise maximum (ReLU)
e.g., one-sided projections

$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$

bilinear constraints

$$z^{k+1} \geq 0, \quad z^{k+1} \geq z^k$$
$$(z^{k+1})^T(z^{k+1} - z^k) = 0$$

SDP relaxation

$$z^{k+1} \geq 0, \quad z^{k+1} \geq z^k$$
$$\mathbf{tr}\left(\begin{bmatrix} I & -I/2 \\ -I/2 & 0 \end{bmatrix} M\right) = 0$$

⚠️ depends on iterate dimensions!

$$M \succeq \begin{bmatrix} z^{k+1} \\ z^k \end{bmatrix} \begin{bmatrix} z^{k+1} \\ z^k \end{bmatrix}^T$$

SDP Example: unconstrained QP verification

problem instances

$$\text{minimize} \quad (1/2)z^T P z + \underset{\substack{\uparrow\\ \text{problem instances}}}{x}^T z$$

SDP Example: unconstrained QP verification

problem instances

$$\text{minimize} \quad (1/2)z^T P z + x^T z$$

verification problem

gradient descent

$$\begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\|_2^2 \\ \text{subject to} & z^{k+1} = z^k - \theta(Pz^k + x), \quad k = 0, \ldots, K-1 \\ & z^0 \in Z, \quad x \in \mathcal{X} \end{array}$$

warm-start set

SDP Example: unconstrained QP verification

problem instances

$$\text{minimize} \quad (1/2)z^T P z + \boxed{x}^T z$$

verification problem

gradient descent

$$\begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\|_2^2 \\ \text{subject to} & \boxed{z^{k+1} = z^k - \theta(Pz^k + x)}, \quad k = 0, \dots, K-1 \\ & z^0 \in \boxed{Z}, \quad x \in \mathcal{X} \end{array}$$

warm-start set

If Z is an ellipsoid and $\mathcal{X}$ a singleton (or viceversa)

$\Rightarrow$ the SDP formulation is exact!

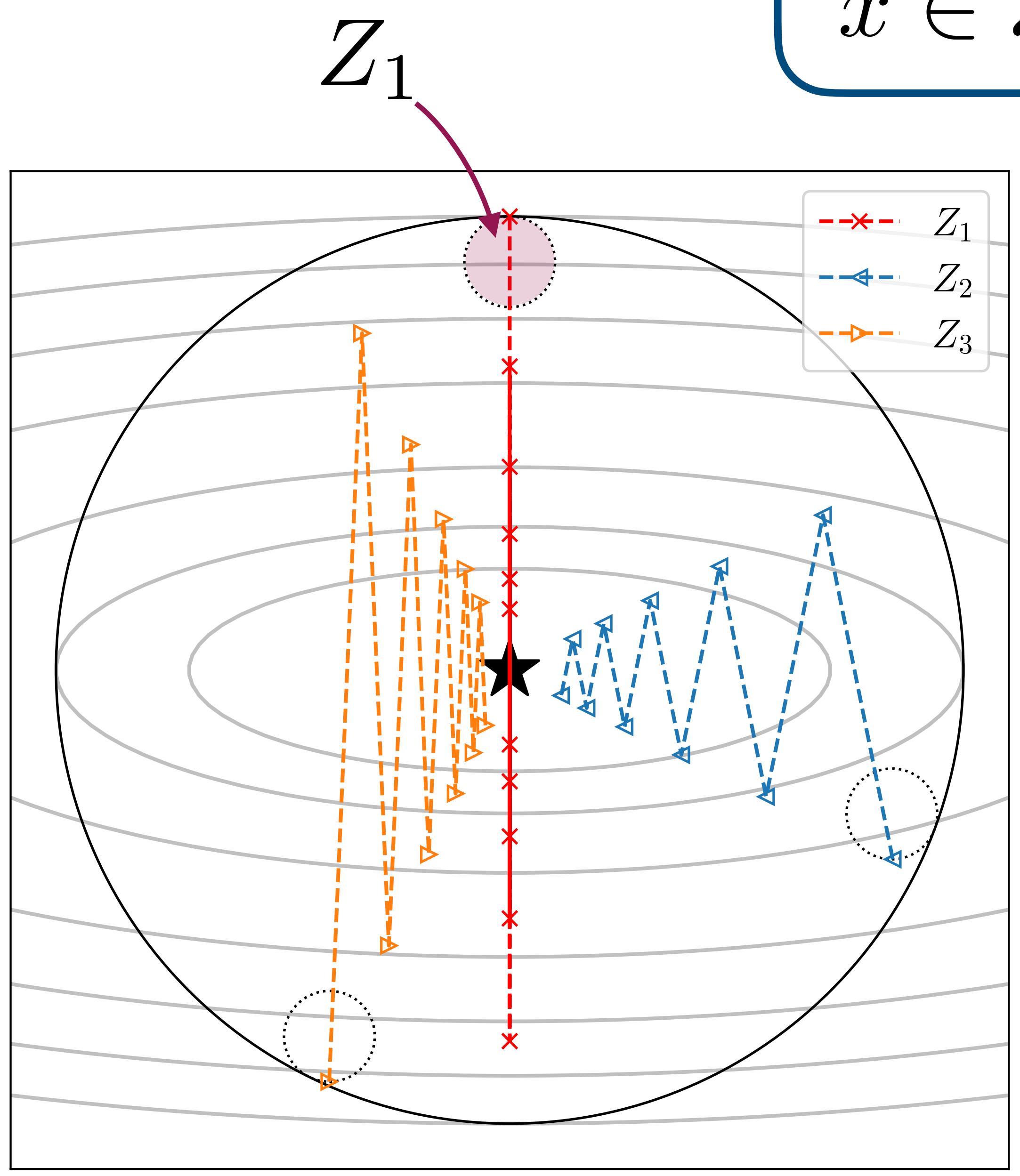
S lemma

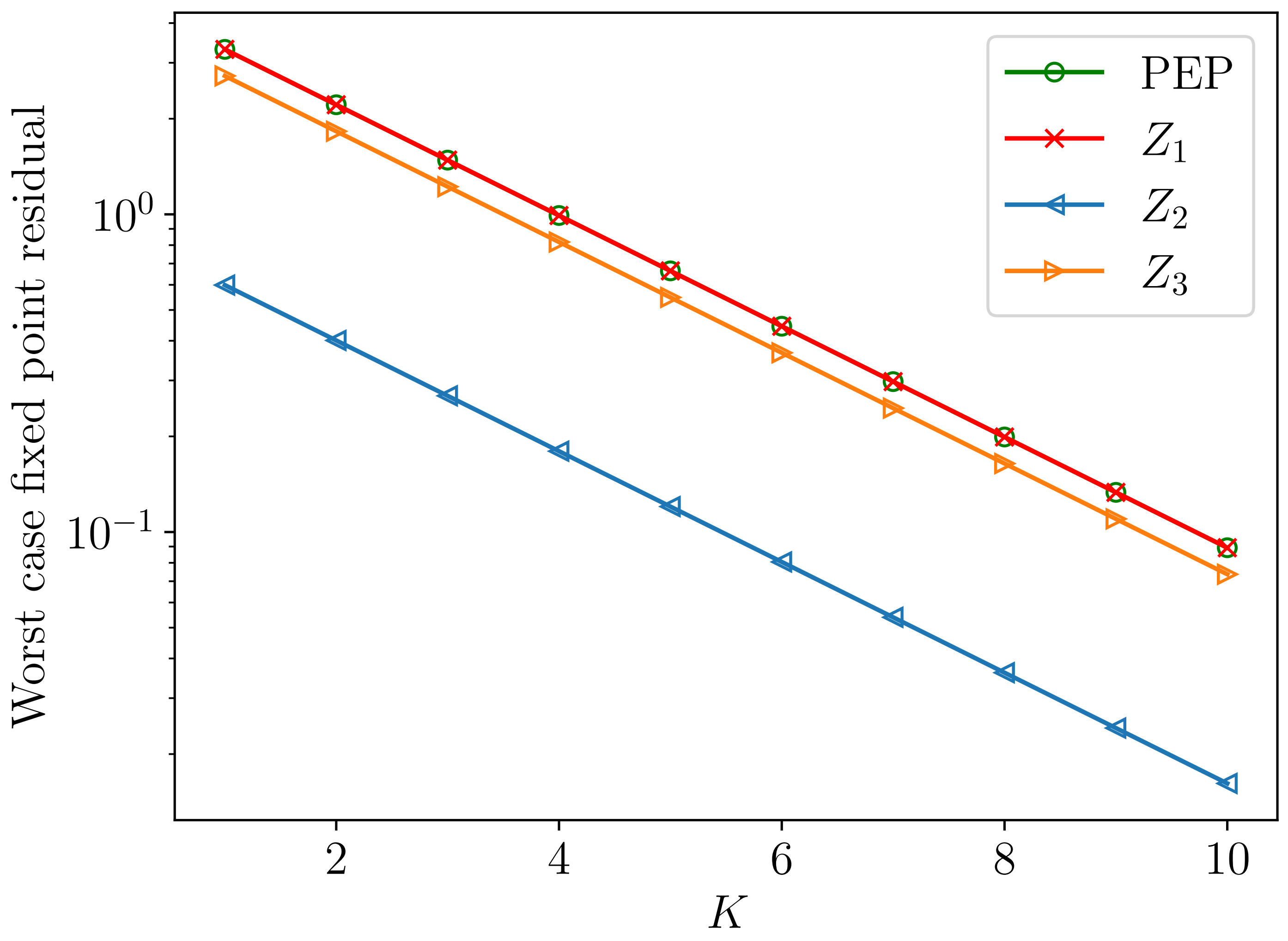
SDP Example (exact): unconstrained QP verification

warm-starts

$Z = Z_1, Z_2,$ or Z_3

$x \in \mathcal{X} = \{0\}$





PEP SDP can't distinguish warm-starts

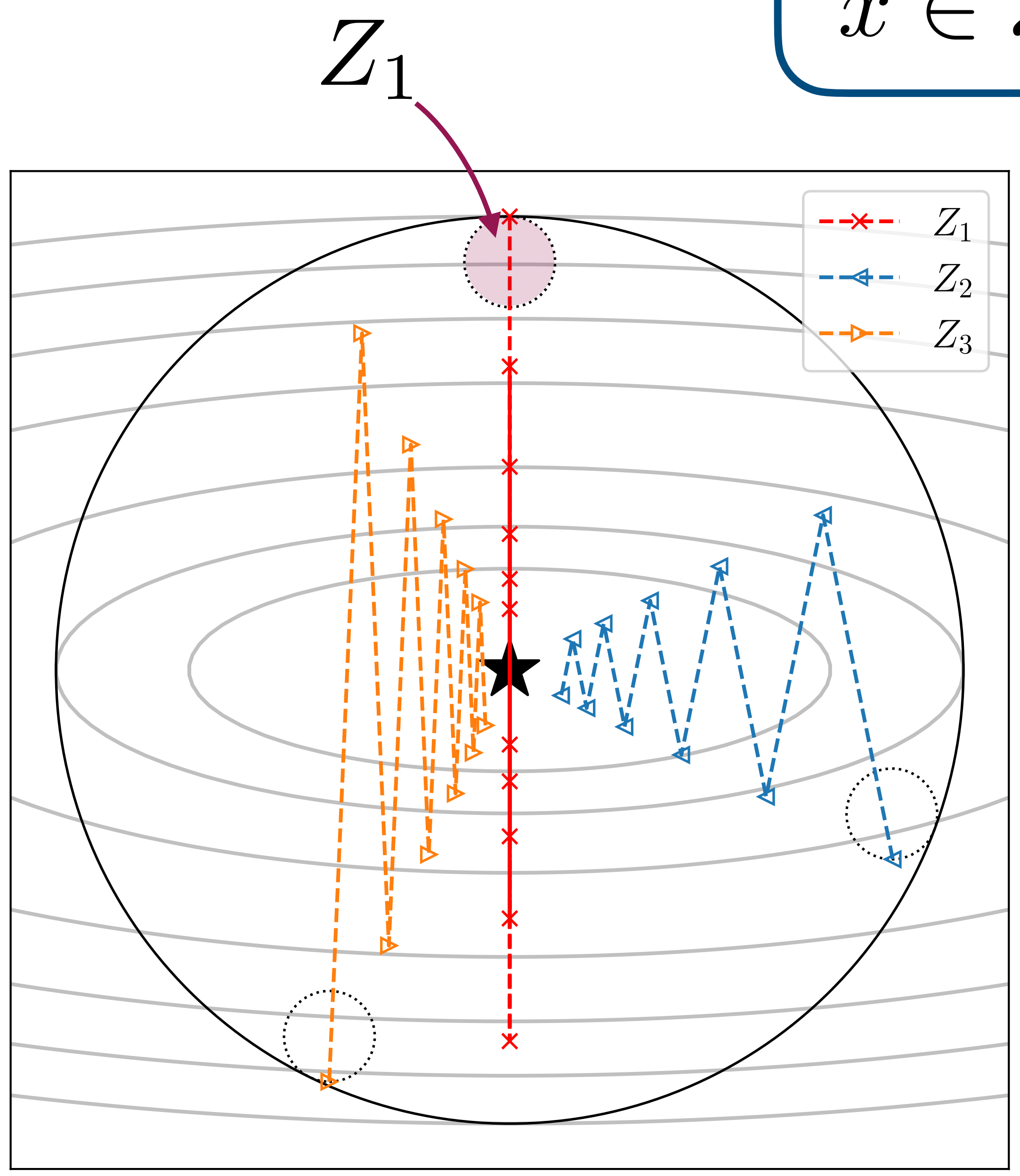
Verification of First-Order Methods for Parametric Quadratic Optimization
V. Ranjan and B. Stellato
arXiv e-prints:2403.03331 (2025)
github.com/stellatogrp/sdp_algo_verify

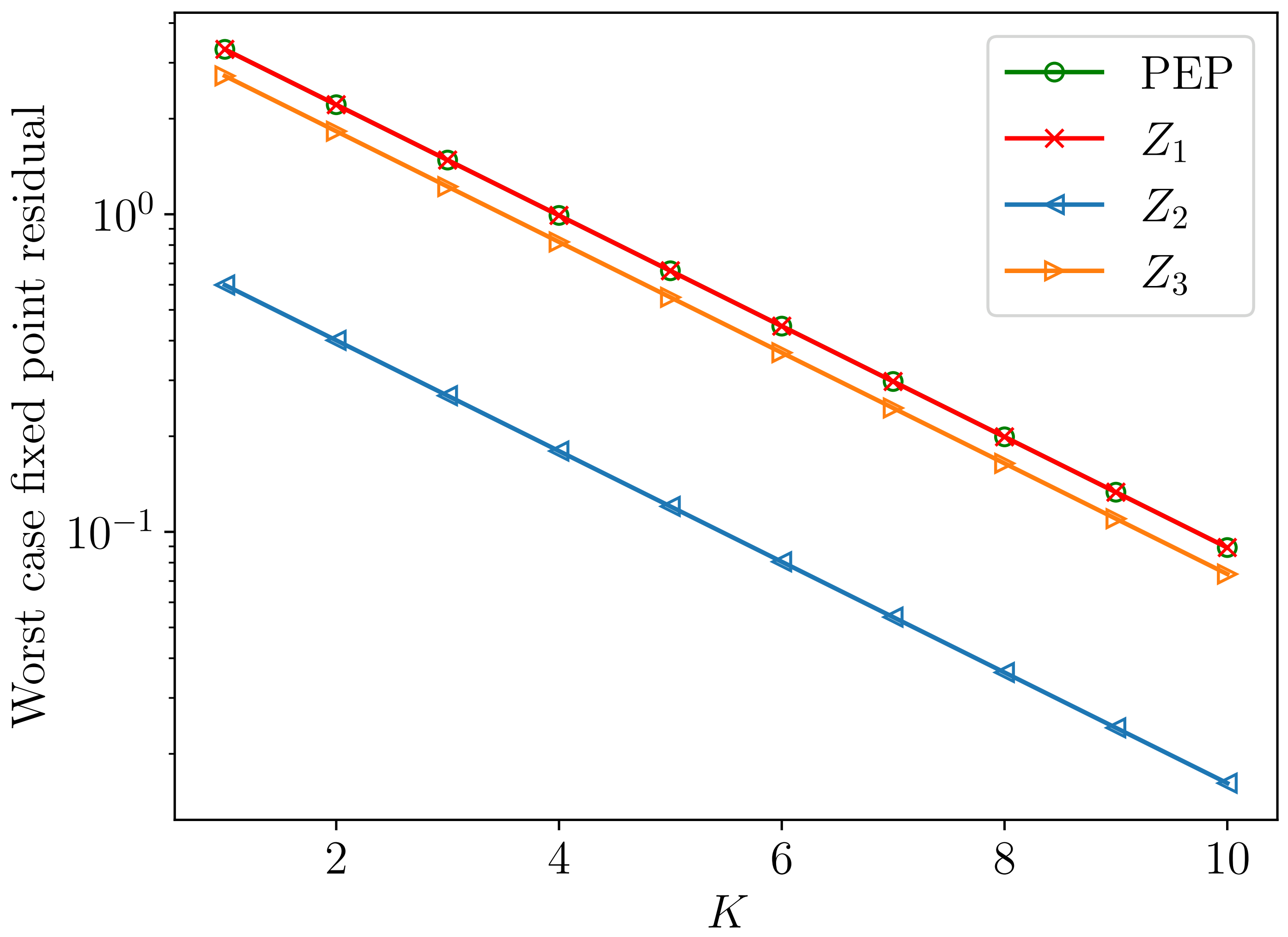
SDP Example (exact): unconstrained QP verification

warm-starts

$Z = Z_1, Z_2,$ or Z_3

$x \in \mathcal{X} = \{0\}$





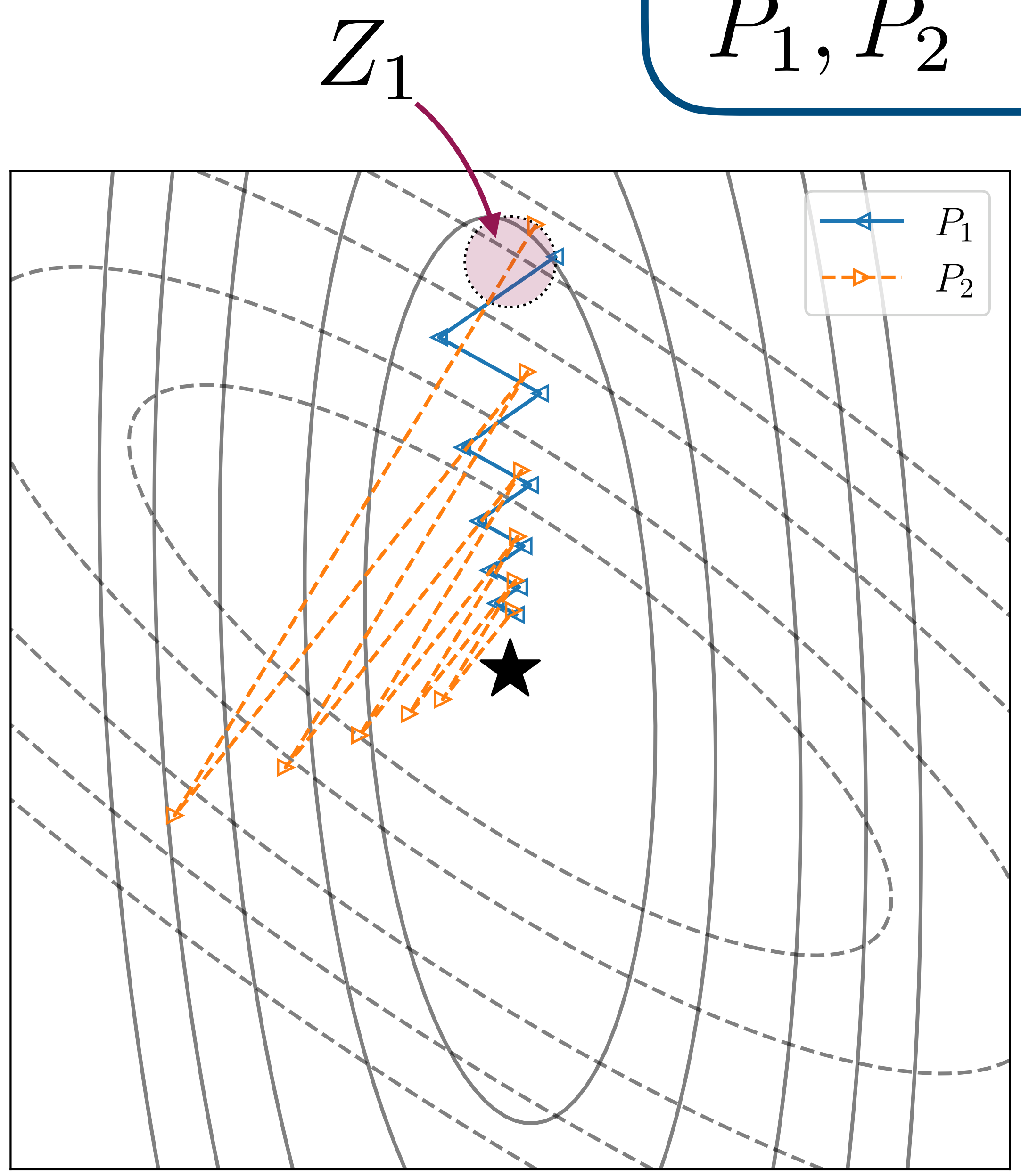
PEP SDP can't distinguish warm-starts

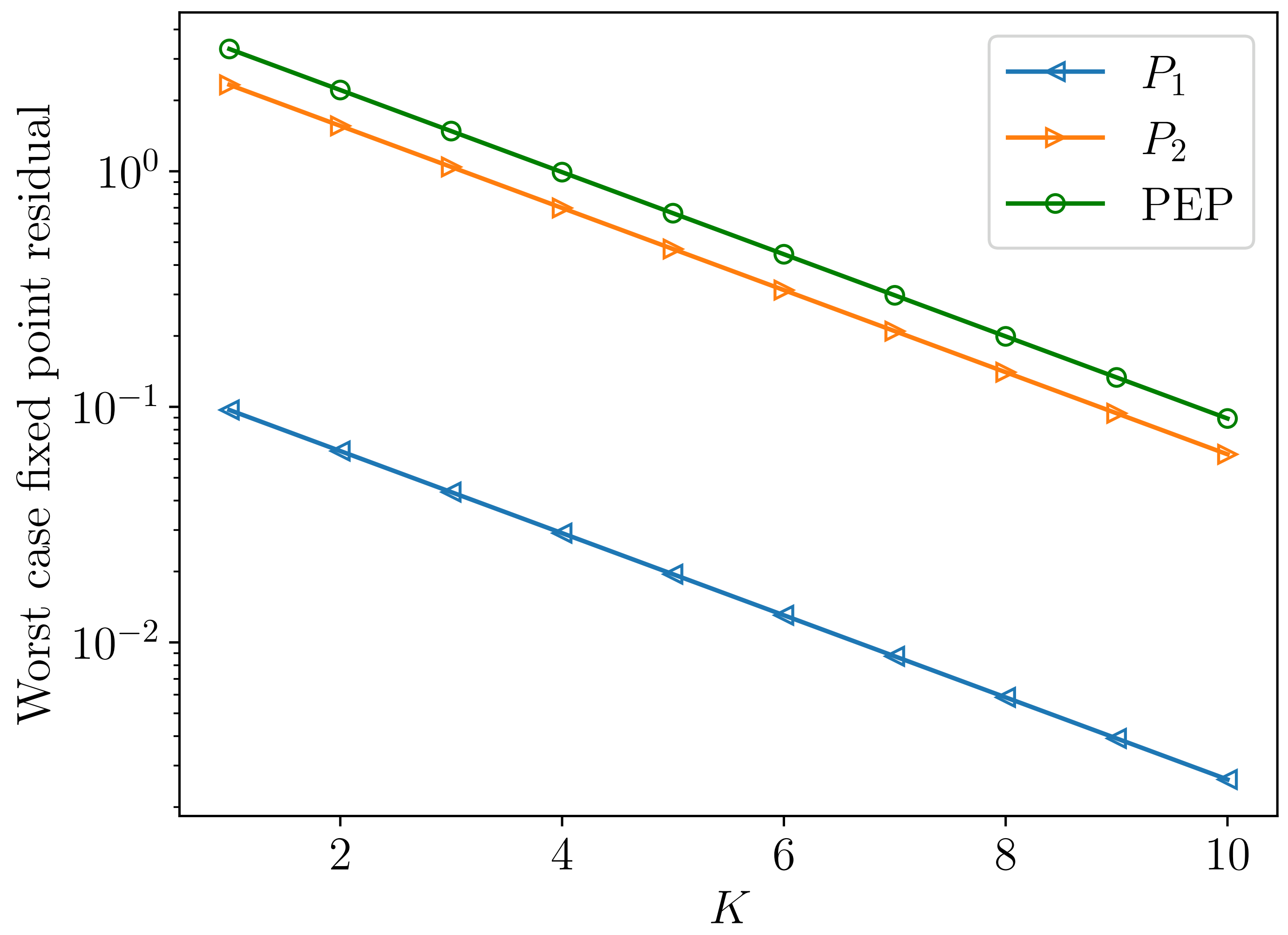
rotated functions

$Z = Z_1$

$x \in \mathcal{X} = \{0\}$

P_1, P_2 rotations of P





PEP SDP can't distinguish rotated quadratic functions

Verification of First-Order Methods for Parametric Quadratic Optimization
V. Ranjan and B. Stellato
arXiv e-prints:2403.03331 (2025)
github.com/stellatogrp/sdp_algo_verify

Exact MIP formulation of the Verification Problem

$$\max_{x \in \mathcal{X}} r^K(x) = \begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\|_\infty \\ \text{subject to} & z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1 \\ & z^0 = Z(x), \quad x \in \mathcal{X} \end{array}$$

Exact MIP formulation of the Verification Problem

$$\max_{x \in \mathcal{X}} r^K(x) = \quad \begin{array}{ll} \text{maximize} & \boxed{\|z^K - z^{K-1}\|_\infty} \\ \text{subject to} & z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1 \\ & z^0 = Z(x), \quad x \in \mathcal{X} \end{array}$$

performance metric
ℓ_∞-norm

Exact MIP formulation of the Verification Problem

$$\max_{x \in \mathcal{X}} r^K(x) = \quad \text{maximize} \quad \boxed{\|z^K - z^{K-1}\|_\infty} \longleftarrow \text{performance metric } \ell_\infty\text{-norm}$$

$$\text{subject to} \quad z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1$$

$$z^0 = Z(x), \quad x \in \mathcal{X}$$

piecewise affine steps → exact reformulations

Elementwise maximum (ReLU)

e.g., one-sided projections

$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$

Exact MIP formulation of the Verification Problem

$$\max_{x \in \mathcal{X}} r^K(x) = \quad \text{maximize} \quad \boxed{\|z^K - z^{K-1}\|_\infty}$$

performance metric
ℓ_∞-norm

$$\text{subject to} \quad z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1$$

$$z^0 = Z(x), \quad x \in \mathcal{X}$$

piecewise affine steps → exact reformulations

Elementwise maximum (ReLU)
e.g., one-sided projections

$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\} \longrightarrow$$

MILP constraints

$$z^{k+1} \geq 0, \quad z^{k+1} \geq z^k$$

$$z^{k+1} \leq z^k - \mathbf{diag}(1-\delta)\underline{z}^k$$

$$z^{k+1} \leq \mathbf{diag}(\delta)\bar{z}^k$$

$$\delta_i \in \{0, 1\}$$

Exact MIP formulation of the Verification Problem

$$\max_{x \in \mathcal{X}} r^K(x) = \quad \text{maximize} \quad \boxed{\|z^K - z^{K-1}\|_\infty}$$

performance metric
ℓ_∞-norm

$$\text{subject to} \quad z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1$$

$$z^0 = Z(x), \quad x \in \mathcal{X}$$

piecewise affine steps → exact reformulations

Elementwise maximum (ReLU)
e.g., one-sided projections

$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$

MILP constraints

$$z^{k+1} \geq 0, \quad z^{k+1} \geq z^k$$

$$z^{k+1} \leq z^k - \mathbf{diag}(1 - \delta)\underline{z}^k$$

$$z^{k+1} \leq \mathbf{diag}(\delta)\bar{z}^k$$

$$\delta_i \in \{0, 1\}$$

binary variables

Exact MIP formulation of the Verification Problem

$$\max_{x \in \mathcal{X}} r^K(x) = \quad \text{maximize} \quad \boxed{\|z^K - z^{K-1}\|_\infty}$$

$$\text{subject to} \quad z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1$$

$$z^0 = Z(x), \quad x \in \mathcal{X}$$

performance metric
ℓ_∞-norm

piecewise affine steps → exact reformulations

Elementwise maximum (ReLU)
e.g., one-sided projections

$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\} \longrightarrow$$

MILP constraints

variable bounds

$$z^{k+1} \geq 0, \quad z^{k+1} \geq z^k$$

$$z^{k+1} \leq z^k - \mathbf{diag}(1 - \delta)\underline{z}^k$$

$$z^{k+1} \leq \mathbf{diag}(\delta)\bar{z}^k$$

$$\delta_i \in \{0, 1\}$$

binary variables

Exact MIP formulation of the Verification Problem

$$\max_{x \in \mathcal{X}} r^K(x) = \quad \text{maximize} \quad \|z^K - z^{K-1}\|_\infty$$

performance metric
ℓ_∞-norm

$$\text{subject to} \quad z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1$$

$$z^0 = Z(x), \quad x \in \mathcal{X}$$

piecewise affine steps → exact reformulations

Elementwise maximum (ReLU)
e.g., one-sided projections

$$z^{k+1} = (z^k)_+ = \max\{z^k, 0\}$$

MILP constraints

variable bounds

$$z^{k+1} \geq 0, \quad z^{k+1} \geq z^k$$

$$z^{k+1} \leq z^k - \mathbf{diag}(1 - \delta)\underline{z}^k$$

$$z^{k+1} \leq \mathbf{diag}(\delta)\bar{z}^k$$

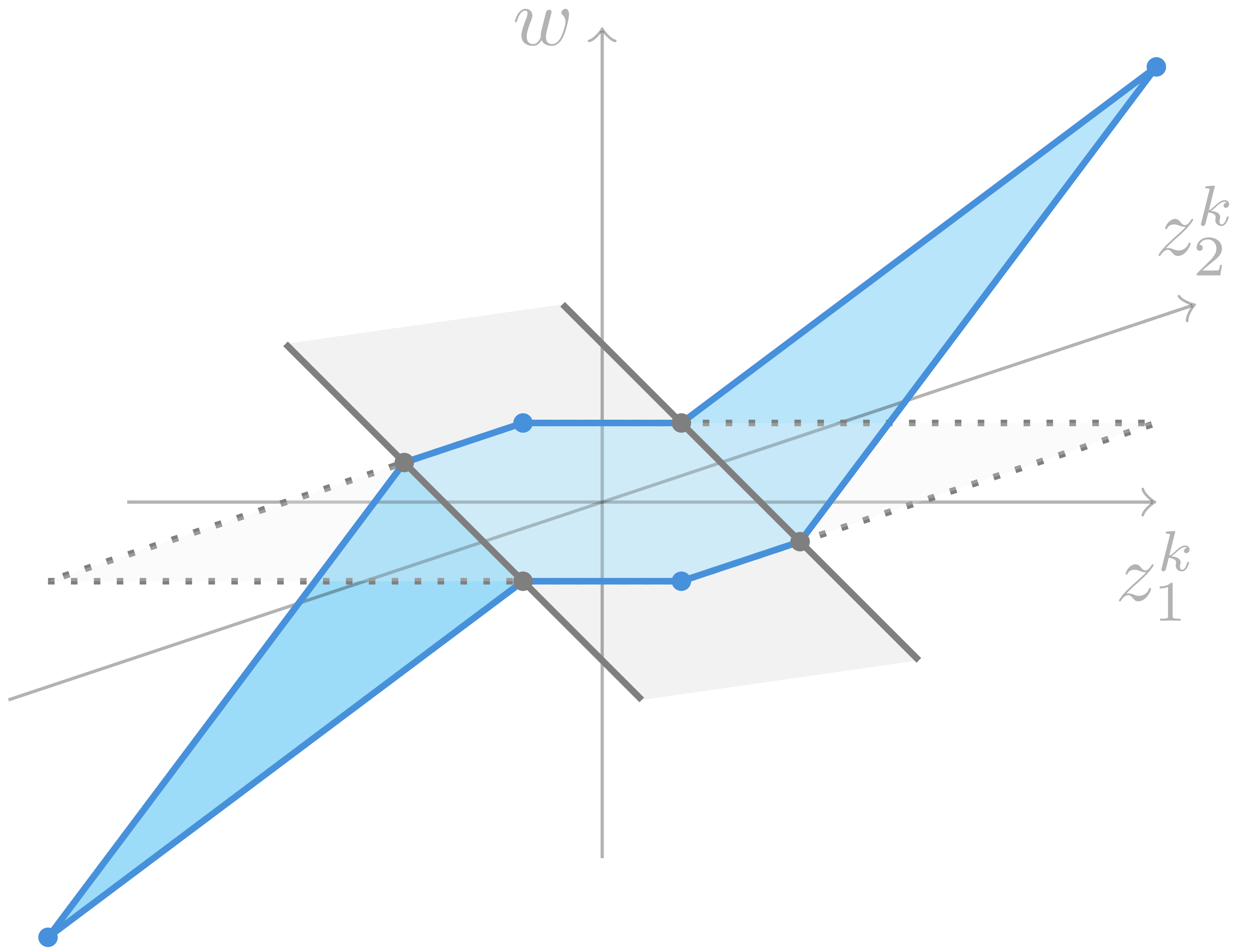
$$\delta_i \in \{0, 1\}$$

binary variables

⚠️ MILP solution may be very slow!

Constructing strong MIP formulations

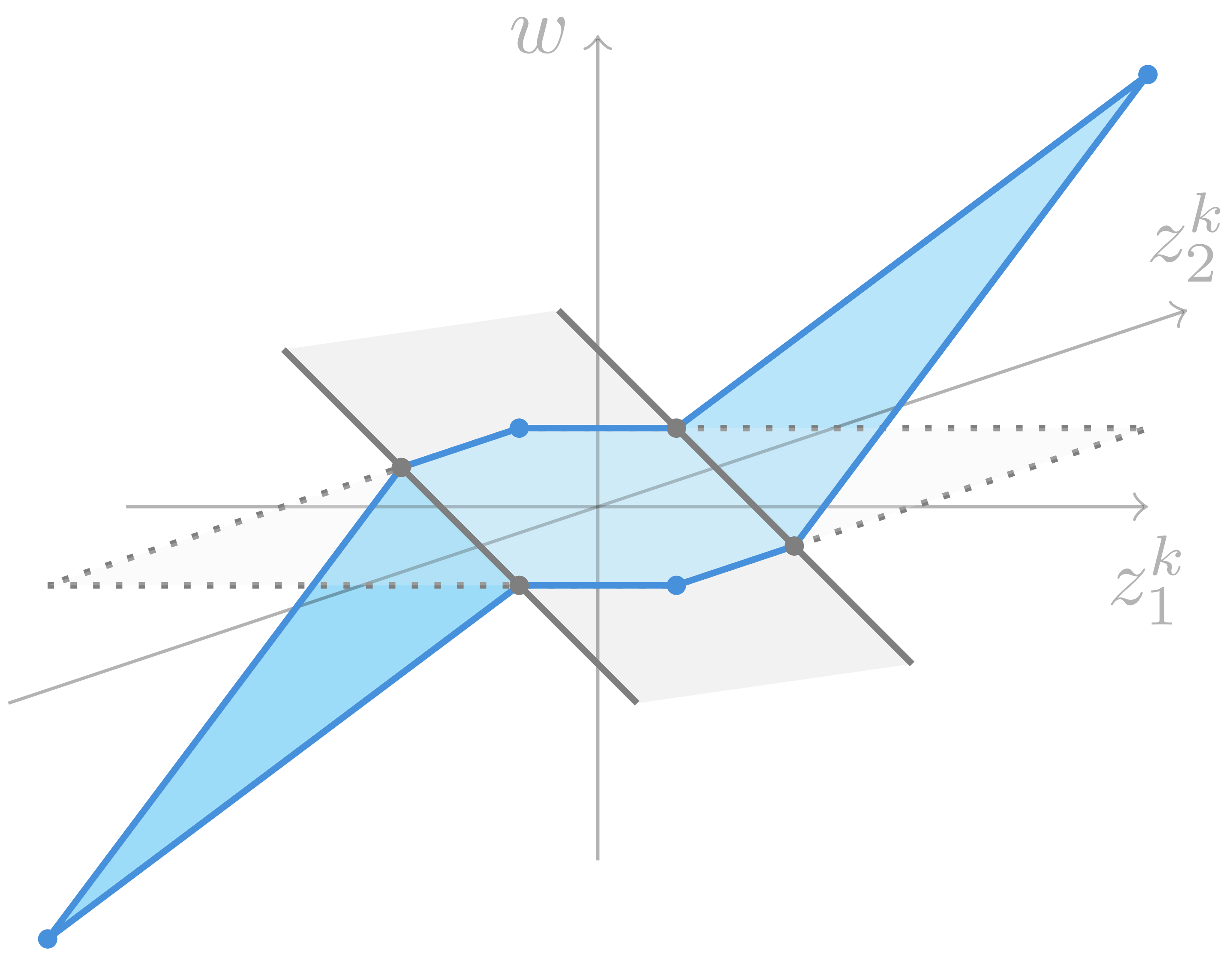
piecewise affine steps
soft-thresholding operator
$$w = \phi_\gamma(z_1^k + z_2^k)$$



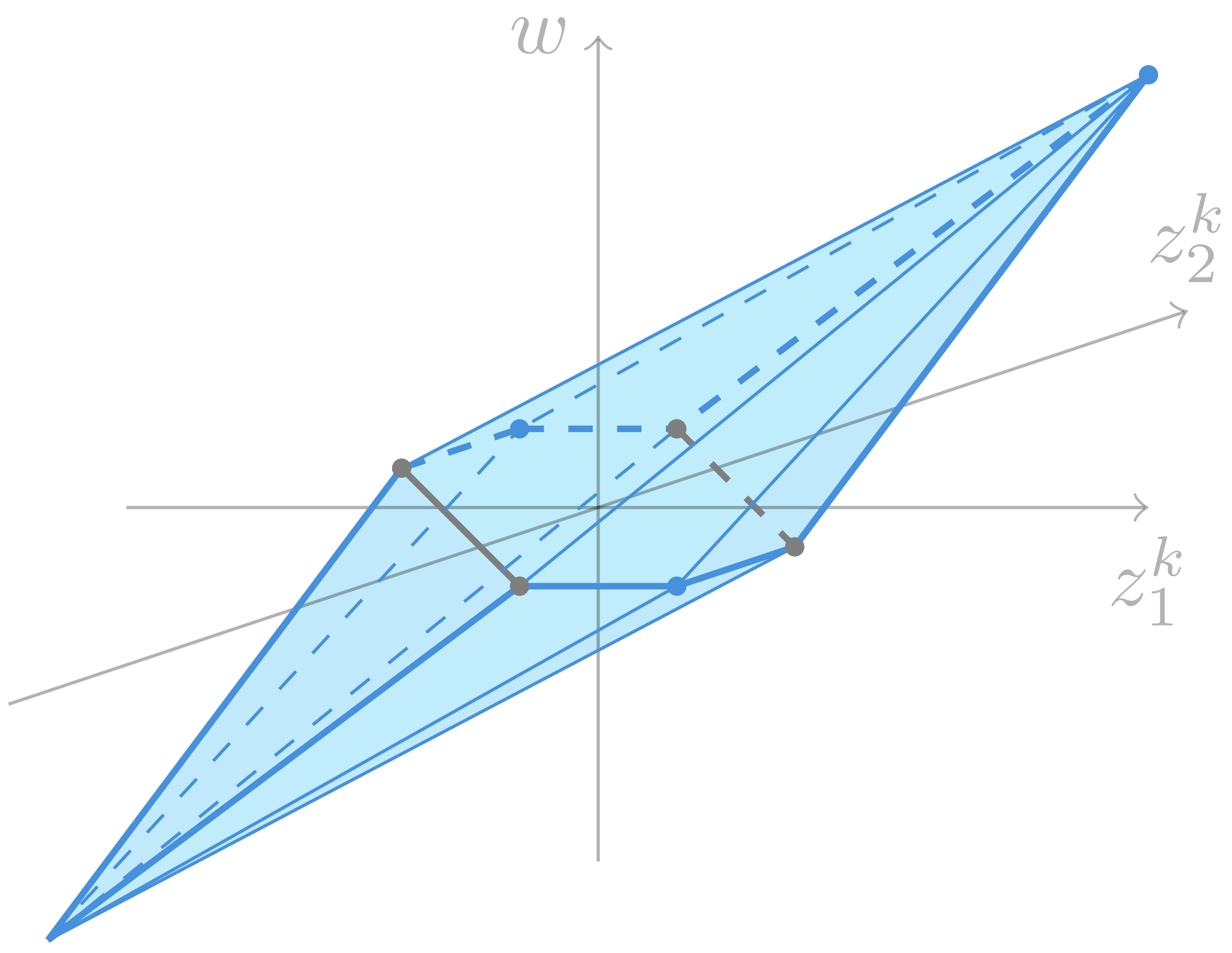
Constructing strong MIP formulations

piecewise affine steps
soft-thresholding operator

$$w = \phi_\gamma(z_1^k + z_2^k)$$



convex hull



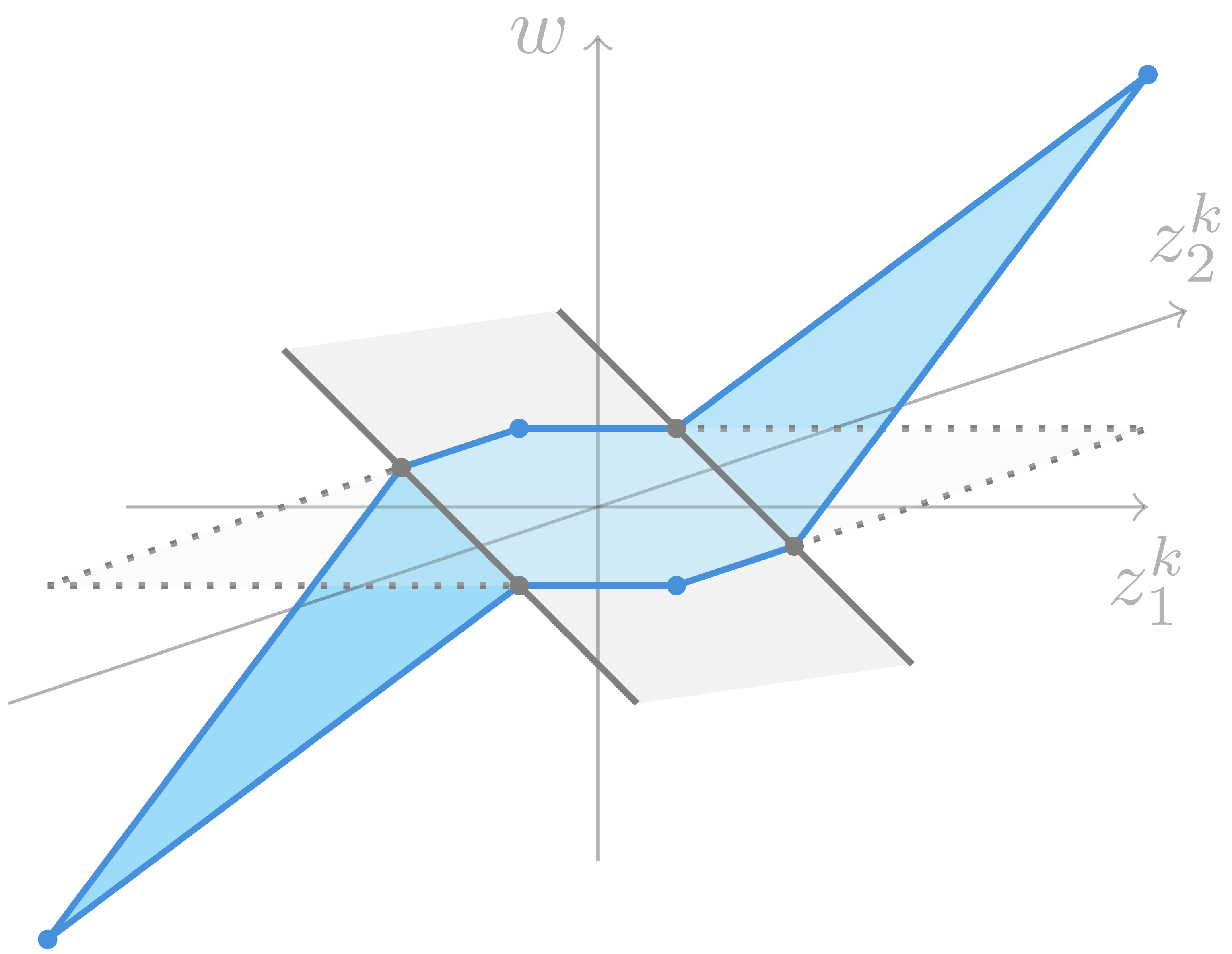
Inspired by: Anderson et al. (2020), Tjandraatmadja et al. (2020), Tsay et al. (2021), Hojny et al. (2024), Huchette et al. (2025)

Constructing strong MIP formulations

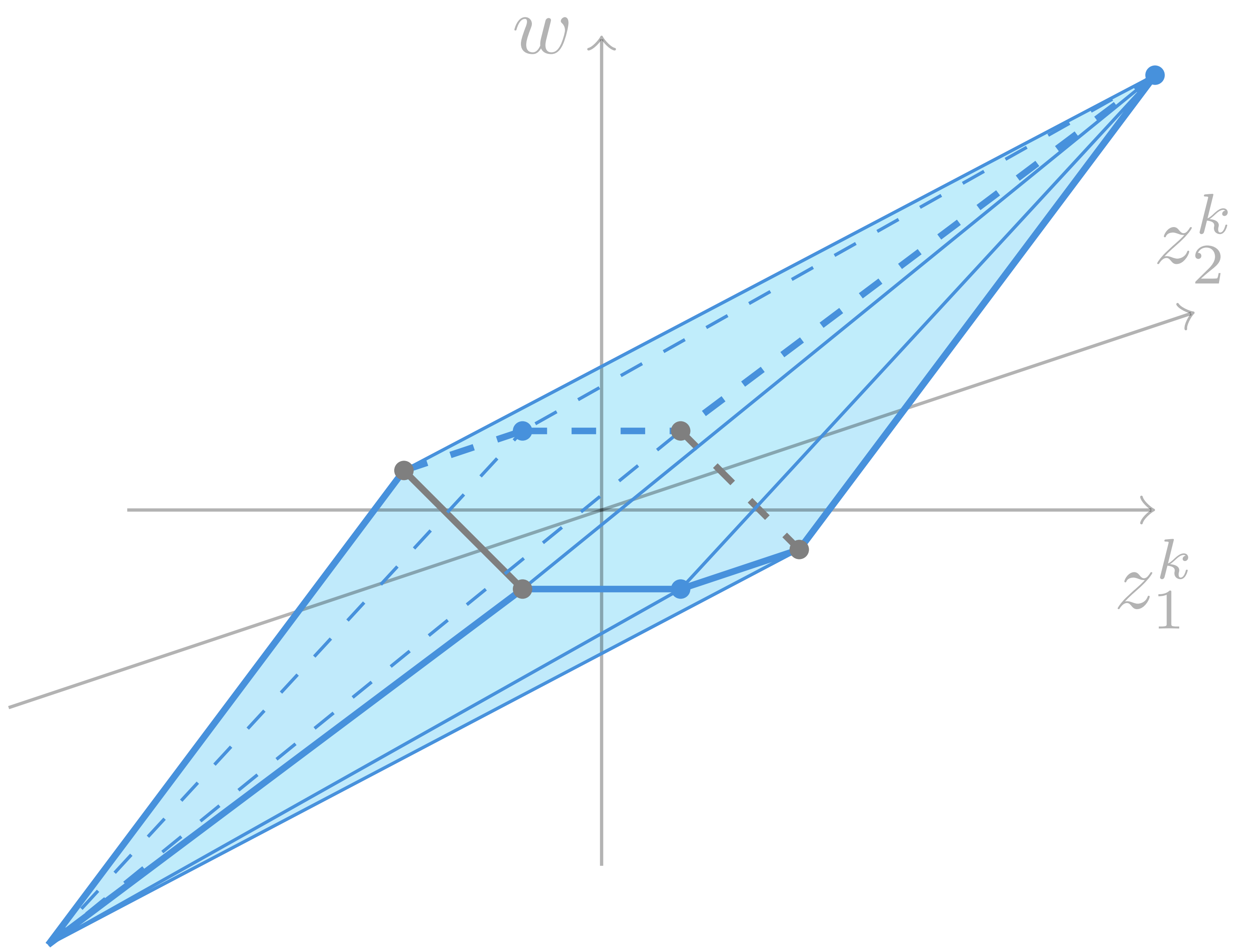
piecewise affine steps

soft-thresholding operator

$$w = \phi_\gamma(z_1^k + z_2^k)$$



convex hull



⚠️ exponential number of inequalities!

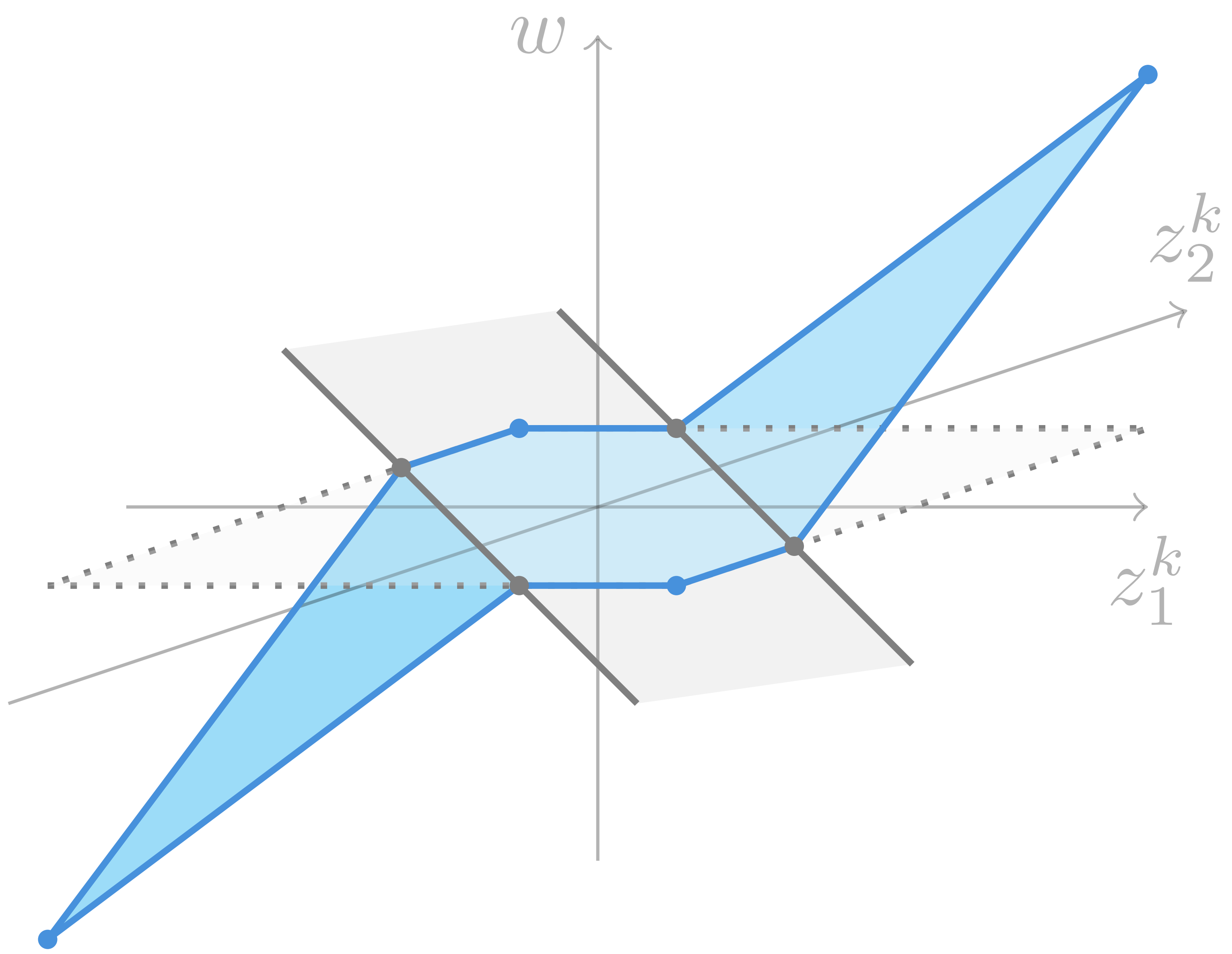
Inspired by: Anderson et al. (2020), Tjandraatmadja et al. (2020), Tsay et al. (2021), Hojny et al. (2024), Huchette et al. (2025)

Constructing strong MIP formulations

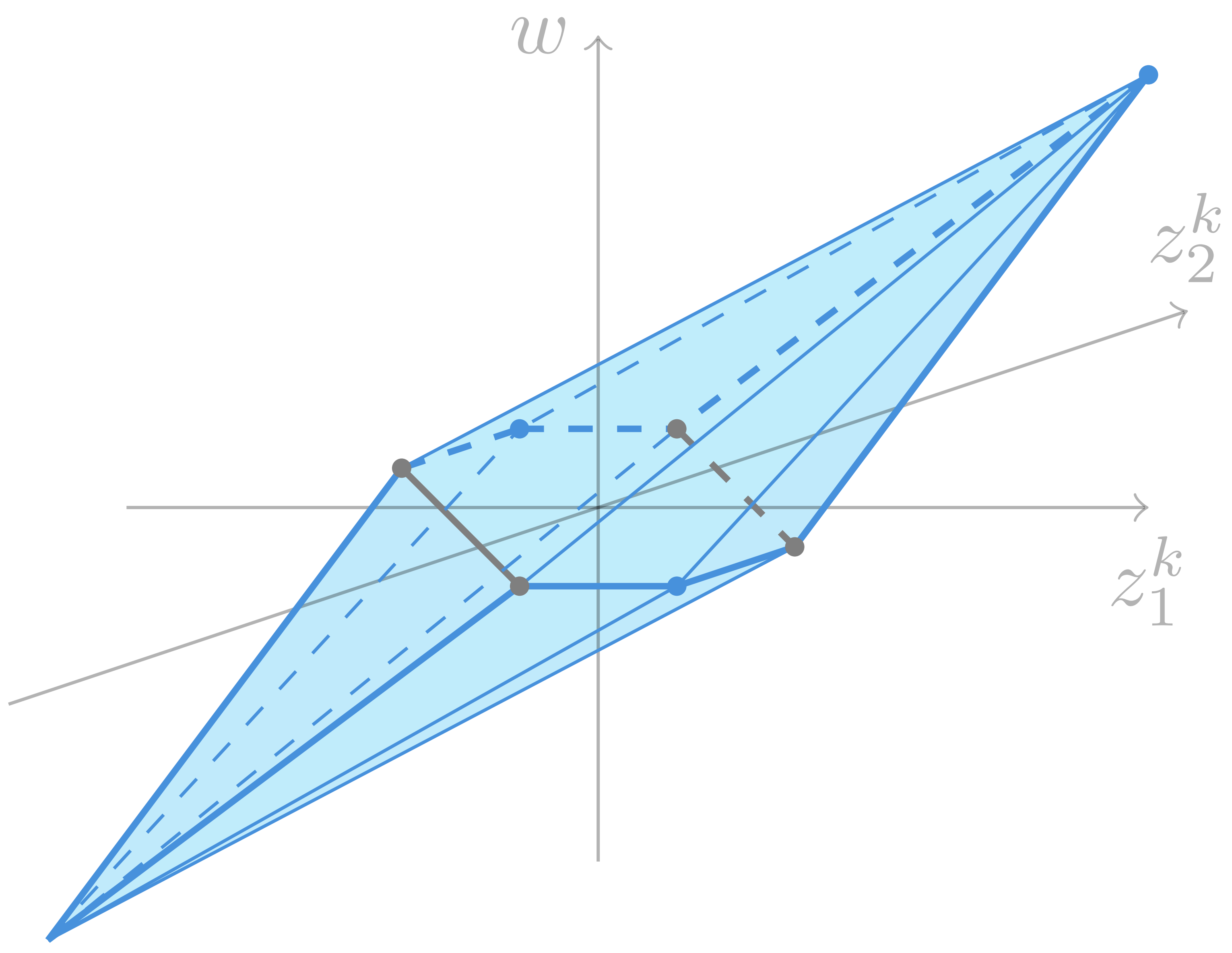
piecewise affine steps

soft-thresholding operator

$$w = \phi_\gamma(z_1^k + z_2^k)$$



convex hull



Inspired by: Anderson et al. (2020), Tjandraatmadja et al. (2020), Tsay et al. (2021), Hojny et al. (2024), Huchette et al. (2025)

exponential number of inequalities!

separation problem solvable in linear time

Using operator theory to tighten MIP formulations

$$
\max_{x \in \mathcal{X}} r^K(x) = \quad \begin{array}{ll} \text{maximize} & \boxed{\|z^K - z^{K-1}\|_\infty} \longleftarrow \textbf{performance metric} \\ \text{subject to} & z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1 \\ & z^0 = Z(x), \quad x \in \mathcal{X} \end{array}
$$

Using operator theory to tighten MIP formulations

$$\max_{x \in \mathcal{X}} r^K(x) = \quad \text{maximize} \quad \|z^K - z^{K-1}\|_\infty \longleftarrow \text{performance metric}$$

$$\text{subject to} \quad z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1$$

$$z^0 = Z(x), \quad x \in \mathcal{X}$$

operator theory bound

$$\|z^K - z^{K-1}\|_\infty \leq \alpha_K$$

e.g., linear convergence

$$\alpha_K = C\tau^K \longleftarrow \textit{rate}$$

Using operator theory to tighten MIP formulations

$$\max_{x \in \mathcal{X}} r^K(x) = \quad \text{maximize} \quad \|z^K - z^{K-1}\|_\infty$$

performance metric

$$\text{subject to} \quad z^{k+1} = T(z^k, x), \quad k = 0, \ldots, K-1$$

$$z^0 = Z(x), \quad x \in \mathcal{X}$$

operator theory bound

$$\|z^K - z^{K-1}\|_\infty \leq \alpha_K$$

e.g., linear convergence

$$\alpha_K = C\tau^K$$

rate

previous iterate bounds

(bound tightening, interval propagation, etc)

$$\underline{z}^{K-1} \leq z^{K-1} \leq \overline{z}^{K-1}$$

Using operator theory to tighten MIP formulations

$$\max_{x \in \mathcal{X}} r^K(x) = \quad \text{maximize} \quad \|z^K - z^{K-1}\|_\infty \quad \longleftarrow \textbf{performance metric}$$

$$\text{subject to} \quad z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1$$

$$z^0 = Z(x), \quad x \in \mathcal{X}$$

operator theory bound

$$\|z^K - z^{K-1}\|_\infty \leq \alpha_K$$

e.g., linear convergence

$$\alpha_K = C\tau^K \longleftarrow \textit{rate}$$

previous iterate bounds

(bound tightening, interval propagation, etc)

$$\underline{z}^{K-1} \leq z^{K-1} \leq \bar{z}^{K-1}$$

combine $\longrightarrow$

bounds on latest iterate

$$\underline{z}^{K-1} - \alpha_K \mathbf{1} \leq z^K \leq \bar{z}^{K-1} + \alpha_K \mathbf{1}$$

Using operator theory to tighten MIP formulations

$$\max_{x \in \mathcal{X}} r^K(x) = \quad \text{maximize} \quad \|z^K - z^{K-1}\|_\infty$$

performance metric

$$\text{subject to} \quad z^{k+1} = T(z^k, x), \quad k = 0, \dots, K-1$$

$$z^0 = Z(x), \quad x \in \mathcal{X}$$

operator theory bound

$$\|z^K - z^{K-1}\|_\infty \leq \alpha_K$$

e.g., linear convergence

$$\alpha_K = C\tau^K$$

rate

previous iterate bounds

(bound tightening, interval propagation, etc)

$$\underline{z}^{K-1} \leq z^{K-1} \leq \bar{z}^{K-1}$$

combine

bounds on latest iterate

$$\underline{z}^{K-1} - \alpha_K \mathbf{1} \leq z^K \leq \bar{z}^{K-1} + \alpha_K \mathbf{1}$$

main idea

solve verification problem for increasing K

..... $\xrightarrow{\text{compute bounds on } z^{K-1}}$ verify for $K-1$ $\xrightarrow{\text{compute bounds on } z^K}$ verify for K — .....

Examples of exact worst-case analysis

Sparse coding for signal reconstruction

$$\text{minimize} \quad (1/2)\|Dz - x\|_2^2 + \lambda\|z\|_1$$

Sparse coding for signal reconstruction

$$\text{minimize} \quad (1/2)\|Dz - x\|_2^2 + \lambda\|z\|_1$$

noisy
signal

Sparse coding for signal reconstruction

$$\text{minimize} \quad (1/2)\|\underbrace{D}_{\text{known dictionary}}z - \underbrace{x}_{\text{noisy signal}}\|_2^2 + \lambda\|z\|_1$$

Sparse coding for signal reconstruction

reconstructed
signal

minimize $\quad (1/2)\|\underbrace{D}_{\text{known dictionary}}z - \underbrace{x}_{\text{noisy signal}}\|_2^2 + \lambda\|z\|_1$

Sparse coding for signal reconstruction

reconstructed signal

$$\text{minimize} \quad (1/2)\|Dz - x\|_2^2 + \lambda\|z\|_1$$

known dictionary

noisy signal

Iterative Soft-Thresholding Algorithm (ISTA)

$$z^{k+1} = \phi_{\lambda\theta}\left((I - \theta D^T D)z^k + \theta D^T x\right)$$

soft-thresholding operator

Sparse coding for signal reconstruction

reconstructed signal

$$\text{minimize} \quad (1/2)\|Dz - x\|_2^2 + \lambda\|z\|_1$$

known dictionary

noisy signal

Iterative Soft-Thresholding Algorithm (ISTA)

$$z^{k+1} = \phi_{\lambda\theta}\left((I - \theta D^T D)z^k + \theta D^T x\right)$$

soft-thresholding operator

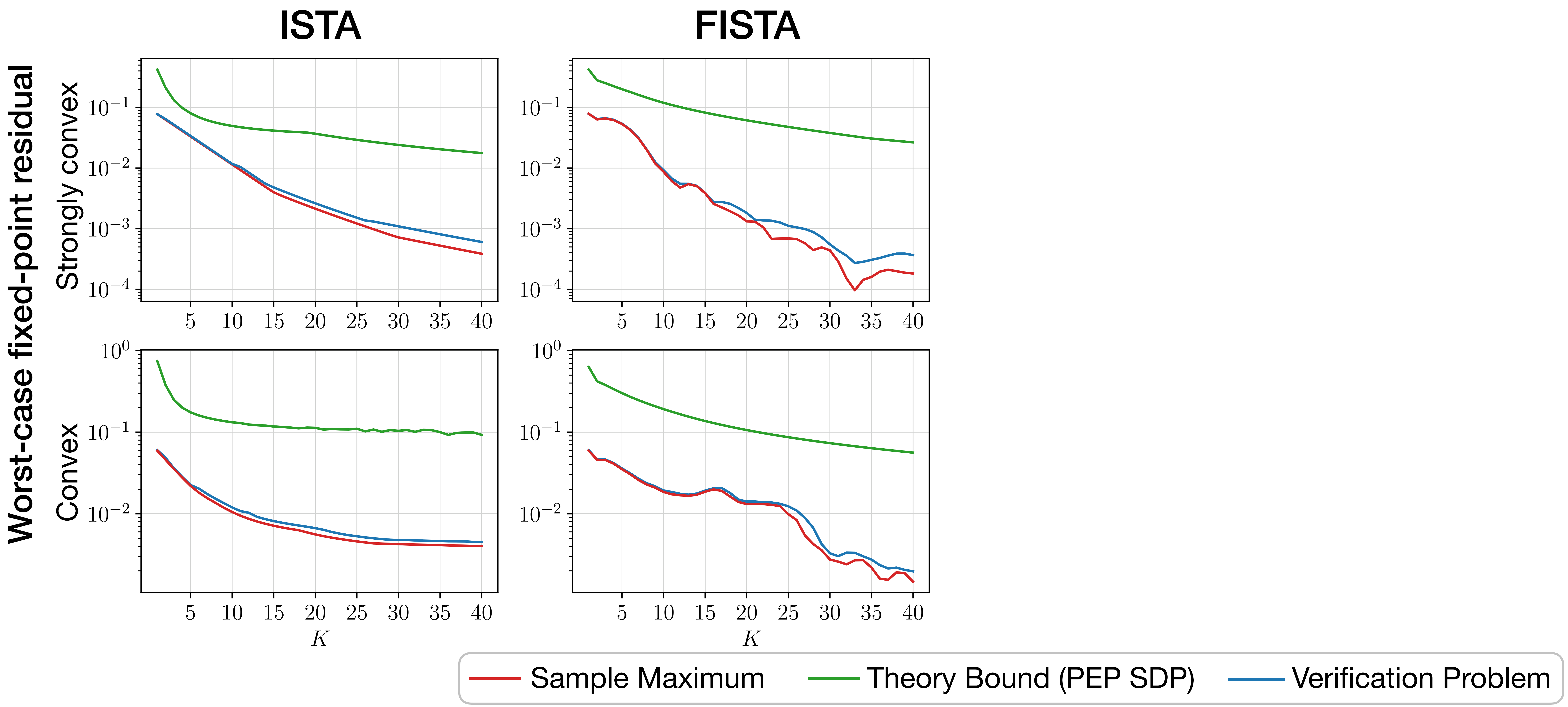
Fast Iterative Soft-Thresholding Algorithm (FISTA)

$$w^{k+1} = \phi_{\lambda\theta}\left((I - \theta D^T D)z^k + \theta D^T x\right)$$

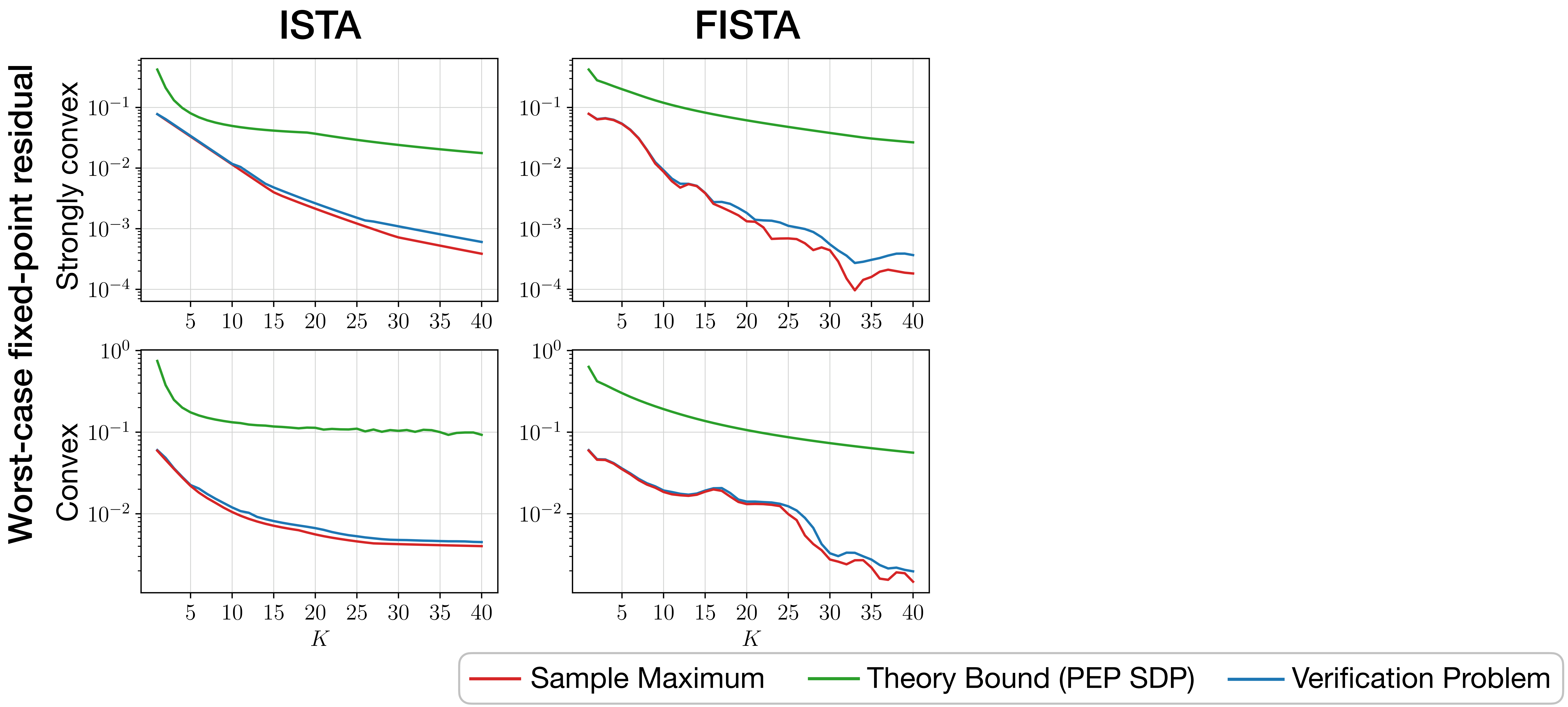
$$z^{k+1} = w^{k+1} + \underbrace{(\beta_k - 1)/\beta_{k+1}}_{\text{momentum}}(w^{k+1} - w^k)$$

soft-thresholding operator

Verification results for sparse coding example

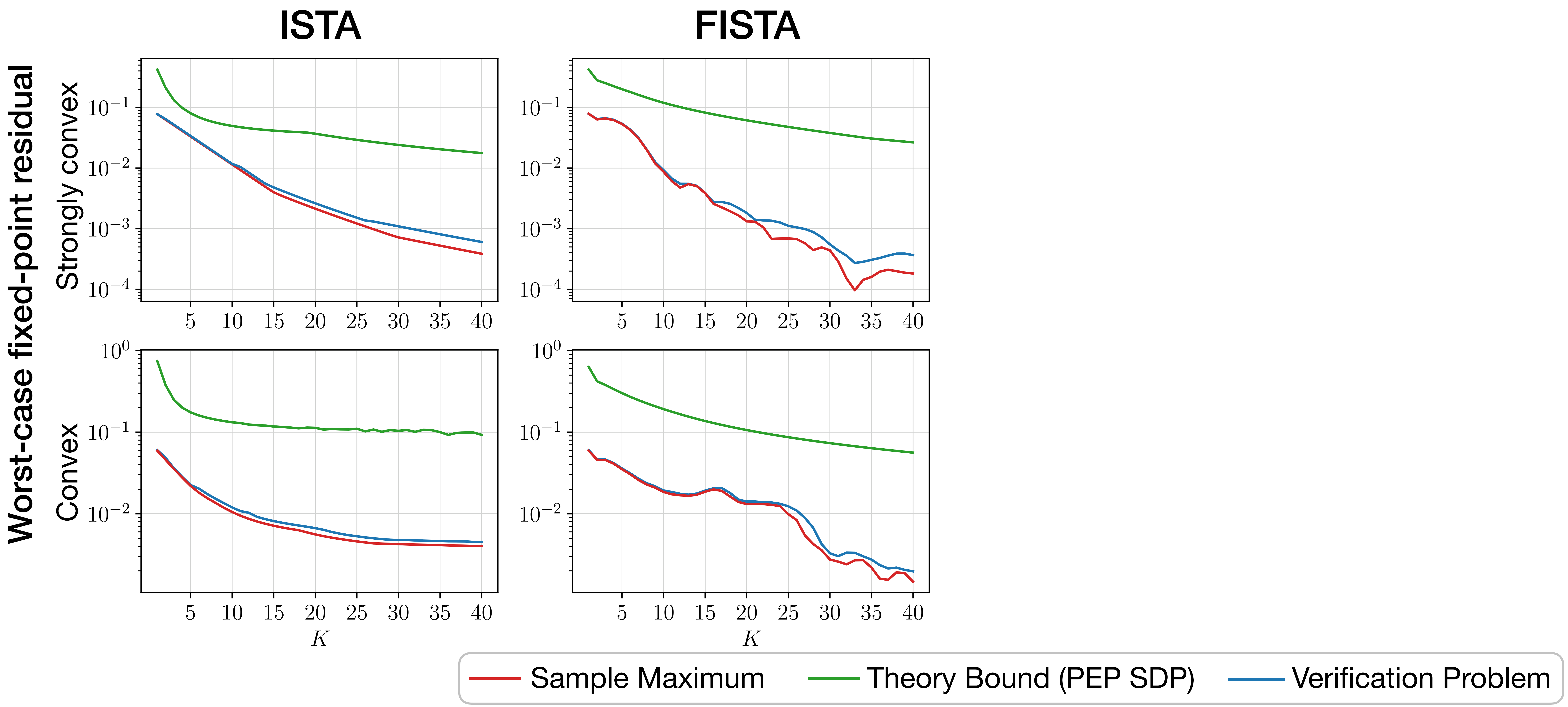


Verification results for sparse coding example



significant reduction in
worst-case fixed-point residual
(exploiting parametric structure)

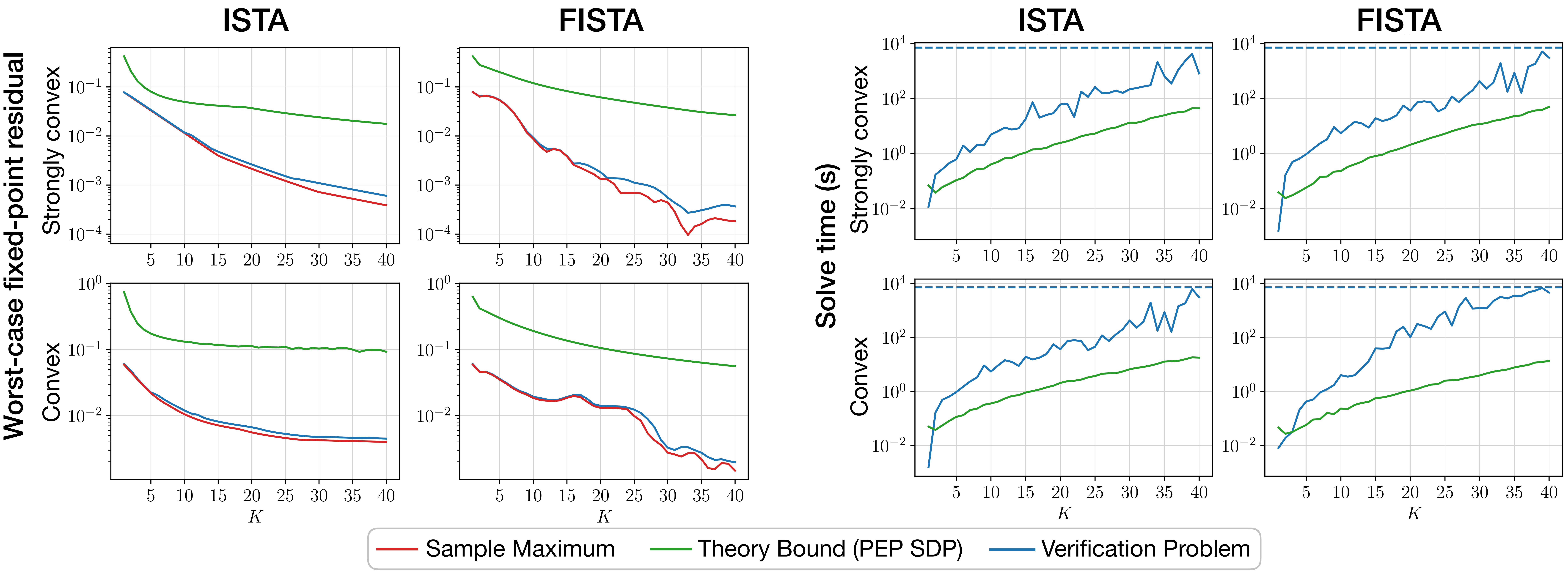
Verification results for sparse coding example



significant reduction in worst-case fixed-point residual *(exploiting parametric structure)*

exactly captures the ripples of the FISTA acceleration

Verification results for sparse coding example



significant reduction in worst-case fixed-point residual *(exploiting parametric structure)*

exactly captures the ripples of the FISTA acceleration

Optimal control

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{t=0}^{T} s_t^T Q s_t + u_t^T R u_t & \\
\text{subject to} & s_{t+1} = A^{\text{dyn}} s_t + B^{\text{dyn}} u_t, & \forall t \\
& s_{\min} \leq s_t \leq s_{\max}, & \forall t \\
& u_{\min} \leq u_t \leq u_{\max}, & \forall t \\
& s_0 = x &
\end{array}
$$

Optimal control

$$
\begin{aligned}
\text{minimize} \quad & \sum_{t=0}^{T} s_t^T Q s_t + u_t^T R u_t \\
\text{subject to} \quad & s_{t+1} = A^{\text{dyn}} s_t + B^{\text{dyn}} u_t, && \forall t \\
& s_{\min} \le s_t \le s_{\max}, && \forall t \\
& u_{\min} \le u_t \le u_{\max}, && \forall t \\
& s_0 = x
\end{aligned}
$$

linear dynamics

Optimal control

optimal sequence of states and controls

minimize $\displaystyle\sum_{t=0}^{T} s_t^T Q s_t + u_t^T R u_t$

subject to

$$
\begin{aligned}
s_{t+1} &= A^{\text{dyn}} s_t + B^{\text{dyn}} u_t, && \forall t \\
s_{\min} &\leq s_t \leq s_{\max}, && \forall t \\
u_{\min} &\leq u_t \leq u_{\max}, && \forall t \\
s_0 &= x
\end{aligned}
$$

linear dynamics

Optimal control

optimal sequence of
states and controls

minimize $$\sum_{t=0}^{T} s_t^T Q s_t + u_t^T R u_t$$

subject to $$s_{t+1} = A^{\text{dyn}} s_t + B^{\text{dyn}} u_t, \quad \forall t$$

$$s_{\min} \leq s_t \leq s_{\max}, \quad \forall t$$

$$u_{\min} \leq u_t \leq u_{\max}, \quad \forall t$$

$$s_0 = x$$

linear
dynamics

initial
state

Optimal control

optimal sequence of states and controls

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{t=0}^{T} s_t^T Q s_t + u_t^T R u_t & \\ \text{subject to} & s_{t+1} = A^{\text{dyn}} s_t + B^{\text{dyn}} u_t, & \forall t \\ & s_{\min} \le s_t \le s_{\max}, & \forall t \\ & u_{\min} \le u_t \le u_{\max}, & \forall t \\ & s_0 = x & \end{array}$$

linear dynamics

initial state

condensed formulation

$$z = (u_1, \ldots, u_T)$$

$$\begin{array}{ll} \text{maximize} & (1/2) z^T P z + q(x)^T z \\ \text{subject to} & l(x) \le M z \le u(x) \end{array}$$

Optimal control

optimal sequence of states and controls

$$\begin{array}{lll} \text{minimize} & \sum_{t=0}^{T} s_t^T Q s_t + u_t^T R u_t & \\ \text{subject to} & s_{t+1} = A^{\text{dyn}} s_t + B^{\text{dyn}} u_t, & \forall t \\ & s_{\min} \leq s_t \leq s_{\max}, & \forall t \\ & u_{\min} \leq u_t \leq u_{\max}, & \forall t \\ & s_0 = x & \end{array}$$

linear dynamics

initial state

condensed formulation

$$z = (u_1, \ldots, u_T)$$

$$\begin{array}{ll} \text{maximize} & (1/2) z^T P z + q(x)^T z \\ \text{subject to} & l(x) \leq Mz \leq u(x) \end{array}$$

OSQP ADMM splitting

$$w^{k+1} = \mathcal{S}_{[l(x), u(x)]}(v^k)$$

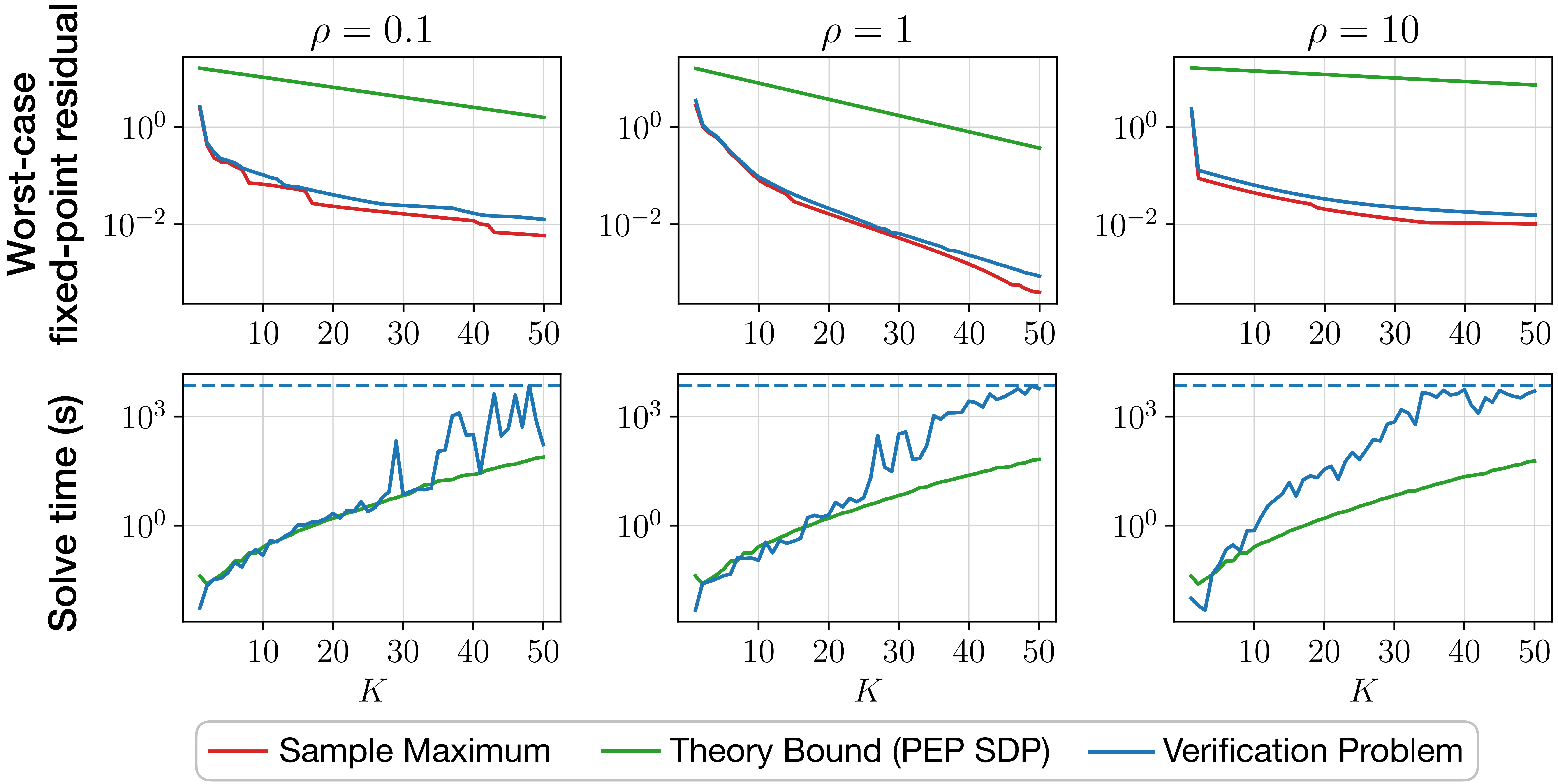
Solve $(P + \sigma I + \rho M^T M) z^{k+1} = \sigma z^k - q(x) + \rho M^T (2w^{k+1} - v^k)$

$$v^{k+1} = v^k + Mz^{k+1} - w^{k+1}$$

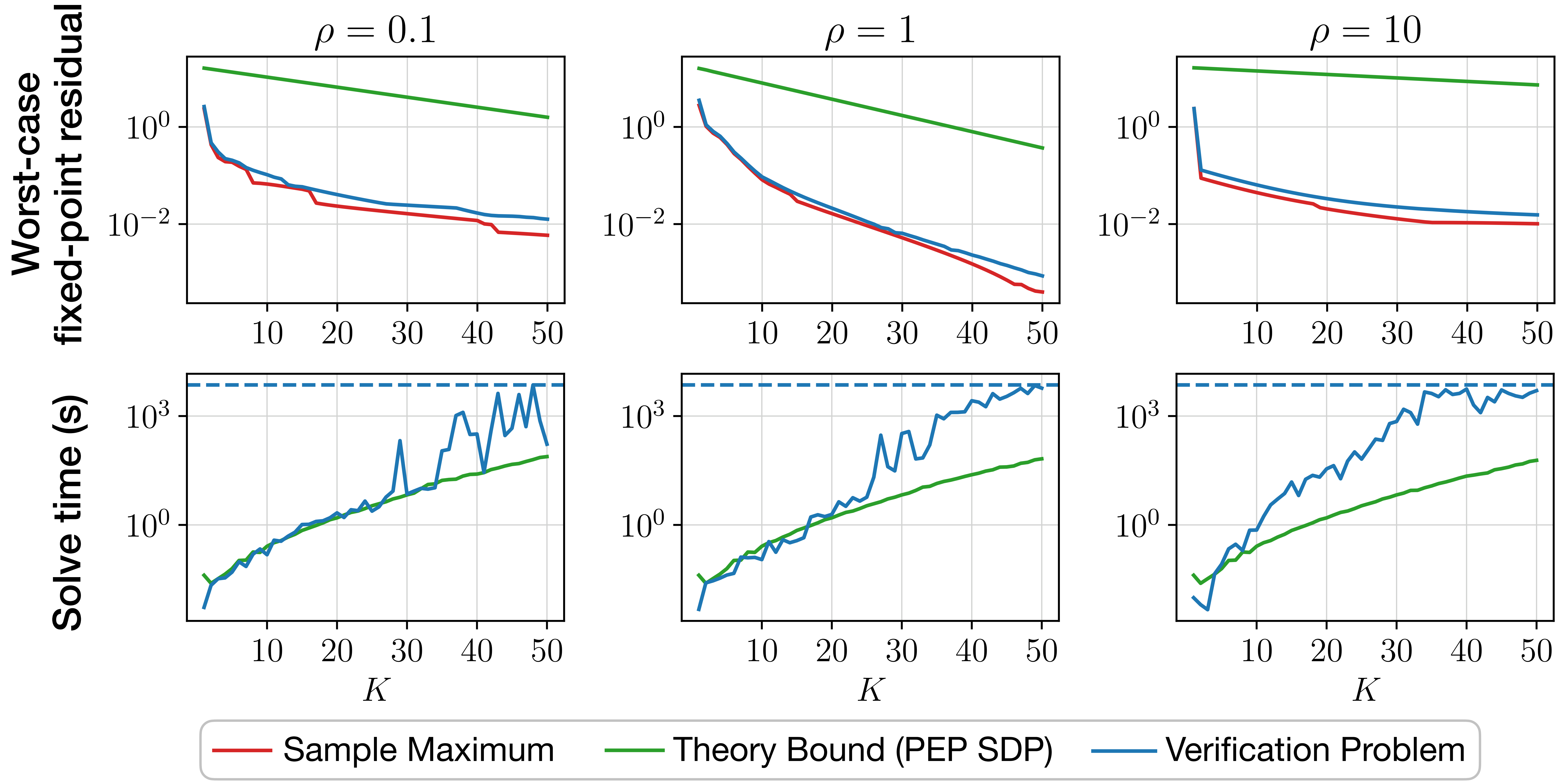
saturated linear unit

linear system

Verification results for optimal control problem

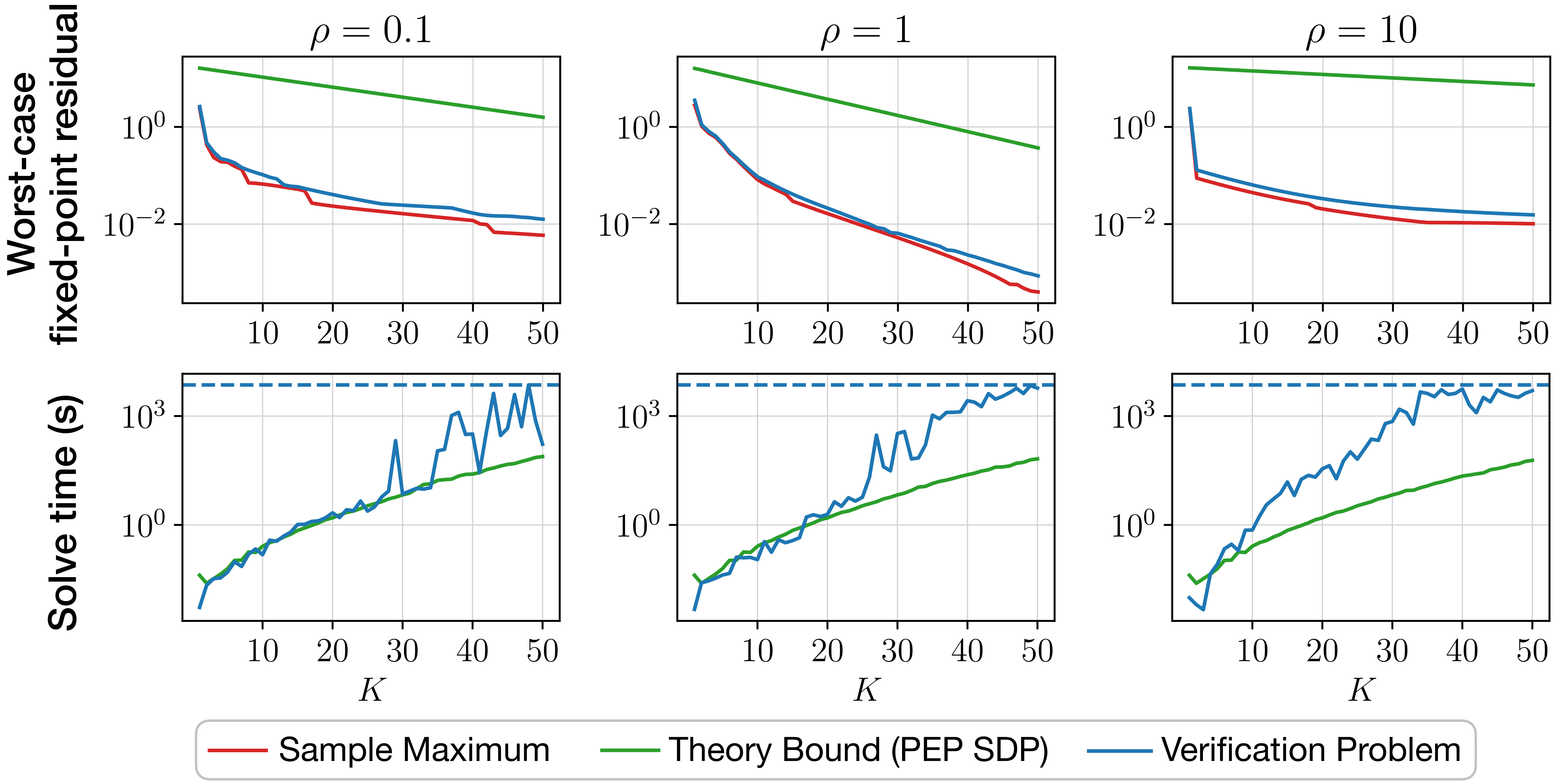


Verification results for optimal control problem



exactly quantifies the iterations required → crucial in real-time applications!

Verification results for optimal control problem



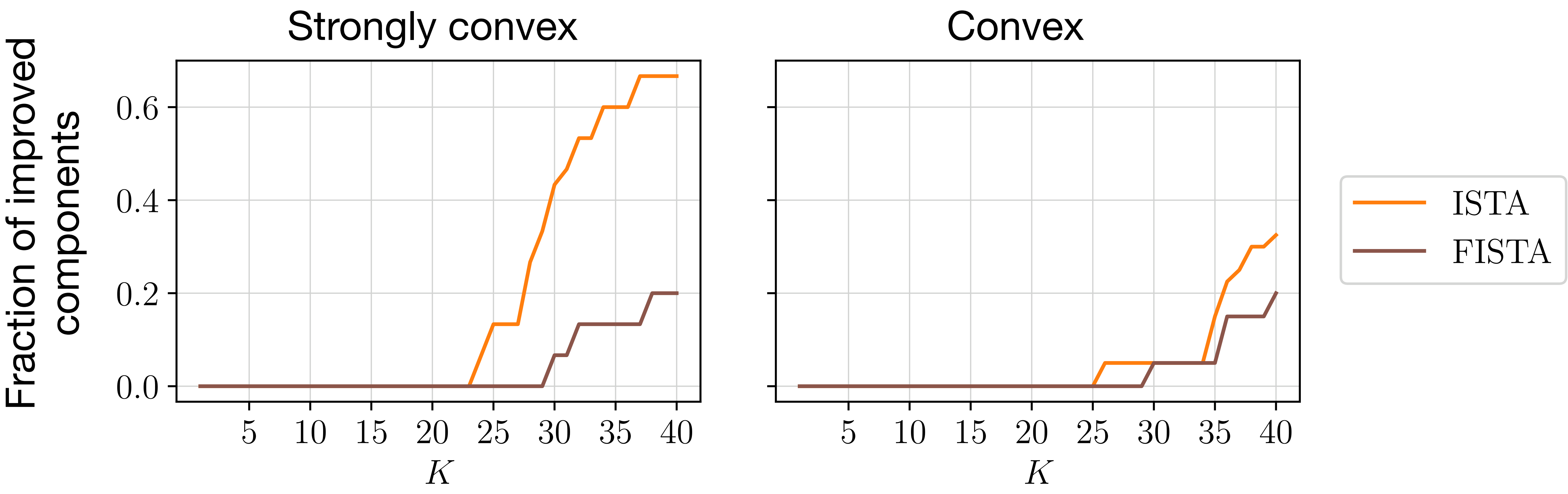
can be used to design algorithms!

exactly quantifies the iterations required

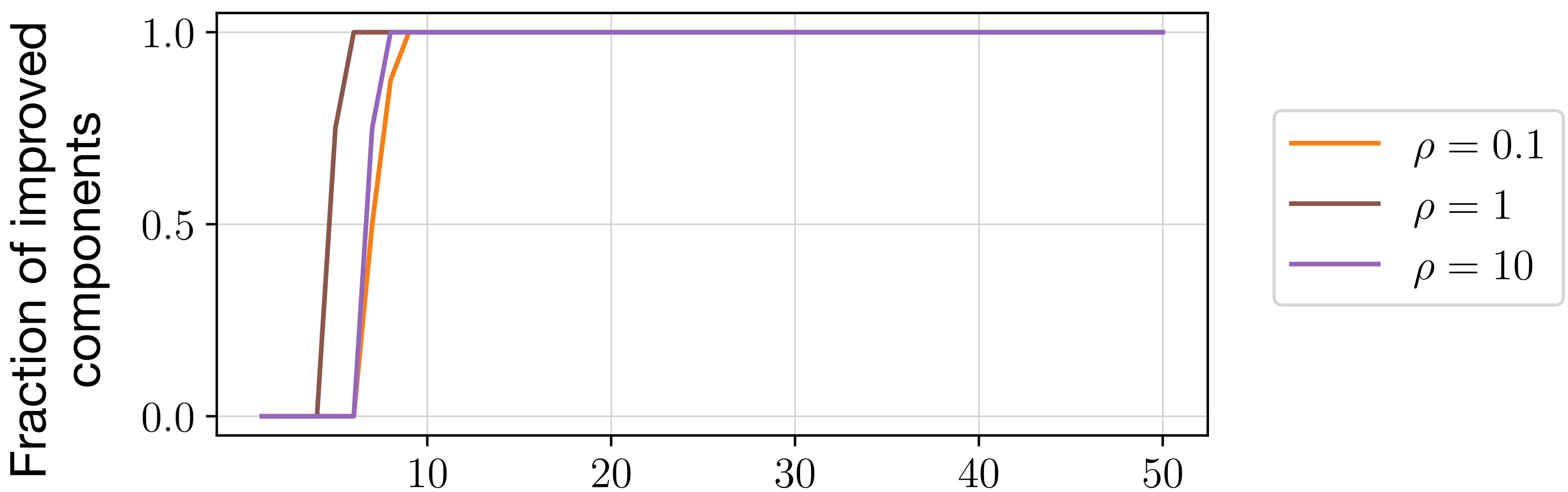
crucial in real-time applications!

Operator theory tightens bounds on the iterates

Sparse coding

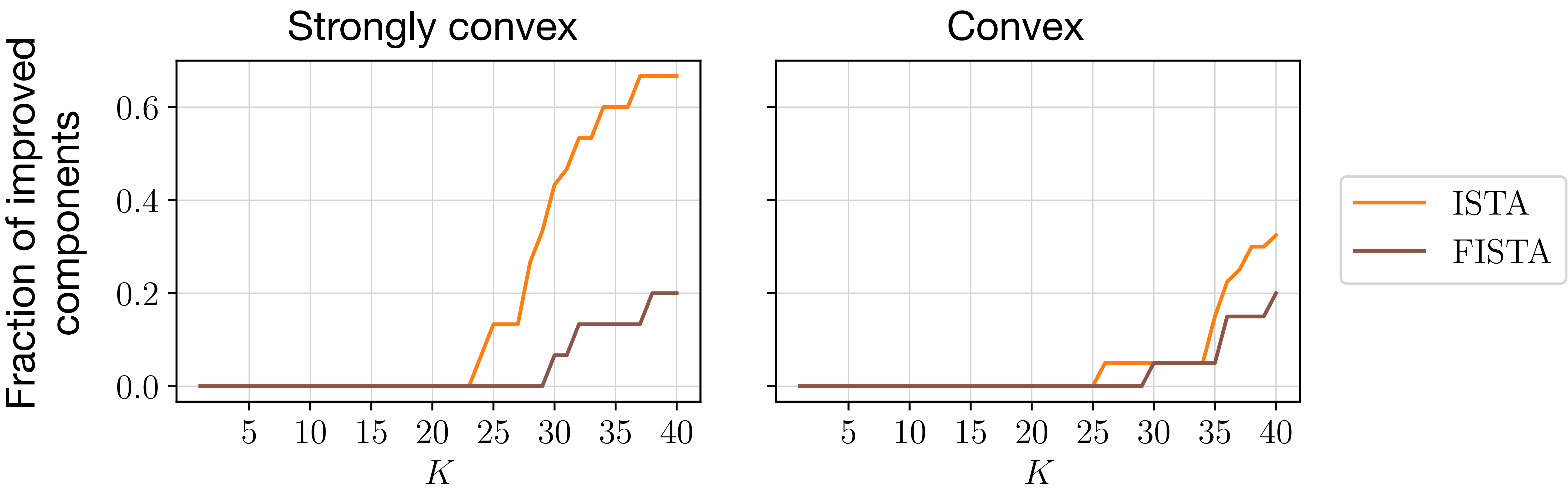


Optimal control

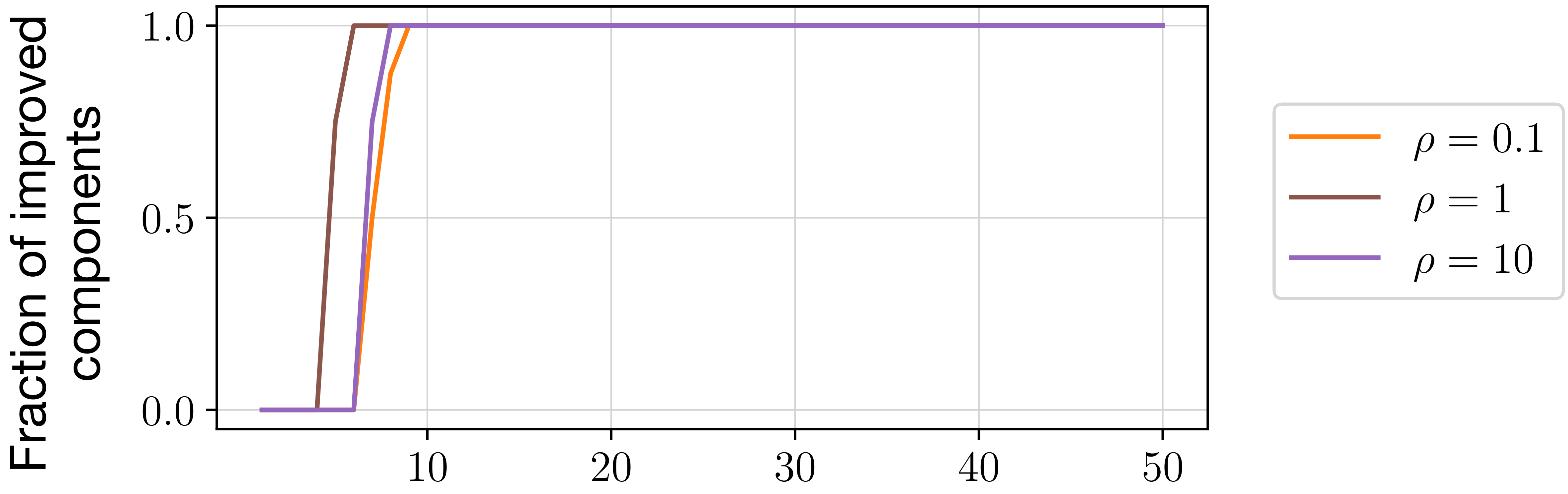


Operator theory tightens bounds on the iterates

Sparse coding



Optimal control



strongly convex problems have the largest benefits
(because of linear convergence)

Analyzing the algorithm performance after K iterations

goal

estimate norm of fixed-point residual

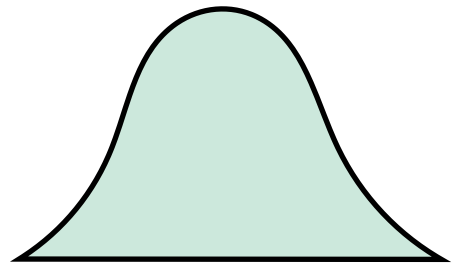
$$r^K(x) = \|z^K - z^{K-1}\|$$

worst-case

$$\max_{x \in \mathcal{X}} r^K(x) \leq \epsilon$$

problem instances

convergence tolerance



probabilistic

$$\mathbf{P}\left(r^K(x) > \epsilon\right) \leq \eta$$

probability bound

problem instances

convergence tolerance

Analyzing the algorithm performance after K iterations

goal

estimate norm of fixed-point residual

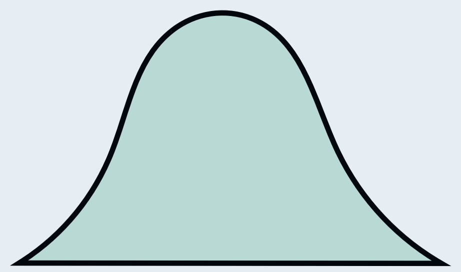
$$r^K(x) = \|z^K - z^{K-1}\|$$

worst-case

$$\max_{\boxed{x \in \mathcal{X}}} r^K(x) \leq \boxed{\epsilon}$$

problem instances

convergence tolerance



probabilistic

$$\boxed{\mathbf{P}}\left(r^K(x) > \boxed{\epsilon}\right) \leq \boxed{\eta}$$

probability bound

problem instances

convergence tolerance

Probabilistic analysis

Performance analysis using statistical learning theory

Goal: *estimate probability of computing bad-quality solutions*

$$\mathbf{P}\left(r^K(x) > \epsilon\right) \leq \eta$$

probability bound

problem instances

convergence tolerance

Performance analysis using statistical learning theory

Goal: *estimate probability of computing bad-quality solutions*

$$\mathbf{P}\left(r^K(x) > \epsilon\right) = \mathop{\mathbf{E}}_{x \sim P}\left(e(x)\right) \longleftarrow \mathbf{1}(r^K(x) > \epsilon)$$

error

Performance analysis using statistical learning theory

Goal: *estimate probability of computing bad-quality solutions*

$$\mathbf{P}\left(r^K(x) > \epsilon\right) = \mathop{\mathbf{E}}_{x \sim P}\left(e(x)\right) \longleftarrow \mathbf{1}(r^K(x) > \epsilon)$$

error

	classification	optimization
input		problem instance *(with parameter x)*

Performance analysis using statistical learning theory

Goal: *estimate probability of computing bad-quality solutions*

$$\mathbf{P}\left(r^K(x) > \epsilon\right) = \underset{x \sim P}{\mathbf{E}}\left(e(x)\right) \longleftarrow \mathbf{1}(r^K(x) > \epsilon)$$

error

	classification	optimization
input		problem instance *(with parameter x)*
hypothesis	cat	residual $r^K(x)$

Performance analysis using statistical learning theory

Goal: *estimate probability of computing bad-quality solutions*

$$\mathbf{P}\left(r^K(x) > \epsilon\right) = \underset{x\sim P}{\mathbf{E}}\left(e(x)\right) \longleftarrow \mathbf{1}(r^K(x) > \epsilon)$$

error

	classification	optimization
input		problem instance *(with parameter x)*
hypothesis	cat	residual $r^K(x)$
error	0 (1 if wrong)	$e(x) = \mathbf{1}(r^K(x) > \epsilon)$

Performance analysis using statistical learning theory

Goal: *estimate probability of computing bad-quality solutions*

$$\mathbf{P}\left(r^K(x) > \epsilon\right) = \mathop{\mathbf{E}}_{x\sim P}\left(e(x)\right) \longleftarrow \overset{\text{error}}{\mathbf{1}(r^K(x) > \epsilon)}$$

	classification	optimization
input		problem instance *(with parameter x)*
hypothesis	cat	residual $r^K(x)$
error	0 (1 if wrong)	$e(x) = \mathbf{1}(r^K(x) > \epsilon)$
guarantees	expected loss on **new data**	expected loss on **new problem instances**

Our recipe to bound performance

Goal: *estimate probability of computing bad-quality solutions*

$$\mathbf{P}\left(r^K(x) > \epsilon\right) = \mathop{\mathbf{E}}_{x\sim P}\left(e(x)\right) \longleftarrow \underbrace{\mathbf{1}(r^K(x) > \epsilon)}_{}$$

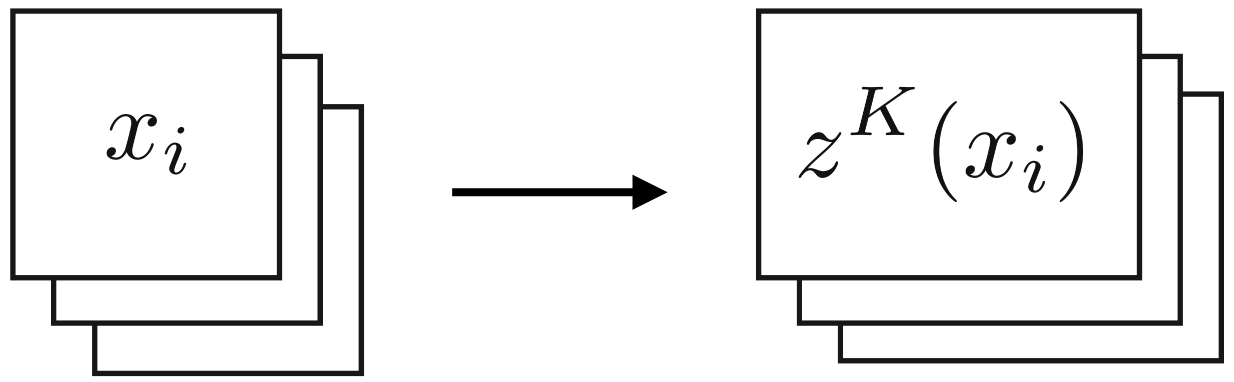
error

step 1

run K steps
for N parametric instances

instances

candidate solutions



Our recipe to bound performance

Goal: *estimate probability of computing bad-quality solutions*

$$\mathbf{P}\left(r^K(x) > \epsilon\right) = \mathop{\mathbf{E}}_{x \sim P}\left(e(x)\right) \longleftarrow \mathbf{1}(r^K(x) > \epsilon)$$

error

step 1

run K steps

for N parametric instances

instances

x_i $\longrightarrow$ $z^K(x_i)$

candidate solutions

$\longrightarrow$

step 2

compute empirical risk

$$\frac{1}{N}\sum_{i=1}^{N} e(x_i)$$

Our recipe to bound performance

Goal: *estimate probability of computing bad-quality solutions*

$$\mathbf{P}\left(r^K(x) > \epsilon\right) = \mathop{\mathbf{E}}_{x \sim P}\left(e(x)\right) \longleftarrow \mathbf{1}(r^K(x) > \epsilon)$$

error

step 1

run K steps

for N parametric instances

instances x_i $\longrightarrow$ **candidate solutions** $z^K(x_i)$

step 2

compute empirical risk

$$\frac{1}{N}\sum_{i=1}^{N} e(x_i)$$

step 3

bound true risk

(next slide)

$\mathop{\mathbf{E}}_{x \sim P}(e(x)) \leq$ bound

Statistical learning gives us probabilistic guarantees

sample convergence bound

with probability $1 - \delta$

$$\mathbf{P}(r^K(x) > \epsilon) = \underbrace{\mathop{\mathbf{E}}_{x \sim P}(e(x))}_{\text{true risk}} \leq \underbrace{\mathrm{kl}^{-1}}_{\text{inverse kl divergence}} \left(\underbrace{\frac{1}{N} \sum_{i=1}^{N} e(x_i)}_{\text{empirical risk}} \,\middle|\, \underbrace{\frac{\log(2/\delta)}{N}}_{\text{regularizer}} \right)$$

number of instances

true risk

inverse kl divergence *(1D convex problem)*

empirical risk

regularizer

Langford (2001)

Statistical learning gives us probabilistic guarantees

sample convergence bound

with probability $1-\delta$

$$\mathbf{P}(r^K(x) > \epsilon) = \underbrace{\mathop{\mathbf{E}}_{x\sim P}(e(x))}_{\text{true risk}} \leq \underbrace{\mathrm{kl}^{-1}}_{\text{inverse kl divergence (1D convex problem)}} \left(\underbrace{\frac{1}{N}\sum_{i=1}^{N} e(x_i)}_{\text{empirical risk}} \;\middle|\; \underbrace{\frac{\log(2/\delta)}{N}}_{\text{regularizer}} \right)$$

number of instances

interpretation of bound equal to B

With probability $1-\delta$, the fixed-point residual is above ϵ after K steps

B fraction of times

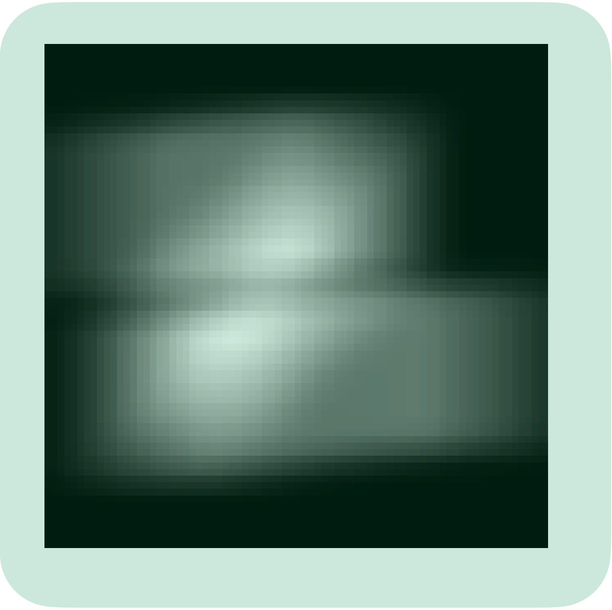
Langford (2001)

Success rates for OSQP in image deblurring

$$\begin{array}{ll} \text{minimize} & \|Az - x\|_2^2 + \lambda \|z\|_1 \\ \text{subject to} & 0 \leq z \leq 1 \end{array}$$

blurred image

deblurred image



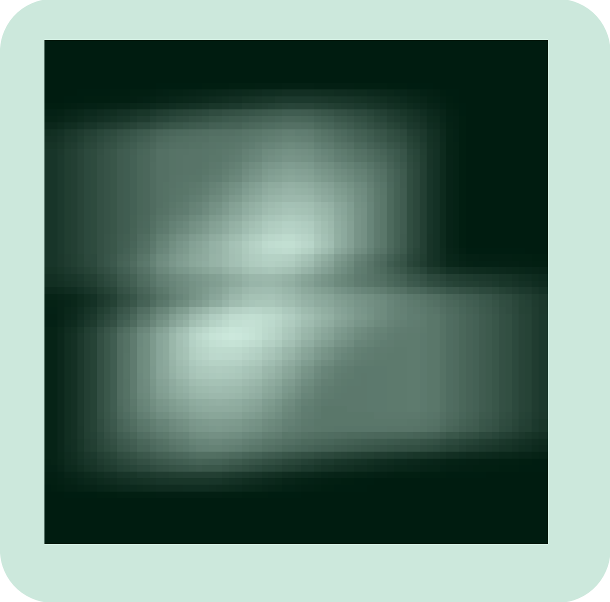
Solve with OSQP splitting

Success rates for OSQP in image deblurring

$$\begin{array}{ll} \text{minimize} & \|Az - x\|_2^2 + \lambda\|z\|_1 \\ \text{subject to} & 0 \leq z \leq 1 \end{array}$$

blurred image

deblurred image



Solve with OSQP splitting

fraction of problems solved

$$1 - \mathop{\mathbf{E}}_{x \sim P}(e(x))$$

error

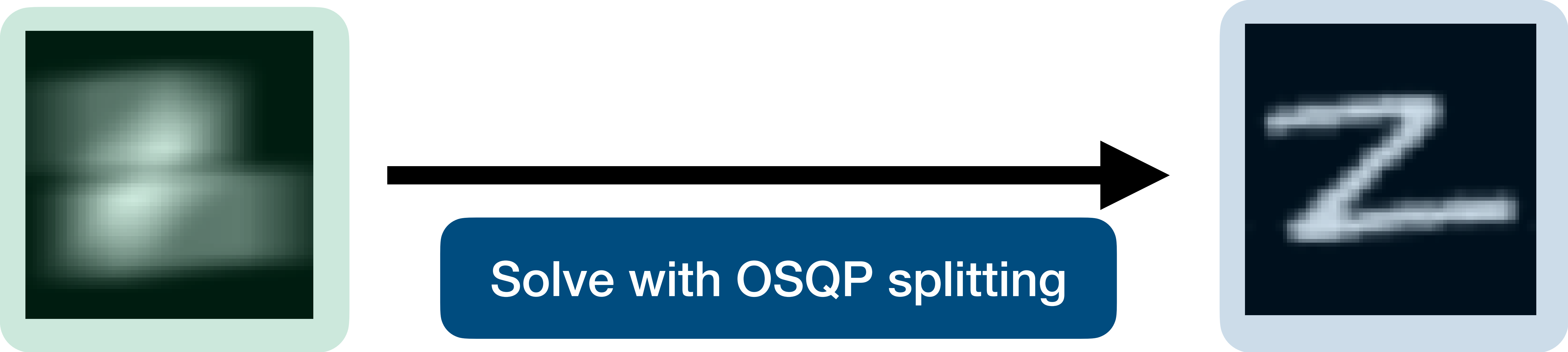
$$\mathbf{1}(r^K(x) > \epsilon)$$

Success rates for OSQP in image deblurring

$$\begin{aligned} \text{minimize} \quad & \|Az - x\|_2^2 + \lambda \|z\|_1 \\ \text{subject to} \quad & 0 \leq z \leq 1 \end{aligned}$$

blurred image

deblurred image

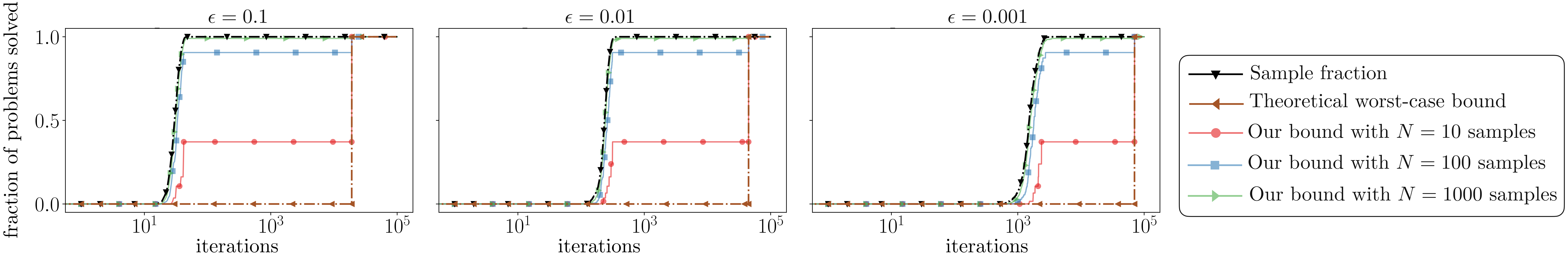


fraction of problems solved

$$1 - \mathbf{E}_{x \sim P}(e(x))$$

error

$$\mathbf{1}(r^K(x) > \epsilon)$$

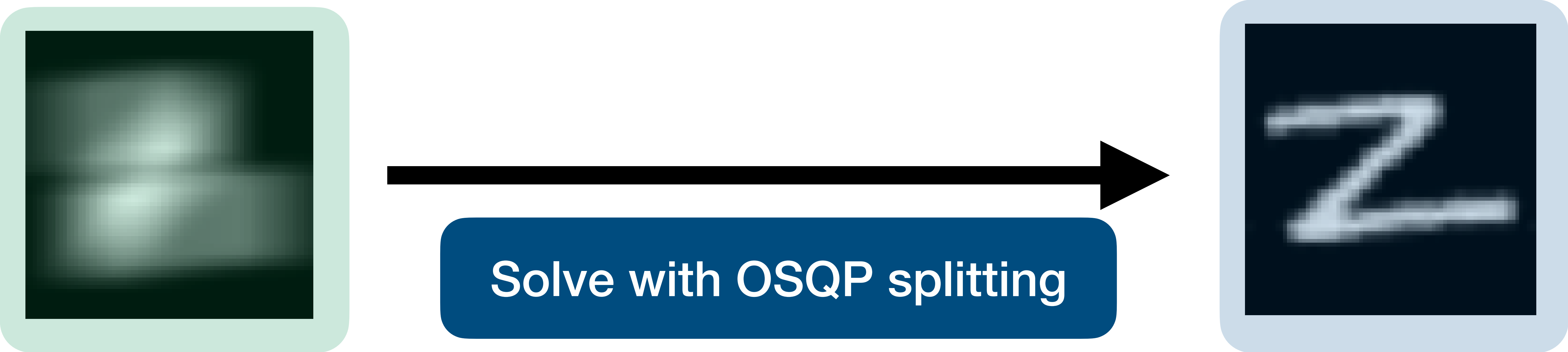


Success rates for OSQP in image deblurring

$$\begin{array}{ll} \text{minimize} & \|Az - x\|_2^2 + \lambda \|z\|_1 \\ \text{subject to} & 0 \leq z \leq 1 \end{array}$$

blurred image

deblurred image



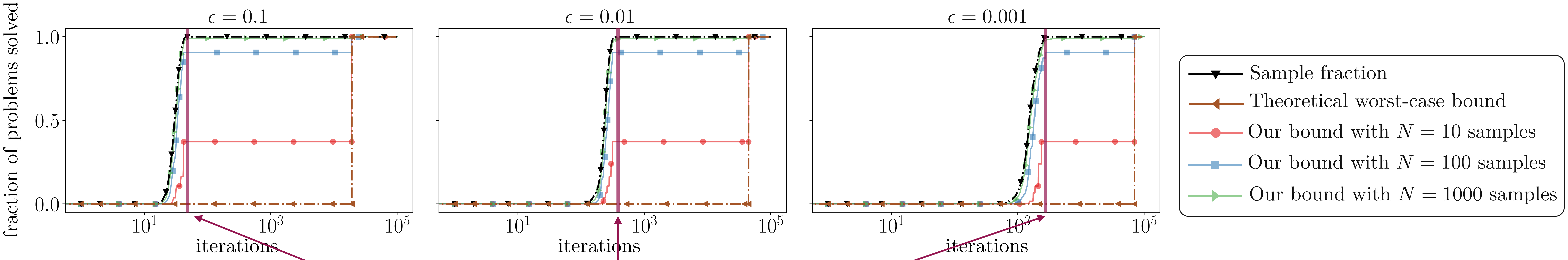
Solve with OSQP splitting

fraction of problems solved

$$1 - \mathop{\mathbf{E}}_{x \sim P} (e(x))$$

error

$$\mathbf{1}(r^K(x) > \epsilon)$$



iterations required
to solve all test instances

Conclusions

Acknowledgements

Vinit Ranjan
Princeton

Jisun Park
Princeton

Rajiv Sambharya
UPenn

Andrea Lodi
CornellTech

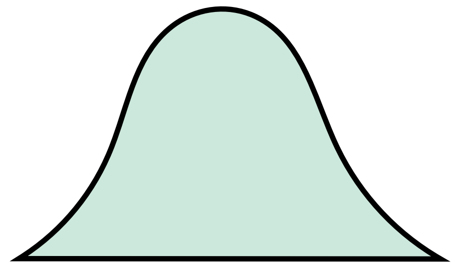
Stefano Gualandi
Univ. of Pavia

Papers

worst-case

Exact Verification of First-Order Methods via Mixed-Integer Linear Programming
V. Ranjan, J. Park, S. Gualandi, A. Lodi, and B. Stellato
arXiv e-prints:2412.11330 (2025)
github.com/stellatogrp/mip_algo_verify

Verification of First-Order Methods for Parametric Quadratic Optimization
V. Ranjan and B. Stellato
arXiv e-prints:2403.03331 (2025)
github.com/stellatogrp/sdp_algo_verify



probabilistic

Data-Driven Performance Guarantees for Classical and Learned Optimizers
R. Sambharya and B. Stellato
arxiv.org: 2404.13831 (2024)
github.com/stellatogrp/data_driven_optimizer_guarantees

Less Pessimistic Analysis of First-order Methods

1. parametric structure matters

Less Pessimistic Analysis of First-order Methods

1. parametric structure matters

2. *a-posteriori* information can reduce pessimism

Less Pessimistic Analysis of First-order Methods

1. **parametric** structure matters

2. ***a-posteriori* information** can reduce pessimism

3. useful to **design new algorithms**

traditional view

- general-purpose
- one-size-fits all

→

new view

- task-specific
- tunable/trainable
- deployable anywhere

Less Pessimistic Analysis of First-order Methods

1. **parametric** structure matters

2. ***a-posteriori* information** can reduce pessimism

3. useful to **design new algorithms**

traditional view

- general-purpose
- one-size-fits all

→

new view

- task-specific
- tunable/trainable
- deployable anywhere

@stellato.io

bstellato@princeton.edu

stellato.io

Backup

SDP relaxation for nonnegative least-squares verification

nonnegative least squares

$$\begin{array}{ll} \text{minimize} & (1/2)\|Az - x\|_2^2 \\ \text{subject to} & z \geq 0 \end{array}$$

parameters

verification problem

projected gradient descent

$$\begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\| \\ \text{subject to} & z^{k+1} = \max\{(I - \theta A^T A)z^k + \theta(A^T x), 0\}, \quad k = 0, \dots, K-1 \\ & z^0 = \{0\}, \quad x \in \mathcal{X} = \{x \mid \|x - 30 \cdot \mathbf{1}\| \leq 0.5\} \end{array}$$

Verification of First-Order Methods for Parametric Quadratic Optimization
V. Ranjan and B. Stellato
arXiv e-prints:2403.03331 (2024)
github.com/stellatogrp/algorithm_verification

SDP relaxation for nonnegative least-squares verification

nonnegative least squares

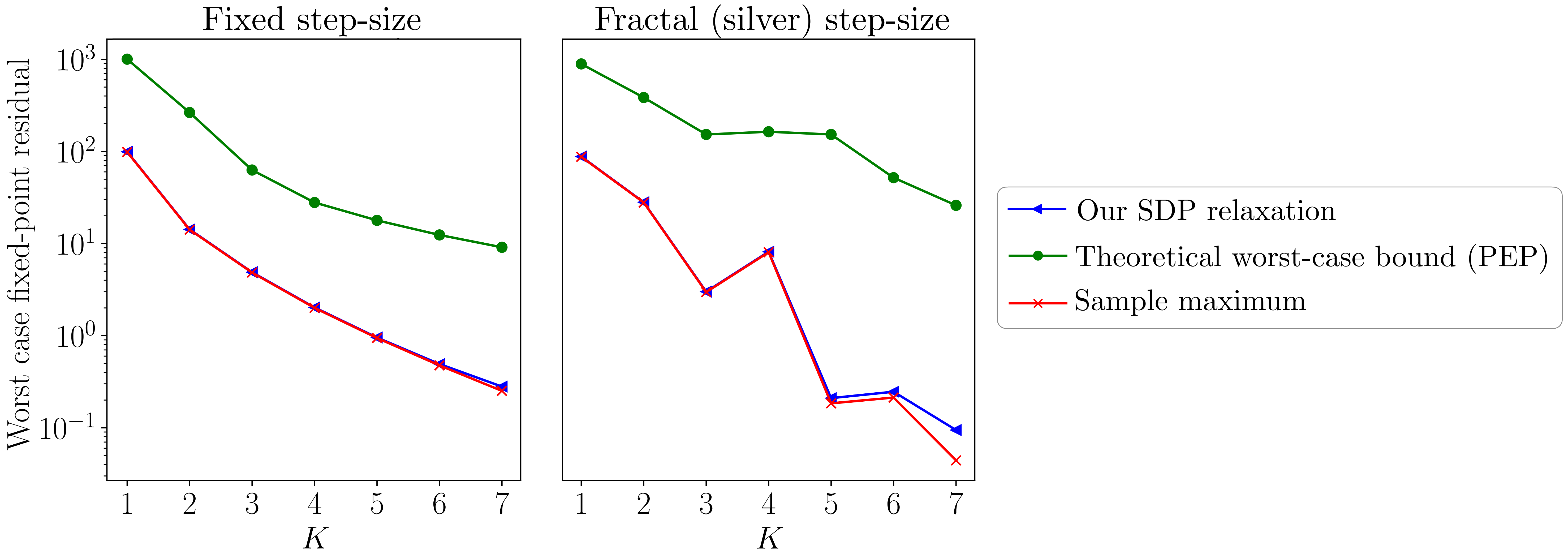
$$\begin{array}{ll} \text{minimize} & (1/2)\|Az - x\|_2^2 \\ \text{subject to} & z \geq 0 \end{array}$$

parameters

verification problem

projected gradient descent

$$\begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\| \\ \text{subject to} & z^{k+1} = \max\{(I - \theta A^T A)z^k + \theta(A^T x), 0\}, \quad k = 0, \dots, K-1 \\ & z^0 = \{0\}, \quad x \in \mathcal{X} = \{x \mid \|x - 30 \cdot \mathbf{1}\| \leq 0.5\} \end{array}$$



[Verification of First-Order Methods for Parametric Quadratic Optimization]
V. Ranjan and B. Stellato
arXiv e-prints:2403.03331 (2024)
github.com/stellatogrp/algorithm_verification

SDP relaxation for nonnegative least-squares verification

nonnegative least squares

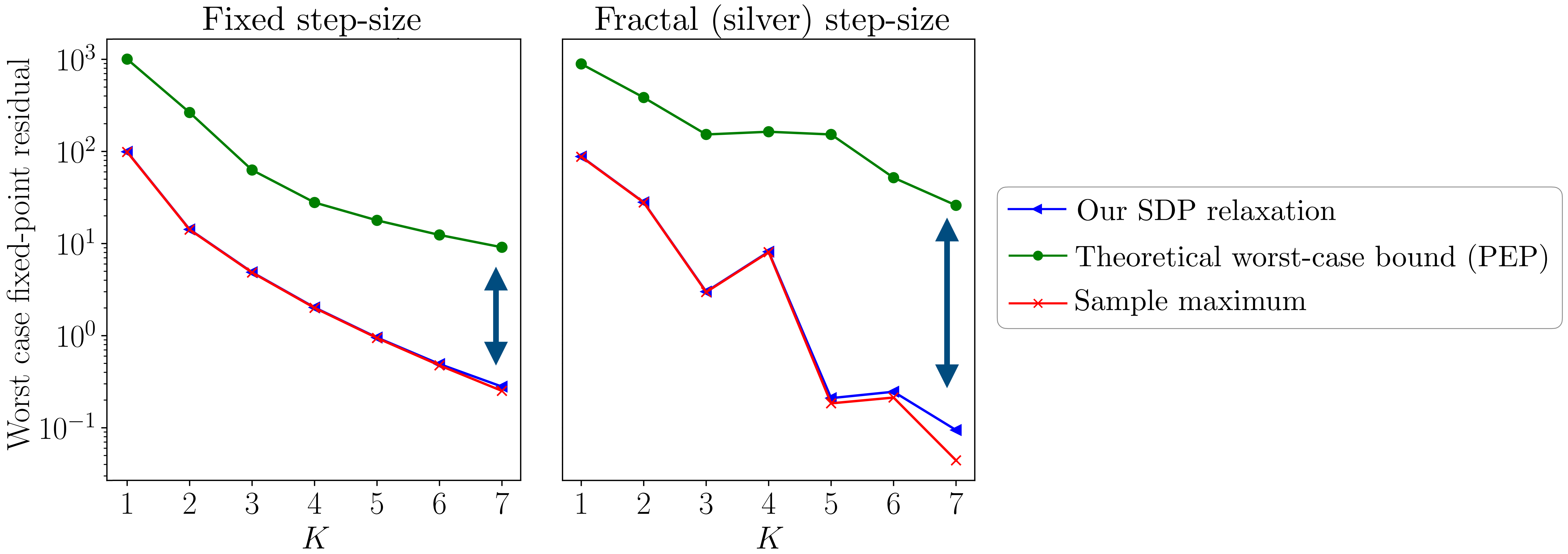
$$\begin{array}{ll} \text{minimize} & (1/2)\|Az - \boxed{x}\|_2^2 \\ \text{subject to} & z \geq 0 \end{array}$$

parameters

verification problem

$$\begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\| \\ \text{subject to} & z^{k+1} = \boxed{\max\{(I - \theta A^T A)z^k + \theta(A^T x), 0\}}, \quad k = 0, \dots, K-1 \\ & z^0 = \{0\}, \quad x \in \mathcal{X} = \{x \mid \|x - 30 \cdot \mathbf{1}\| \leq 0.5\} \end{array}$$

projected gradient descent



significant reduction
(exploiting parametric structure)

Verification of First-Order Methods for Parametric Quadratic Optimization
V. Ranjan and B. Stellato
arXiv e-prints:2403.03331 (2024)
github.com/stellatogrp/algorithm_verification

SDP relaxation for nonnegative least-squares verification

nonnegative least squares

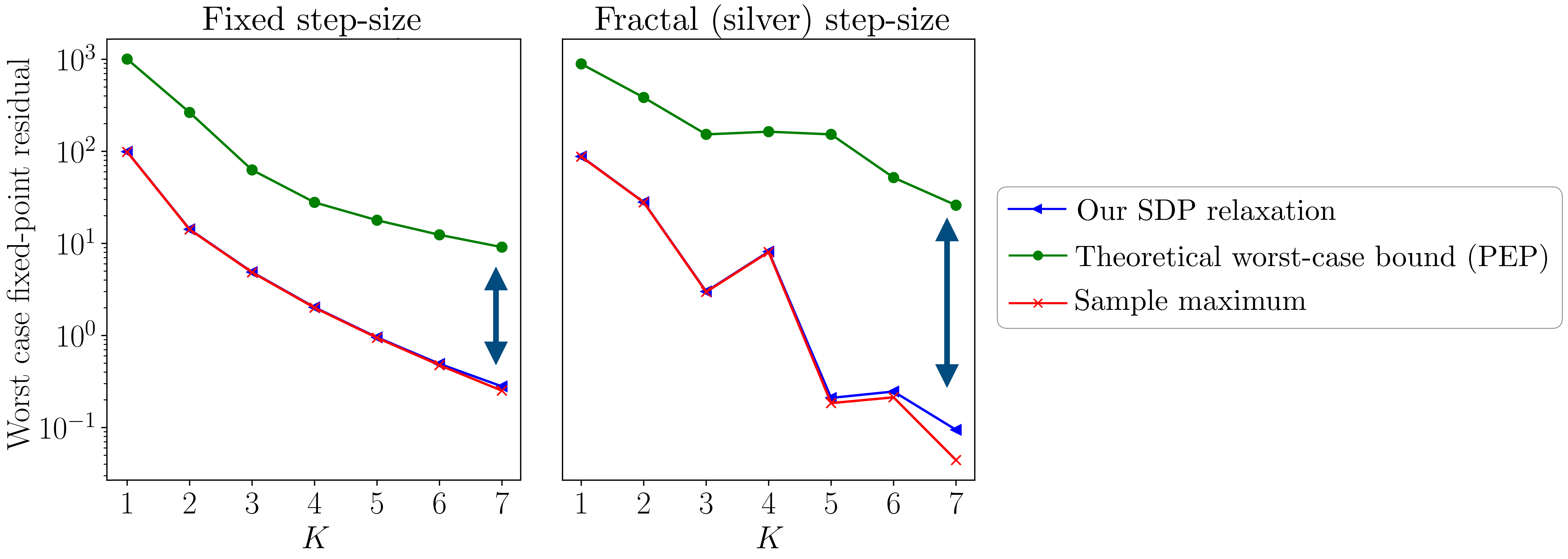
$$\begin{array}{ll} \text{minimize} & (1/2)\|Az - x\|_2^2 \\ \text{subject to} & z \geq 0 \end{array}$$

parameters

verification problem

projected gradient descent

$$\begin{array}{ll} \text{maximize} & \|z^K - z^{K-1}\| \\ \text{subject to} & z^{k+1} = \max\{(I - \theta A^T A)z^k + \theta(A^T x), 0\}, \quad k = 0, \dots, K-1 \\ & z^0 = \{0\}, \quad x \in \mathcal{X} = \{x \mid \|x - 30 \cdot \mathbf{1}\| \leq 0.5\} \end{array}$$



significant reduction
(exploiting parametric structure)

computationally more expensive than PEP
(up to 1000 seconds for these instances)

Verification of First-Order Methods for Parametric Quadratic Optimization
V. Ranjan and B. Stellato
arXiv e-prints:2403.03331 (2024)
github.com/stellatogrp/algorithm_verification

Objective of verification problem as MIP

$$\|s^K - s^{K-1}\|_\infty = \|t\|_\infty = \delta_K$$

- lower bounds $\underline{s}^{K-1}, \underline{s}^K$
- upper bounds, $\bar{s}^{K-1}$ and $\bar{s}^K$

$\longrightarrow$

- lower bound $\underline{t} = \underline{s}^K - \bar{s}^{K-1}$
- upper bound $\bar{t} = \bar{s}^K - \underline{s}^{K-1}$

exact reformulation

$$t = t^+ - t^-, \quad t^+ \leq \bar{t} \odot w, \quad t^- \leq -\underline{t} \odot (\mathbf{1} - w)$$
$$t^+ + t^- \leq \delta_K \leq t^+ + t^- + \max\{\bar{t}, -\underline{t}\} \odot (\mathbf{1} - \gamma)$$
$$\mathbf{1}^T \gamma = 1, \quad t^+ \geq 0, \quad t^- \geq 0$$

$w \in \{0,1\}^n$ (absolute values of the components of t)
$\gamma \in \{0,1\}^d$ (maximum inside the ℓ_∞-norm)

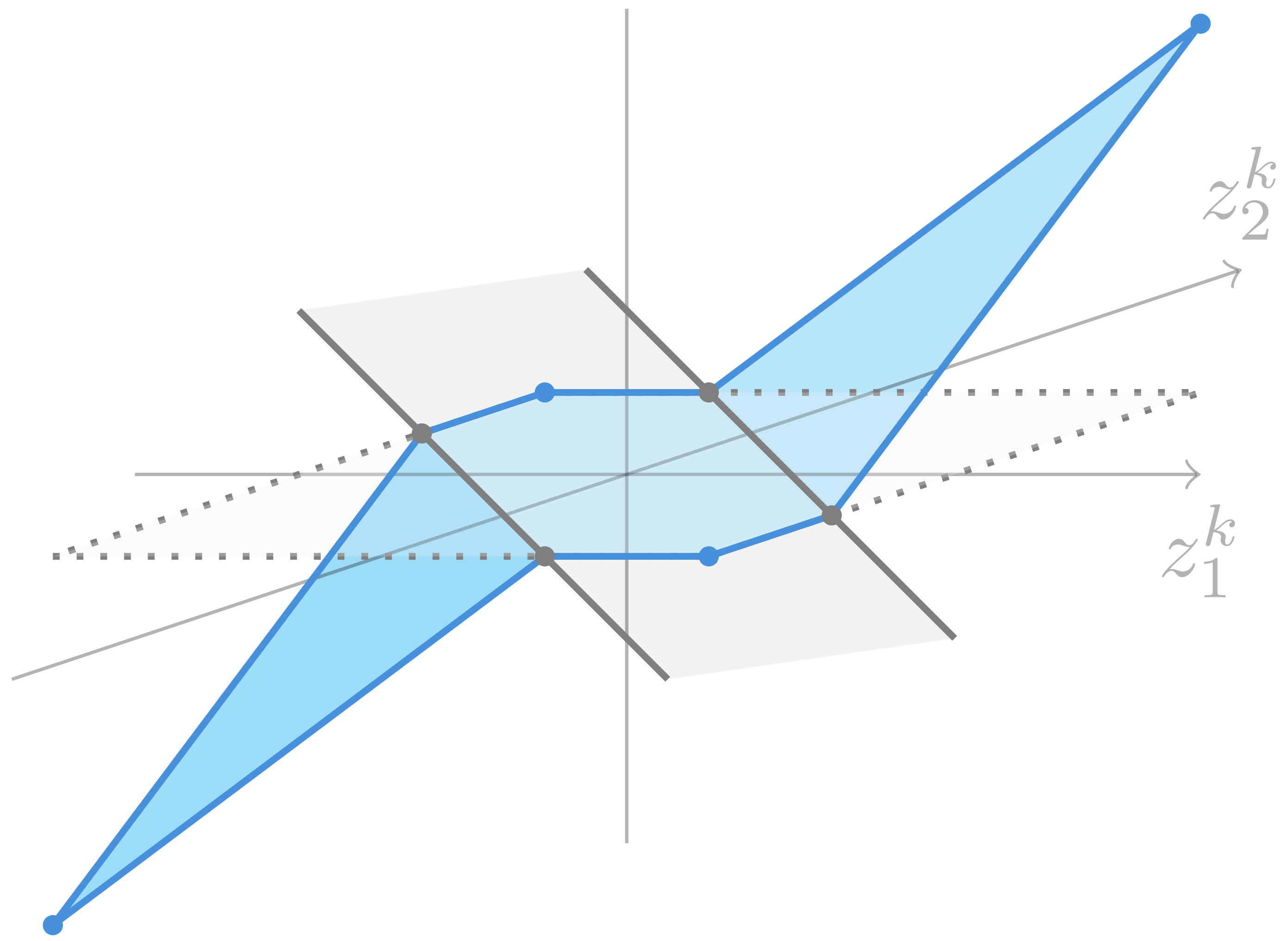
Soft-thresholding operator

$$w = \phi_\lambda(a^T z) = \begin{cases} a^T z - \lambda & a^T z > \lambda \\ 0 & |a^T z| \leq \lambda \\ a^T z + \lambda & a^T z < -\lambda \end{cases}$$

region

$$\Phi = \{(z, w) \in [\underline{z}, \bar{z}] \times \mathbf{R} \mid w = \phi_\lambda(a^T z)\}$$

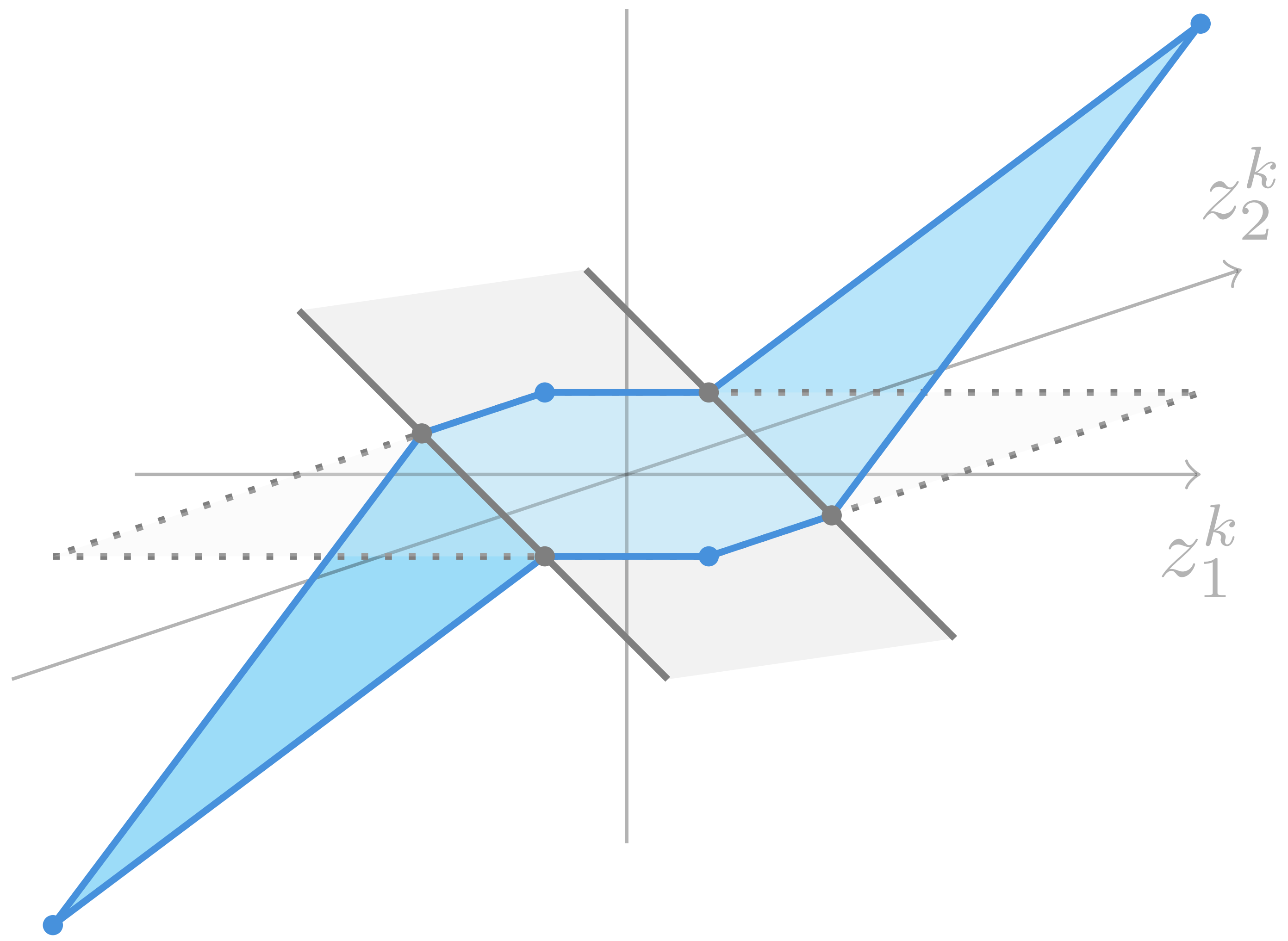
example: $\phi_\lambda(z_1^k + z_2^k)$



Soft-thresholding operator

$$w = \phi_\lambda(a^T z) = \begin{cases} a^T z - \lambda & a^T z > \lambda \\ 0 & |a^T z| \leq \lambda \\ a^T z + \lambda & a^T z < -\lambda \end{cases}$$

example: $\phi_\lambda(z_1^k + z_2^k)$



region

$$\Phi = \{(z, w) \in [\underline{z}, \bar{z}] \times \mathbf{R} \mid w = \phi_\lambda(a^T z)\}$$

lower and upper bounds (needed for convex hull)

$$\ell_I = \sum_{i \in I} a_i \ell_i^0 + \sum_{i \notin I} a_i u_i^0$$

$$u_I = \sum_{i \in I} a_i u_i^0 + \sum_{i \notin I} a_i \ell_i^0$$

$$u_i^0 = \begin{cases} \bar{z}_i & a_i \geq 0 \\ \underline{z}_i & \text{otherwise} \end{cases} \qquad \ell_i^0 = \begin{cases} \underline{z}_i & a_i \geq 0 \\ \bar{z}_i & \text{otherwise} \end{cases}$$

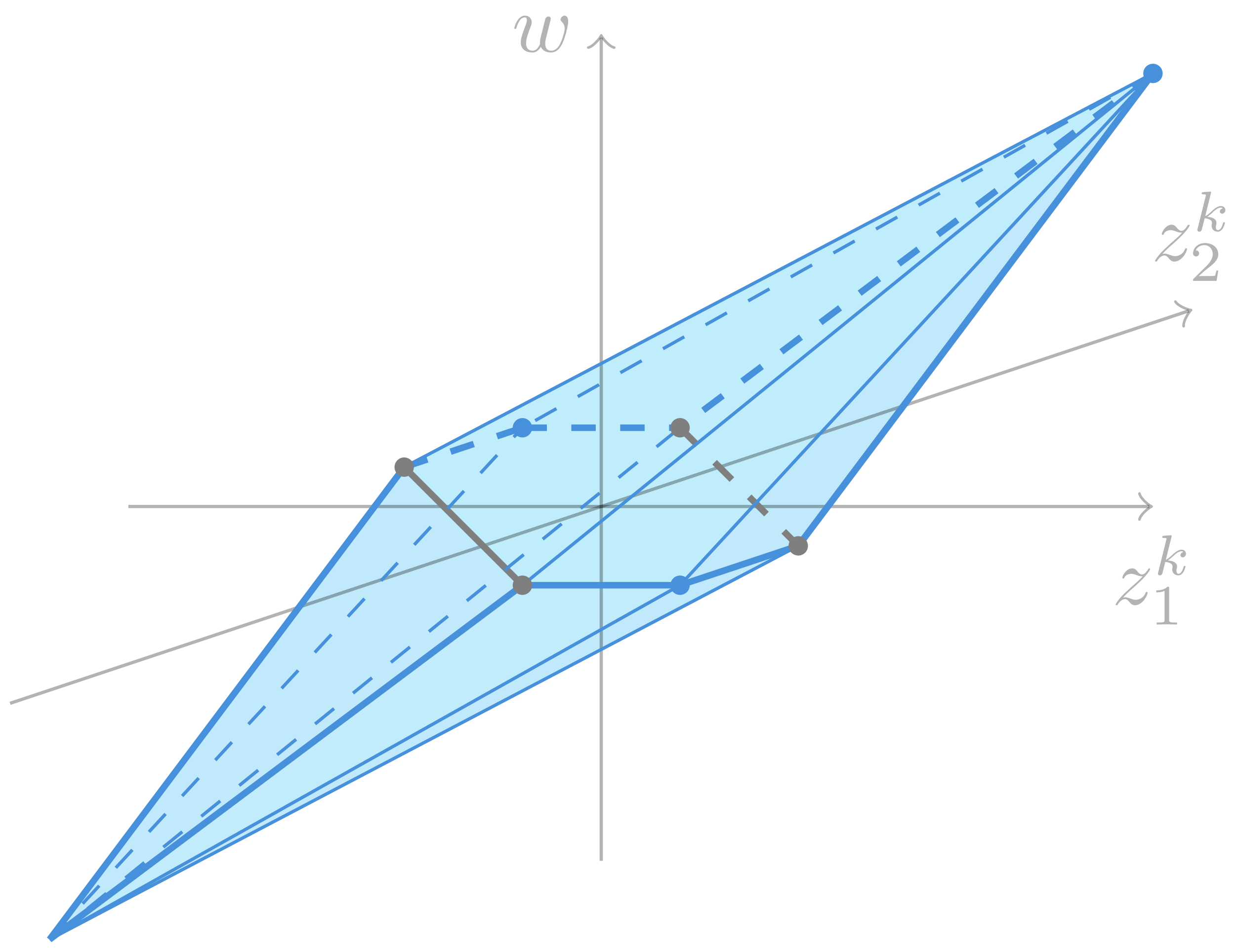
Convex hull of soft-thresholding operator

convex hull

$$\mathbf{conv}(\Phi) = \left\{ (z,w) \in [\underline{z}, \bar{z}] \times \mathbf{R} \;\middle|\; \begin{cases} w = a^T z - \lambda & \ell_{\{1,\ldots,n\}} > \lambda \\ w = a^T z + \lambda & u_{\{1,\ldots,n\}} < -\lambda \\ w = 0 & -\lambda \le \ell_{\{1,\ldots,n\}} \le u_{\{1,\ldots,n\}} \le \lambda \\ (z,w) \in Q & \text{otherwise} \end{cases} \right\}$$

$$Q = \left\{ \begin{array}{l} a^T z - \lambda \le w \le a^T z + \lambda \\ \frac{\ell_J + \lambda}{\ell_J - \lambda}(a^T z - \lambda) \le w \le \frac{u_J - \lambda}{u_J + \lambda}(a^T z + \lambda) \\ w \le \sum_{i \in I} a_i (z_i - \ell_i^0) + \frac{\ell_I - \lambda}{u_o^0 - \ell_o^0}(z_o - \ell_o^0), \; \forall (I, o) \in \mathcal{I}^+ \\ w \ge \sum_{i \in I} a_i (z_i - u_i^0) + \frac{u_I + \lambda}{\ell_o^0 - u_o^0}(z_o - u_o^0), \; \forall (I, o) \in \mathcal{I}^- \end{array} \right\}$$

example: $\phi_\lambda(z_1^k + z_2^k)$

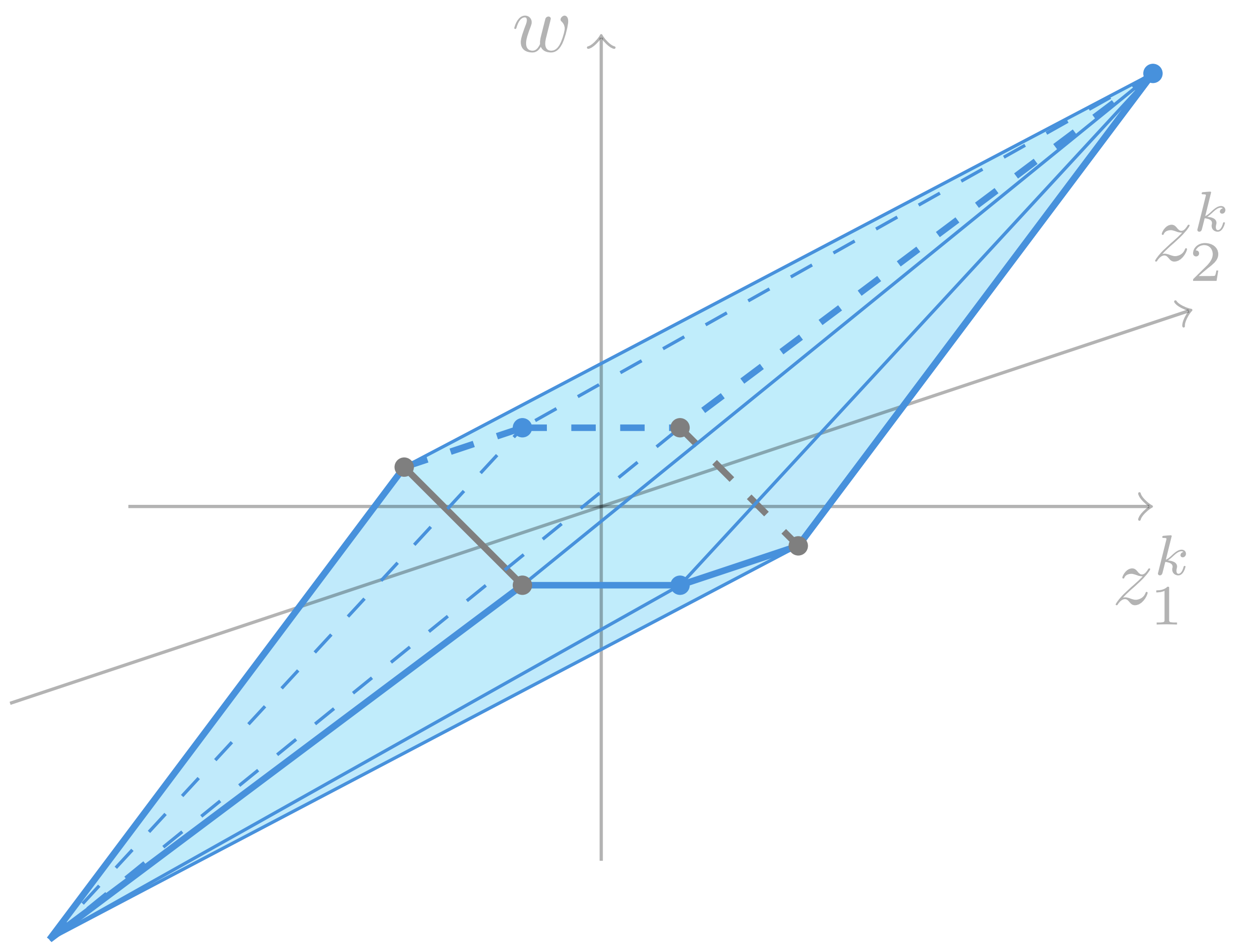


Convex hull of soft-thresholding operator

convex hull

$$\mathbf{conv}(\Phi) = \left\{ (z, w) \in [\underline{z}, \bar{z}] \times \mathbf{R} \;\middle|\; \begin{cases} w = a^T z - \lambda & \ell_{\{1,\ldots,n\}} > \lambda \\ w = a^T z + \lambda & u_{\{1,\ldots,n\}} < -\lambda \\ w = 0 & -\lambda \leq \ell_{\{1,\ldots,n\}} \leq u_{\{1,\ldots,n\}} \leq \lambda \\ (z, w) \in Q & \text{otherwise} \end{cases} \right\}$$

example: $\phi_\lambda(z_1^k + z_2^k)$



$$Q = \left\{ \begin{array}{l} a^T z - \lambda \leq w \leq a^T z + \lambda \\ \frac{\ell_J + \lambda}{\ell_J - \lambda}(a^T z - \lambda) \leq w \leq \frac{u_J - \lambda}{u_J + \lambda}(a^T z + \lambda) \\ w \leq \sum_{i \in I} a_i (z_i - \ell_i^0) + \frac{\ell_I - \lambda}{u_o^0 - \ell_o^0}(z_o - \ell_o^0), \; \forall (I, o) \in \mathcal{I}^+ \\ w \geq \sum_{i \in I} a_i (z_i - u_i^0) + \frac{u_I + \lambda}{\ell_o^0 - u_o^0}(z_o - u_o^0), \; \forall (I, o) \in \mathcal{I}^- \end{array} \right\}$$

exponential number of inequalities

$$\mathcal{I}^- = \{(I, o) \in 2^{\{1,\ldots,n\}} \times \{1, \ldots, n\} \mid u_I \leq -\lambda < u_{I \cup \{o\}}, \; w_I \neq 0\}$$

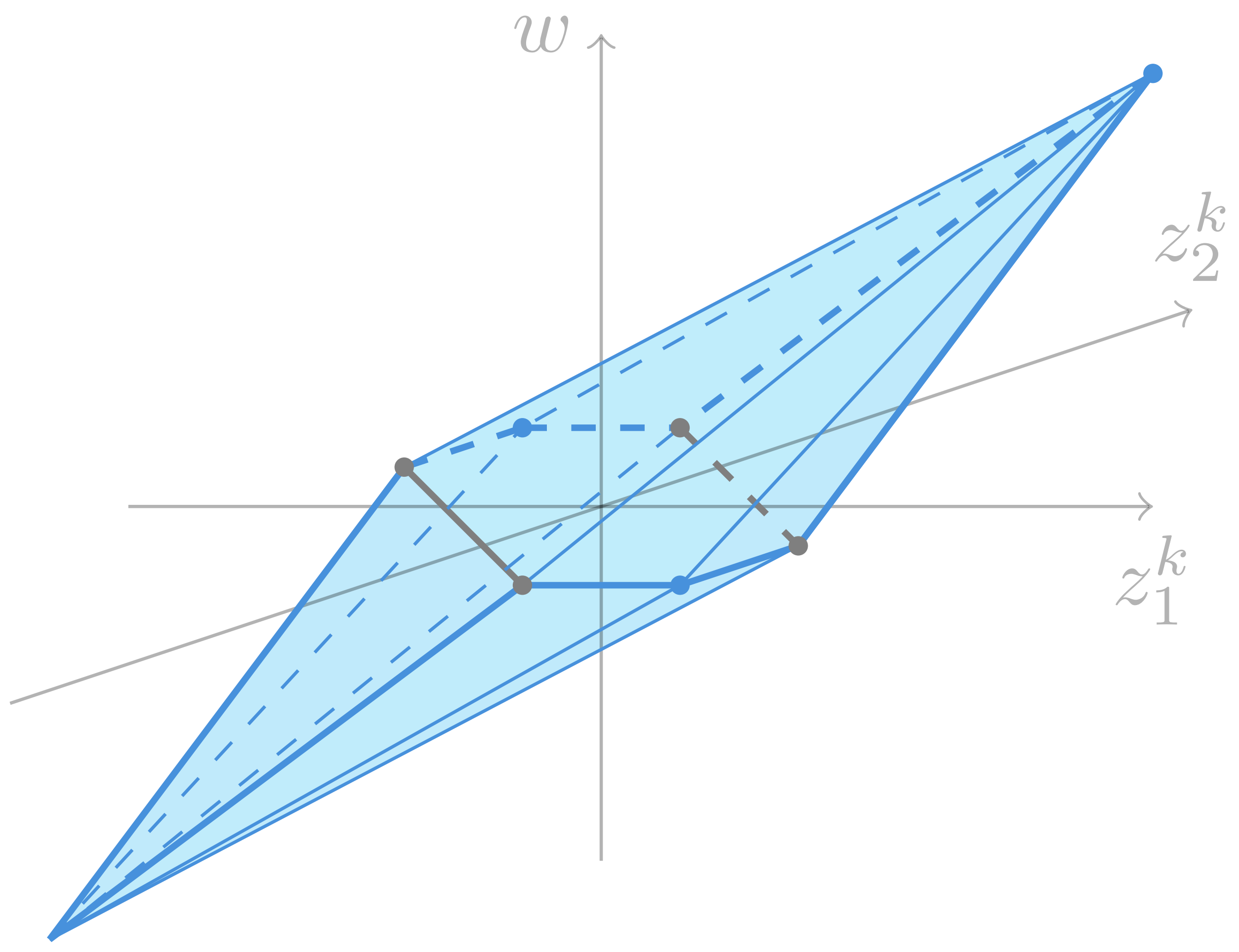
$$\mathcal{I}^+ = \{(I, o) \in 2^{\{1,\ldots,n\}} \times \{1, \ldots, n\} \mid \ell_{I \cup \{o\}} < \lambda \leq \ell_I, \; w_I \neq 0\}$$

Convex hull of soft-thresholding operator

convex hull

$$\mathbf{conv}(\Phi) = \left\{ (z,w) \in [\underline{z}, \bar{z}] \times \mathbf{R} \;\middle|\; \begin{cases} w = a^T z - \lambda & \ell_{\{1,\ldots,n\}} > \lambda \\ w = a^T z + \lambda & u_{\{1,\ldots,n\}} < -\lambda \\ w = 0 & -\lambda \le \ell_{\{1,\ldots,n\}} \le u_{\{1,\ldots,n\}} \le \lambda \\ (z,w) \in Q & \text{otherwise} \end{cases} \right\}$$

example: $\phi_\lambda(z_1^k + z_2^k)$



$$Q = \left\{ \begin{array}{l} a^T z - \lambda \le w \le a^T z + \lambda \\ \frac{\ell_J + \lambda}{\ell_J - \lambda}(a^T z - \lambda) \le w \le \frac{u_J - \lambda}{u_J + \lambda}(a^T z + \lambda) \\ w \le \sum_{i \in I} a_i (z_i - \ell_i^0) + \frac{\ell_I - \lambda}{u_o^0 - \ell_o^0}(z_o - \ell_o^0), \ \forall (I,o) \in \mathcal{I}^+ \\ w \ge \sum_{i \in I} a_i (z_i - u_i^0) + \frac{u_I + \lambda}{\ell_o^0 - u_o^0}(z_o - u_o^0), \ \forall (I,o) \in \mathcal{I}^- \end{array} \right\}$$

separation problem can be solved in linear time (by sorting)

exponential number of inequalities

$$\mathcal{I}^- = \{(I,o) \in 2^{\{1,\ldots,n\}} \times \{1,\ldots,n\} \mid u_I \le -\lambda < u_{I\cup\{o\}},\ w_I \ne 0\}$$

$$\mathcal{I}^+ = \{(I,o) \in 2^{\{1,\ldots,n\}} \times \{1,\ldots,n\} \mid \ell_{I\cup\{o\}} < \lambda \le \ell_I,\ w_I \ne 0\}$$

Computing initial radius

PEP

$$R^2 = \begin{array}{ll} \text{maximize} & \|z^0 - z^\star\|_2^2 \\ \text{subject to} & z^\star = T(z^\star, x) \\ & x \in \mathcal{X} \end{array}$$

$\longleftarrow$ fixed-points

example
linear convergence

$$\alpha_K = 2\beta^{k-1}R$$

upper-bounds
∞-norm

Verification

$$R = \begin{array}{ll} \text{maximize} & \|z^0 - z^\star\|_1 \\ \text{subject to} & z^\star = T(z^\star, x) \\ & x \in \mathcal{X} \end{array}$$

$\longleftarrow$ fixed-points