Frank-Wolfe Beyond Polytopes: Convergence on Non-Polyhedral Sets

Sebastian Pokutta

joint work with: Jannis Halbey, Daniel Deza, Max Zimmer, Christophe Roux, and Bartolomeo Stellato

Technische Universität Berlin
and
Zuse Institute Berlin

pokutta@math.tu-berlin.de
@spokutta

RICAM – Radon Institute for Computational and Applied Mathematics

Workshop on Optimization in AI (RICAM Special Semester)

November 24, 2026 · Linz, Austria









What is this talk about?

Introduction

How does the Frank-Wolfe algorithm behave on curved domains?

What is this talk about?

Introduction

How does the Frank-Wolfe algorithm behave on curved domains?

Why? Most results are for polytopes and convergence on curved domains is not well understood.

What is this talk about?

Introduction

How does the Frank-Wolfe algorithm behave on curved domains?

Why? Most results are for polytopes and convergence on curved domains is not well understood.

Today.

1. A lower bound for smooth + strongly convex functions on strongly convex domains
2. An upper bound for smooth + (only) convex functions on curved domains

(Hyperlinked) References are not exhaustive; check references contained therein.

Conditional Gradients
a.k.a. the Frank-Wolfe algorithm

—The Basics—

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

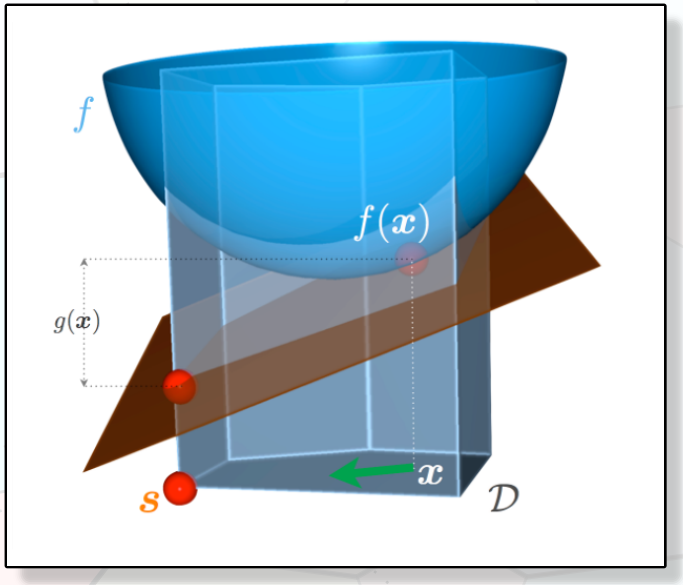
Given a smooth and convex function f and a polytope P, solve optimization problem:

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$



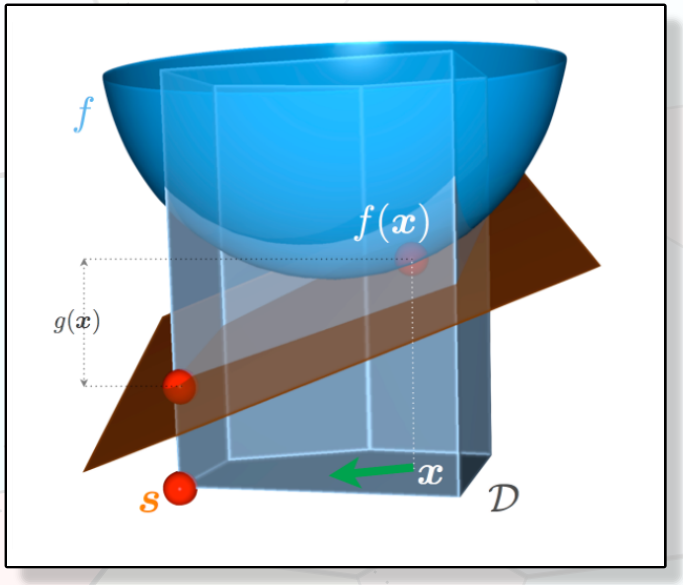
Source: [Jaggi, 2013]

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$



Source: [Jaggi, 2013]

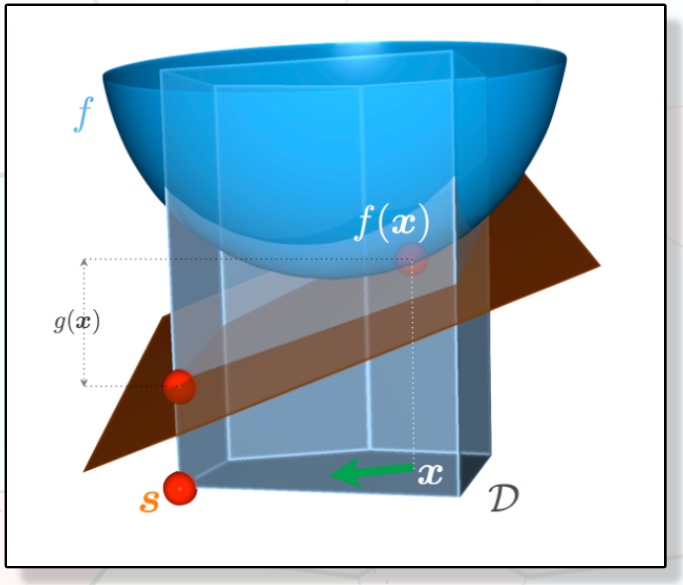
1. Very versatile model
2. Can use various types of information about both f and P
3. Works very well in (continuous) real-world applications
4. At the core of many (all?) learning algorithms (albeit mostly non-convex)

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$



Source: [Jaggi, 2013]

Our setup.

1. Access to P. Linear Minimization Oracle (LMO): Given linear objective c return

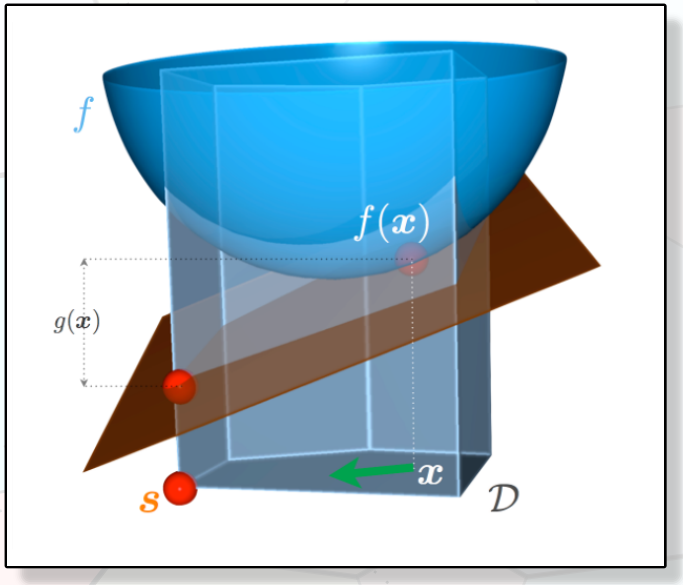
$$x \leftarrow \arg\min_{v \in P} c^T v.$$

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$



Source: [Jaggi, 2013]

Our setup.

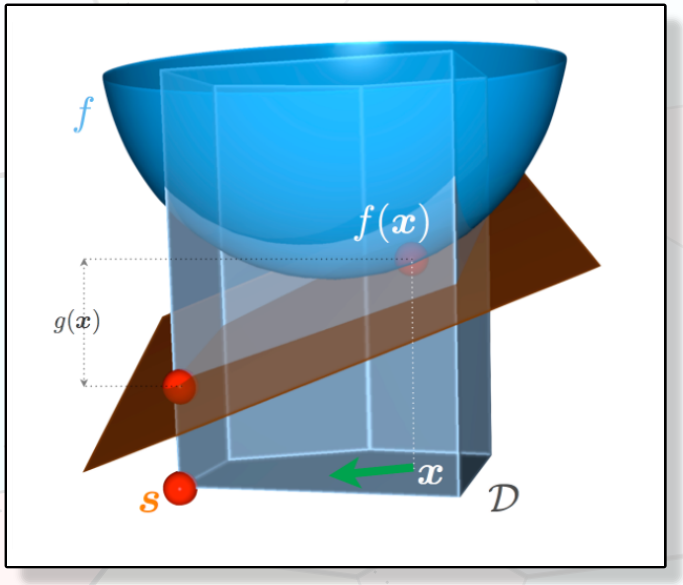
1. Access to P. Linear Minimization Oracle (LMO): Given linear objective c return
$$x \leftarrow \arg\min_{v \in P} c^T v.$$
2. Access to f. First-Order Oracle (FO): Given x return
$$\nabla f(x) \quad \text{and} \quad f(x).$$

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$



Source: [Jaggi, 2013]

Our setup.

1. Access to P. Linear Minimization Oracle (LMO): Given linear objective c return
$$x \leftarrow \arg\min_{v \in P} c^T v.$$
2. Access to f. First-Order Oracle (FO): Given x return
$$\nabla f(x) \quad \text{and} \quad f(x).$$

$\Rightarrow$ *Complexity of convex optimization relative to LO/FO oracle*

The Frank-Wolfe Algorithm

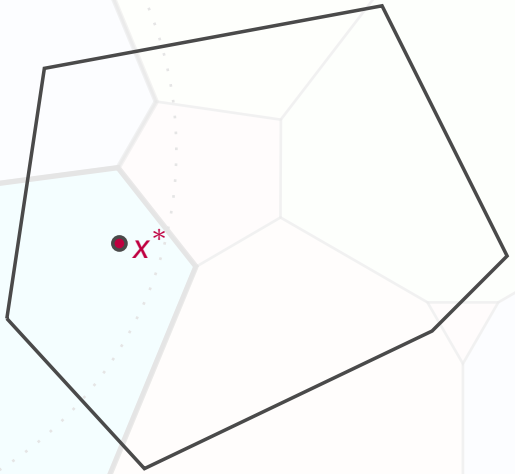
Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

1: **for** $t = 0$ **to** $T - 1$ **do**
2: $\quad v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t) \rangle v$
3: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

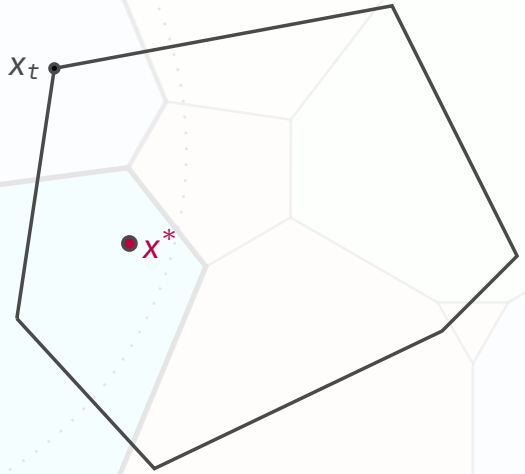
Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

1: **for** $t = 0$ **to** $T - 1$ **do**
2: $\quad v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t) \rangle v$
3: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

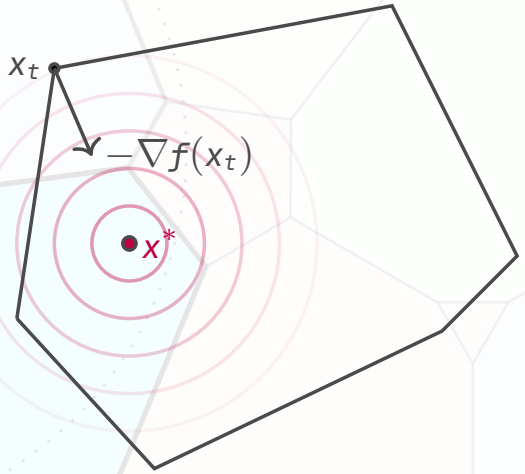
Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

1: **for** $t = 0$ **to** $T - 1$ **do**
2: $\quad v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t) \rangle v$
3: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

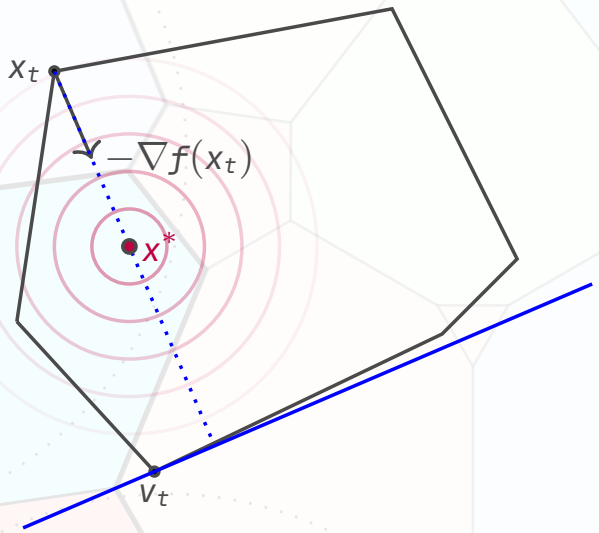
Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

1: **for** $t = 0$ **to** $T - 1$ **do**

2: $\quad v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t) \rangle v$

3: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

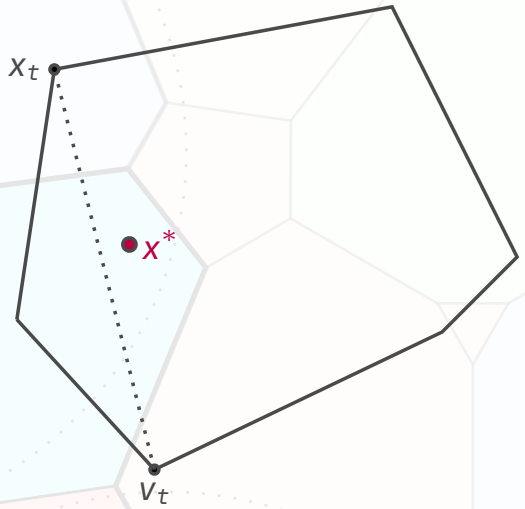
Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

1: **for** $t = 0$ **to** $T - 1$ **do**

2: $\quad v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t) \rangle v$

3: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

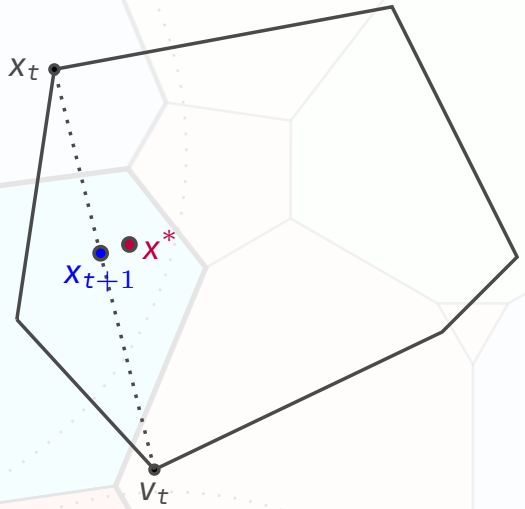
Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

1: **for** $t = 0$ **to** $T - 1$ **do**

2: $\quad v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t) \rangle v$

3: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

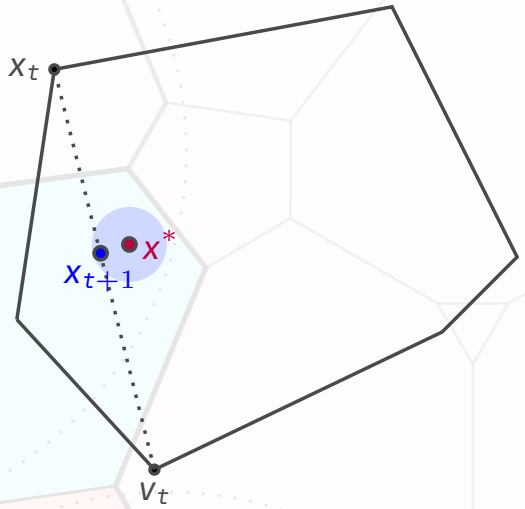
Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

1: **for** $t = 0$ **to** $T - 1$ **do**

2: $\quad v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t) \rangle v$

3: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

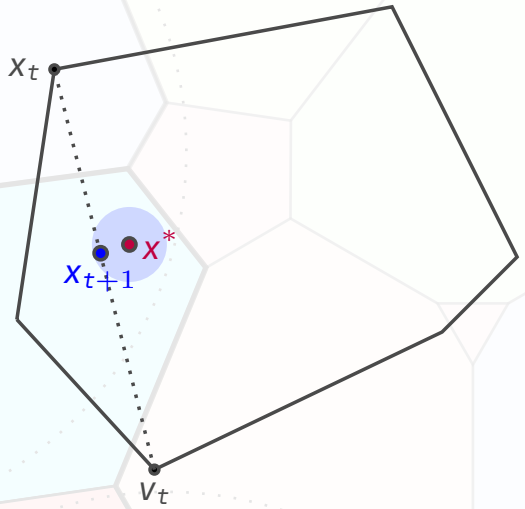
The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$
Output: Iterates $x_1, x_2, \ldots \in P$

1: **for** $t = 0$ **to** $T - 1$ **do**
2: $\quad v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t) \rangle v$
3: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

Advantages:

- Extremely simple and robust: no complicated data structures to maintain
- Easy to implement: requires only the two oracles
- Projection-free: feasibility convex combination and LO oracle.
- Sparsity: optimal solution is a convex combination of (usually) vertices.

The Frank-Wolfe Algorithm

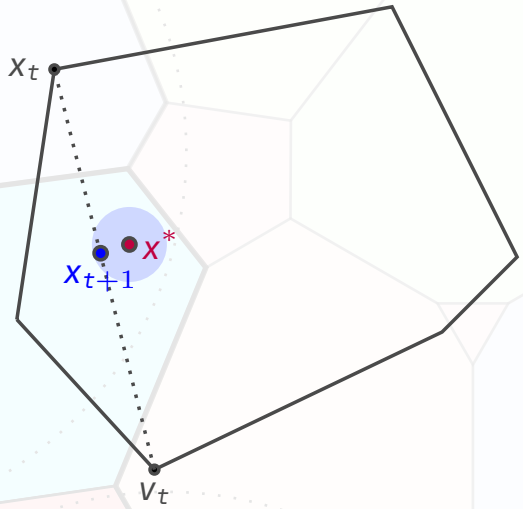
Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

1: **for** $t = 0$ **to** $T - 1$ **do**
2: $\quad v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t) \rangle v$
3: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

Advantages:

- Extremely simple and robust: no complicated data structures to maintain
- Easy to implement: requires only the two oracles
- Projection-free: feasibility convex combination and LO oracle.
- Sparsity: optimal solution is a convex combination of (usually) vertices.

Disadvantages:

- Suboptimal convergence rate of $O(1/T)$

The Frank-Wolfe Algorithm

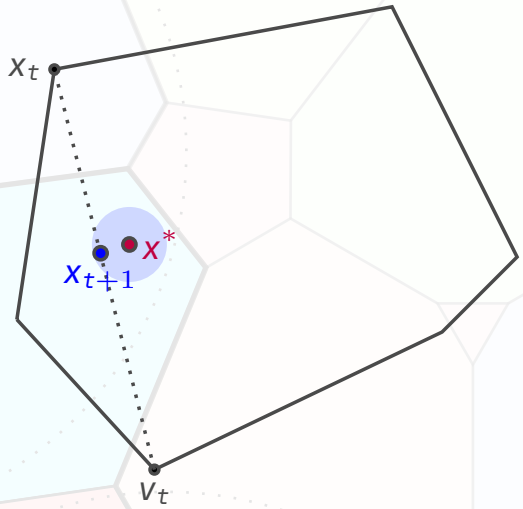
Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

1: **for** $t = 0$ **to** $T - 1$ **do**
2: $\quad v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t) \rangle v$
3: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

Advantages:

- Extremely simple and robust: no complicated data structures to maintain
- Easy to implement: requires only the two oracles
- Projection-free: feasibility convex combination and LO oracle.
- Sparsity: optimal solution is a convex combination of (usually) vertices.

Disadvantages:

- Suboptimal convergence rate of $O(1/T)$

⇒ Despite (theoretically) suboptimal heavily used in applications as simple.

Significant progress over the recent years (incomplete list)

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

1. Strongly convex [Garber and Hazan, 2013, Lacoste-Julien and Jaggi, 2015, Lan and Zhou, 2016, Garber and Meshi, 2016]
2. Non-convex [Lacoste-Julien, 2016]
3. Online [Hazan and Kale, 2012]
4. Stochastic/adaptive gradients [Hazan and Luo, 2016, Reddi et al., 2016, Combettes et al., 2020]
5. Sharp functions and sharp regions [Kerdreux et al., 2019, 2021, 2026]
6. Acceleration [Diakonikolas et al., 2020, Bach, 2020, Carderera et al., 2021]
7. Specialized variants [Freund et al., 2017, Braun et al., 2017, 2019b,a]

Conditional Gradients very competitive:
simple, robust, and great real-world performance.

For more background etc see our book! [Braun et al., 2025]

Lower Bounds for Frank-Wolfe on Strongly Convex Sets

Strongly Convex Sets

Introduction

Definition. A convex set $\mathcal{X}$ is **α-strongly convex** if for all $x, y \in \mathcal{X}$, $\lambda \in [0, 1]$ and z s.t. $\|z\| \leq 1$

$$\lambda x + (1-\lambda)y + \frac{\alpha}{2}\lambda(1-\lambda)\|x-y\|^2 z \subseteq \mathcal{X}$$

Strongly Convex Sets

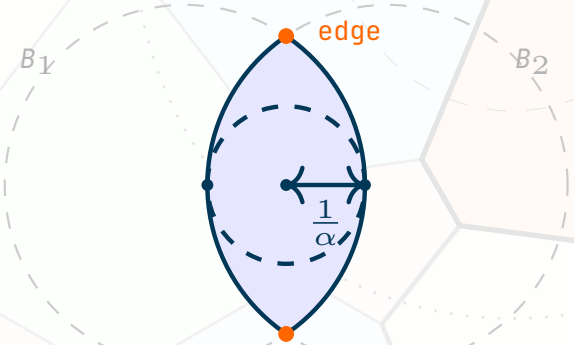
Introduction

Definition. A convex set $\mathcal{X}$ is **α-strongly convex** if for all $x, y \in \mathcal{X}$, $\lambda \in [0,1]$ and z s.t. $\|z\| \leq 1$

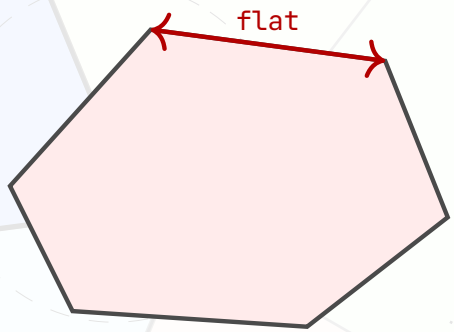
$$\lambda x + (1-\lambda)y + \frac{\alpha}{2}\lambda(1-\lambda)\|x-y\|^2 z \subseteq \mathcal{X}$$

Examples.

1. ℓ_p balls for $1 < p \leq 2$
2. Schatten-p norm balls for $1 < p \leq 2$
3. ellipsoids



$B_1 \cap B_2$: edges OK,
still strongly convex



polytope: no curvature,
not strongly convex

Prior Results

Introduction

General compact convex sets.

- Upper bound: $\mathcal{O}(1/\varepsilon)$ [Frank and Wolfe, 1956, Levitin and Polyak, 1966, Jaggi, 2013]
- Lower bound: $\Omega(1/\varepsilon)$ [Jaggi, 2013, Lan, 2013]

Limitation. The Frank-Wolfe-specific lower bounds use *polyhedral* constructions; they do not transfer to curved feasible regions.

Prior Results

Introduction

General compact convex sets.

- Upper bound: $\mathcal{O}(1/\varepsilon)$ [Frank and Wolfe, 1956, Levitin and Polyak, 1966, Jaggi, 2013]
- Lower bound: $\Omega(1/\varepsilon)$ [Jaggi, 2013, Lan, 2013]

Limitation. The Frank-Wolfe-specific lower bounds use *polyhedral* constructions; they do not transfer to curved feasible regions.

Strongly convex sets.

- Upper bound: $\mathcal{O}(1/\sqrt{\varepsilon})$ [Garber and Hazan, 2015]
- Only generally applicable lower bound: $\Omega\left(\log \frac{1}{\varepsilon}\right)$ [Nemirovsky and Yudin, 1983]

Prior Results

Introduction

General compact convex sets.

- Upper bound: $\mathcal{O}(1/\varepsilon)$ [Frank and Wolfe, 1956, Levitin and Polyak, 1966, Jaggi, 2013]
- Lower bound: $\Omega(1/\varepsilon)$ [Jaggi, 2013, Lan, 2013]

Limitation. The Frank-Wolfe-specific lower bounds use *polyhedral* constructions; they do not transfer to curved feasible regions.

Strongly convex sets.

- Upper bound: $\mathcal{O}(1/\sqrt{\varepsilon})$ [Garber and Hazan, 2015]
- Only generally applicable lower bound: $\Omega\big(\log \frac{1}{\varepsilon}\big)$ [Nemirovsky and Yudin, 1983]

Large gap of $\Omega\big(\log \frac{1}{\varepsilon}\big)$ versus $\mathcal{O}(1/\sqrt{\varepsilon})$.

Optimizer Position matters

Introduction

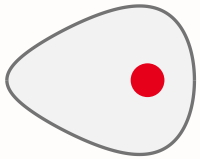
Baseline for strongly convex sets. Without information about the optimizer position Frank-Wolfe needs $\mathcal{O}(1/\sqrt{\varepsilon})$ iterations. [Garber and Hazan, 2015]

Optimizer Position matters

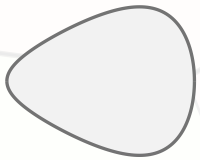
Introduction

Baseline for strongly convex sets. Without information about the optimizer position Frank-Wolfe needs $\mathcal{O}(1/\sqrt{\varepsilon})$ iterations. [Garber and Hazan, 2015]

Location changes the rate. The position of the *unconstrained* optimizer x^*



Interior x^*: upper bound $\mathcal{O}(\log \frac{1}{\varepsilon})$ [Wolfe, 1970, Guélat and Marcotte, 1986]



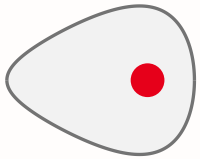
Exterior x^*: upper bound $\mathcal{O}(\log \frac{1}{\varepsilon})$ [Levitin and Polyak, 1966]

Optimizer Position matters

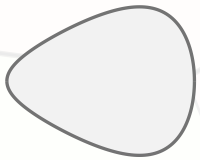
Introduction

Baseline for strongly convex sets. Without information about the optimizer position Frank-Wolfe needs $\mathcal{O}(1/\sqrt{\varepsilon})$ iterations. [Garber and Hazan, 2015]

Location changes the rate. The position of the *unconstrained* optimizer x^*



Interior x^*: upper bound $\mathcal{O}(\log \frac{1}{\varepsilon})$ [Wolfe, 1970, Guélat and Marcotte, 1986]



Exterior x^*: upper bound $\mathcal{O}(\log \frac{1}{\varepsilon})$ [Levitin and Polyak, 1966]

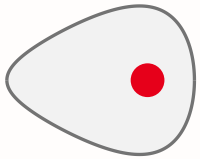
What happens on the boundary?

Optimizer Position matters

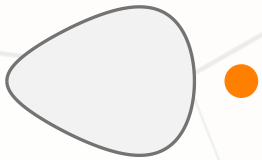
Introduction

Baseline for strongly convex sets. Without information about the optimizer position Frank-Wolfe needs $\mathcal{O}(1/\sqrt{\varepsilon})$ iterations. [Garber and Hazan, 2015]

Location changes the rate. The position of the *unconstrained* optimizer x^*

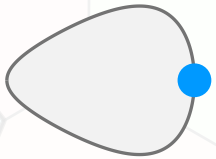


Interior x^*: upper bound $\mathcal{O}(\log \frac{1}{\varepsilon})$ [Wolfe, 1970, Guélat and Marcotte, 1986]



Exterior x^*: upper bound $\mathcal{O}(\log \frac{1}{\varepsilon})$ [Levitin and Polyak, 1966]

What happens on the boundary?



Boundary x^* requires $\Omega(1/\sqrt{\varepsilon})$ [Halbey et al., 2026a]

Constructing a Lower Bound

Model Problem

Lower Bound

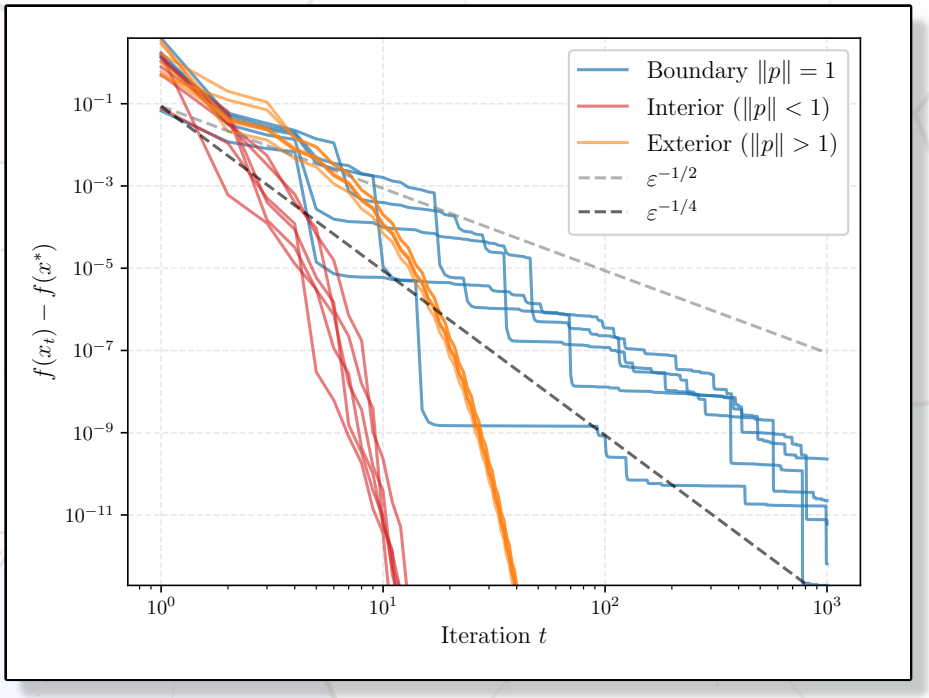
Projection onto the unit ball.

$$\min_{x \in B_1(0)} \|x - p\|^2, \quad \|p\| = 1$$

1. $f(x)$ strongly convex + smooth
2. $B_1(0)$ strongly convex + smooth
3. Optimizer $x^* = p$ on the boundary
4. Run Frank-Wolfe with short steps

Properties.

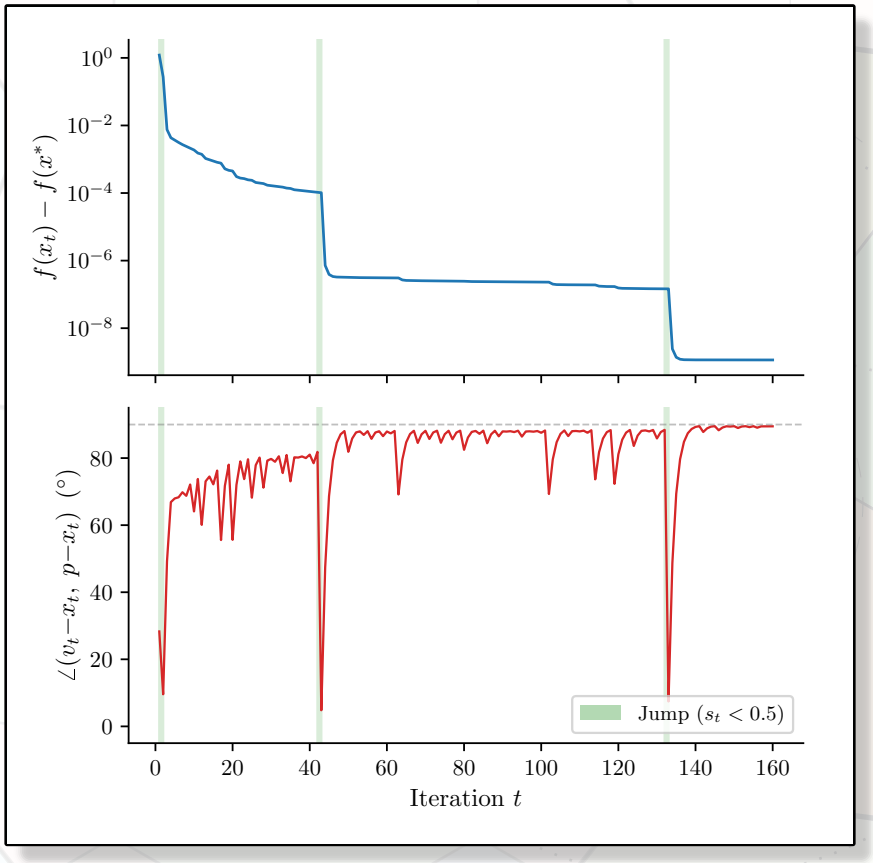
1. All iterates remain in 2D space $\mathrm{Span}\{p, x_0\}$.
2. Closed-form LMO: $v_t = \frac{-\nabla f(x_t)}{\|\nabla f(x_t)\|}$



FW convergence for different positions of p.

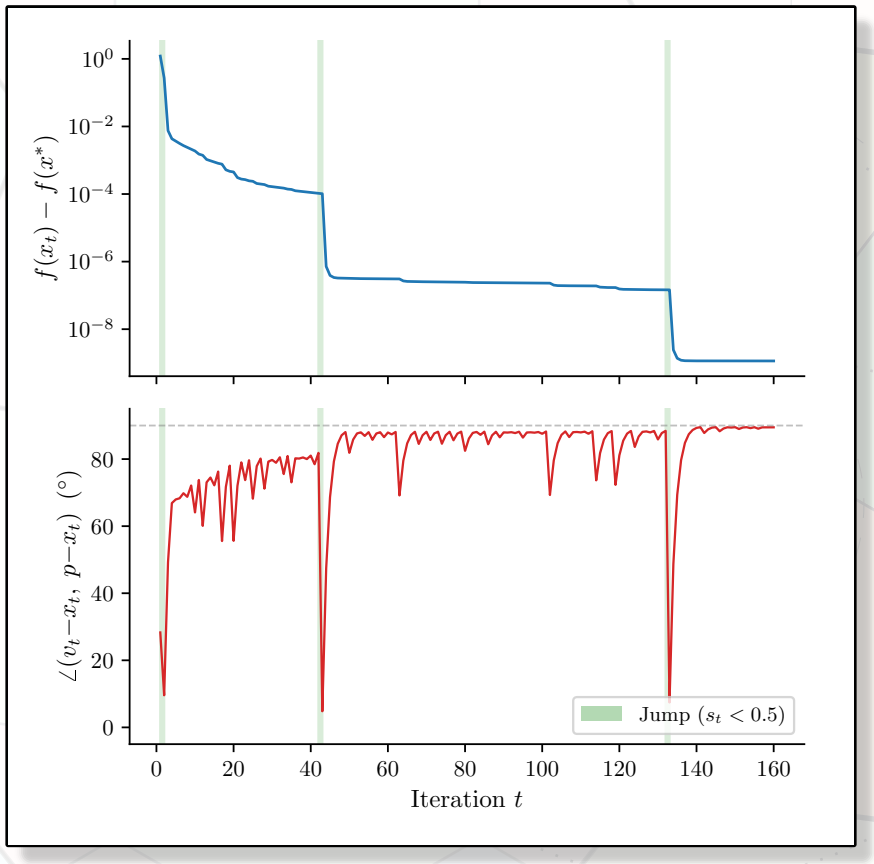
Convergence Behavior

Lower Bound



Convergence Behavior

Lower Bound



Pattern.

1. **Long plateaus.** FW direction nearly orthogonal to optimal direction
   $\Rightarrow$ little progress
2. **Intermittent jumps.** angle drops, directions align
   $\Rightarrow$ steep error decrease

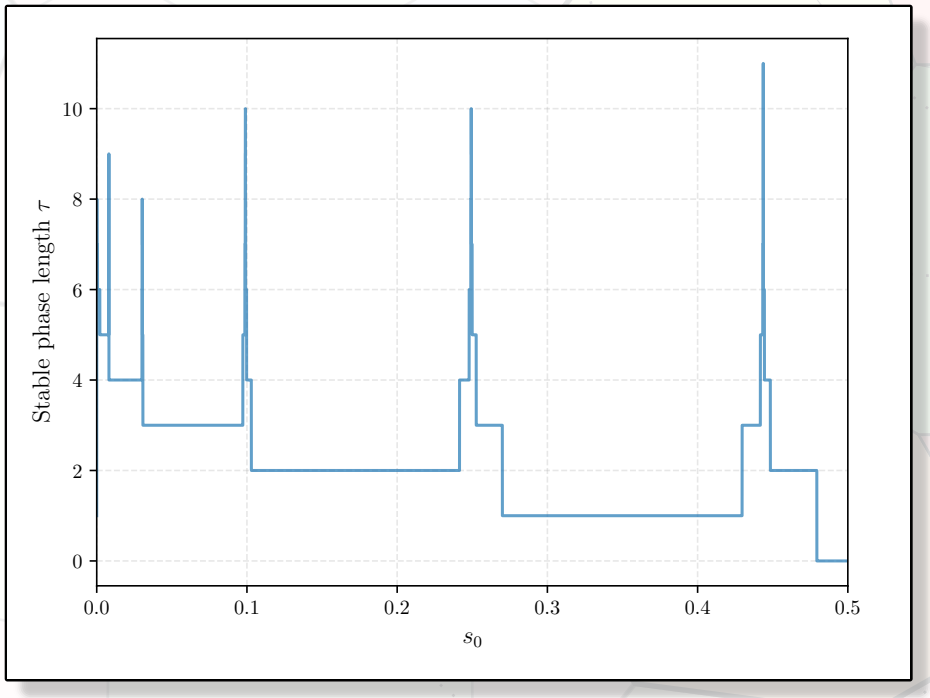
Naive Strategy. Find an initialization such that no jumps occur.

Naive Forward Search

Lower Bound

Goal. Find initializations with a long stable phase.

Grid search. Fix $r_0 = 1$ and vary s_0



Stable phase length vs. initial contraction s_0

Naive Forward Search

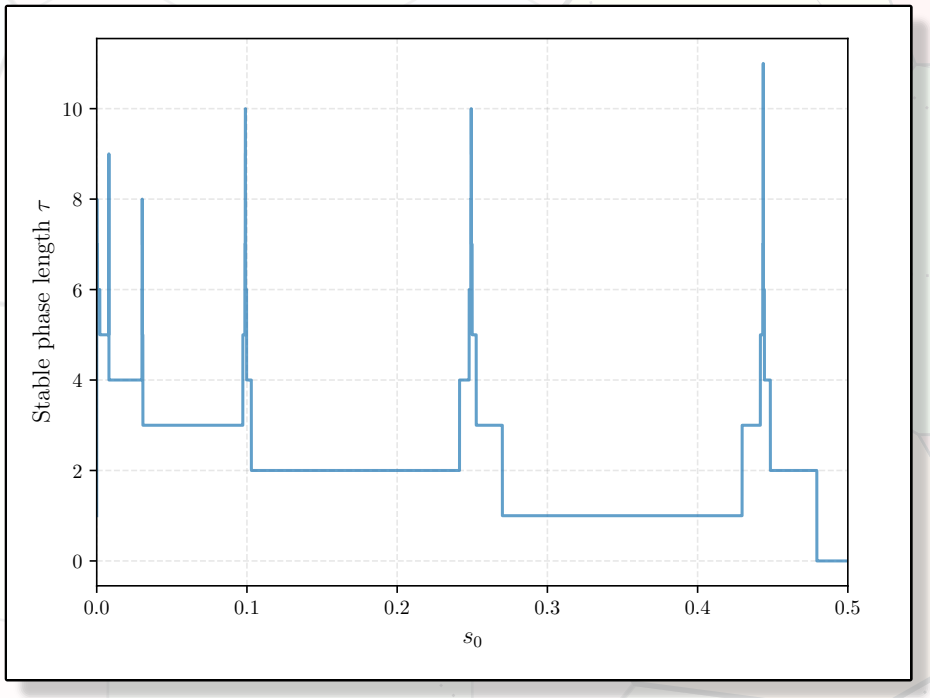
Lower Bound

Goal. Find initializations with a long stable phase.

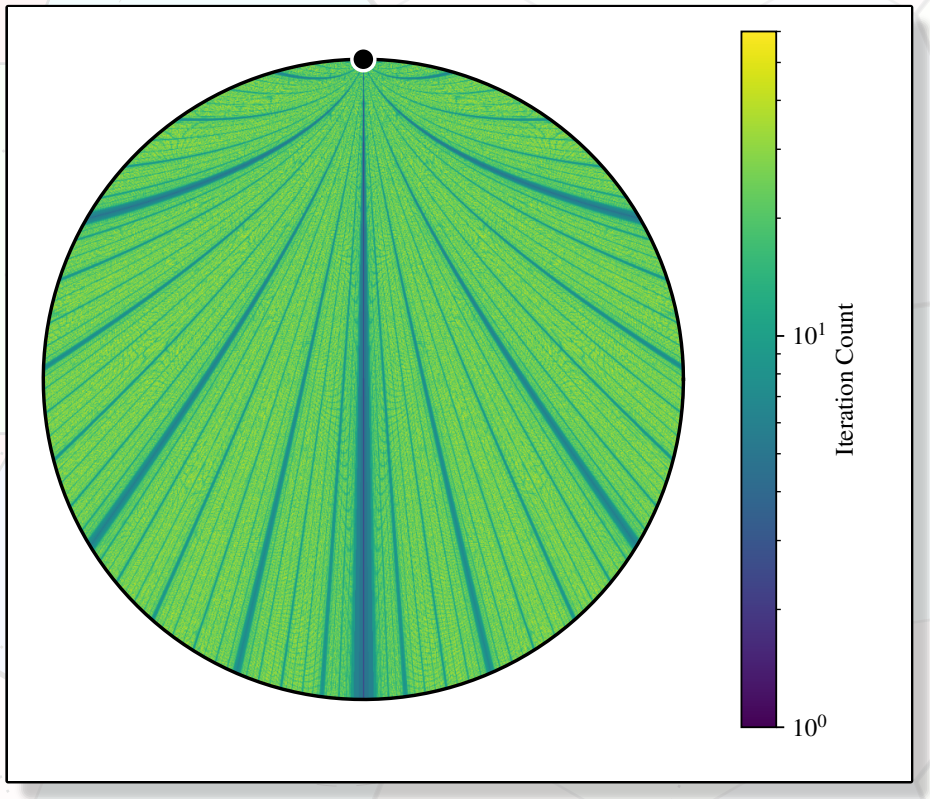
Grid search. Fix $r_0 = 1$ and vary s_0

Bisection. Refine around spikes.

Doomed. Requires extreme numerical precision; approx 1 bit per iteration



Stable phase length vs. initial contraction s_0



Number of FW steps to reach $f(x_t) - f(x^*) \leq 10^{-4}$

Analyzing the Dynamics

Lower Bound

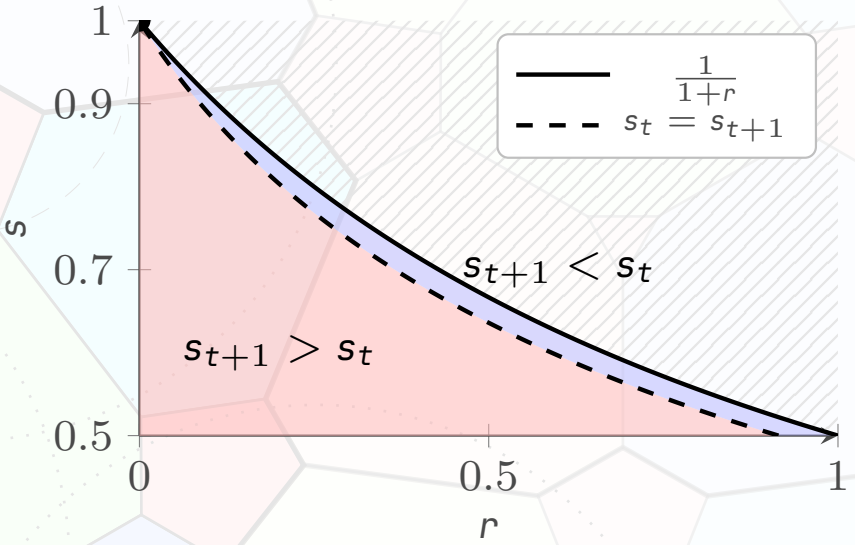
Reparametrize the iterates in 2D.

$$r_t \overset{\text{def}}{=} \|x_t - p\|, \qquad s_t \overset{\text{def}}{=} \frac{r_{t+1}}{r_t}$$

1. r_t: distance to optimum
2. $s_t \in [0, 1]$: contraction per step
3. Convergence: $(r_t, s_t) \to (0, 1)$
4. Exact FW dynamics (exact ls):

$$r_{t+1} = s_t r_t,$$

$$s_{t+1}^2 = \frac{r_{t+2}^2}{r_{t+1}^2} = \frac{1 - (1 + r_t)^2 s_t^2}{2 - 2s_t - (2 + r_t) r_t s_t^2}.$$



Phase diagram in (r, s) space

Analyzing the Dynamics

Lower Bound

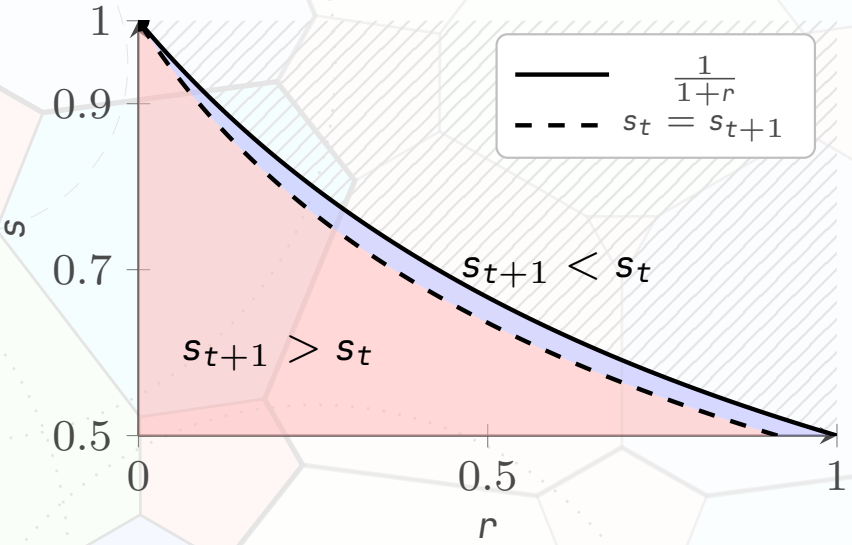
Reparametrize the iterates in 2D.

$$r_t \overset{\text{def}}{=} \|x_t - p\|, \qquad s_t \overset{\text{def}}{=} \frac{r_{t+1}}{r_t}$$

1. r_t: distance to optimum
2. $s_t \in [0, 1]$: contraction per step
3. Convergence: $(r_t, s_t) \to (0, 1)$
4. Exact FW dynamics (exact ls):

$$r_{t+1} = s_t r_t,$$

$$s_{t+1}^2 = \frac{r_{t+2}^2}{r_{t+1}^2} = \frac{1 - (1 + r_t)^2 s_t^2}{2 - 2s_t - (2 + r_t) r_t s_t^2}.$$



Phase diagram in (r, s) space

Plateaus and jumps.

1. **Plateaus.** **stable** region: $s_{t+1} > s_t$ progress slows down
2. **Jumps.** **unstable** region: $s_{t+1} < s_t$ error drops sharply

Backward Reconstruction

Lower Bound

Rolling the dynamics backwards.

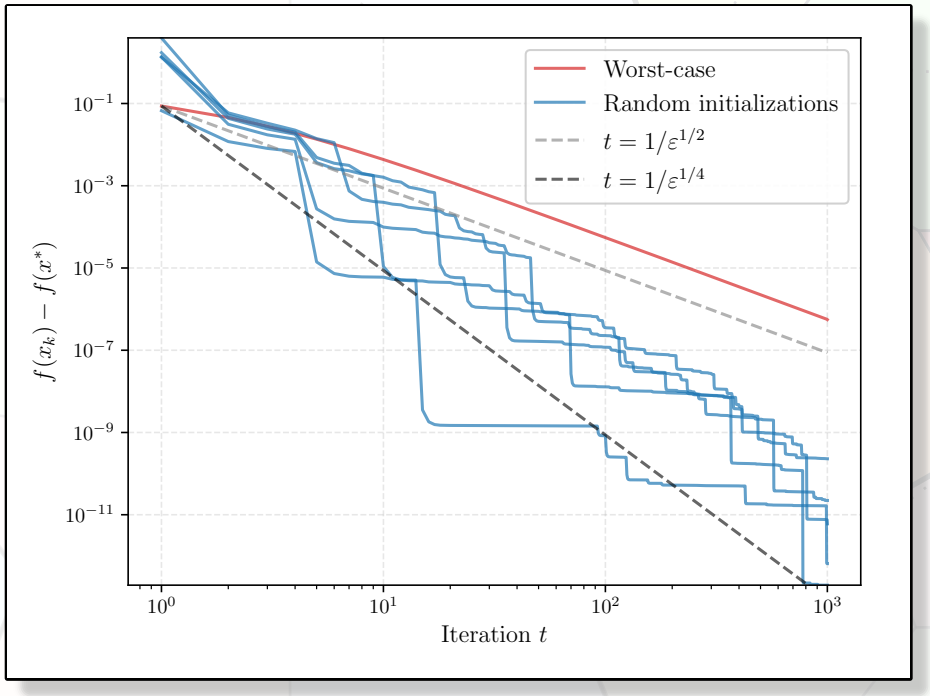
1. Start near $(r^*, s^*) = (0, 1)$ and use $s = 1 - \frac{4}{3}r \Rightarrow (\varepsilon, 1 - \frac{4}{3}\varepsilon)$ via Taylor
2. No need to do expensive forward search
3. Endpoint gives the worst-case initialization
4. Need to work a little bit to stay in the slow regime

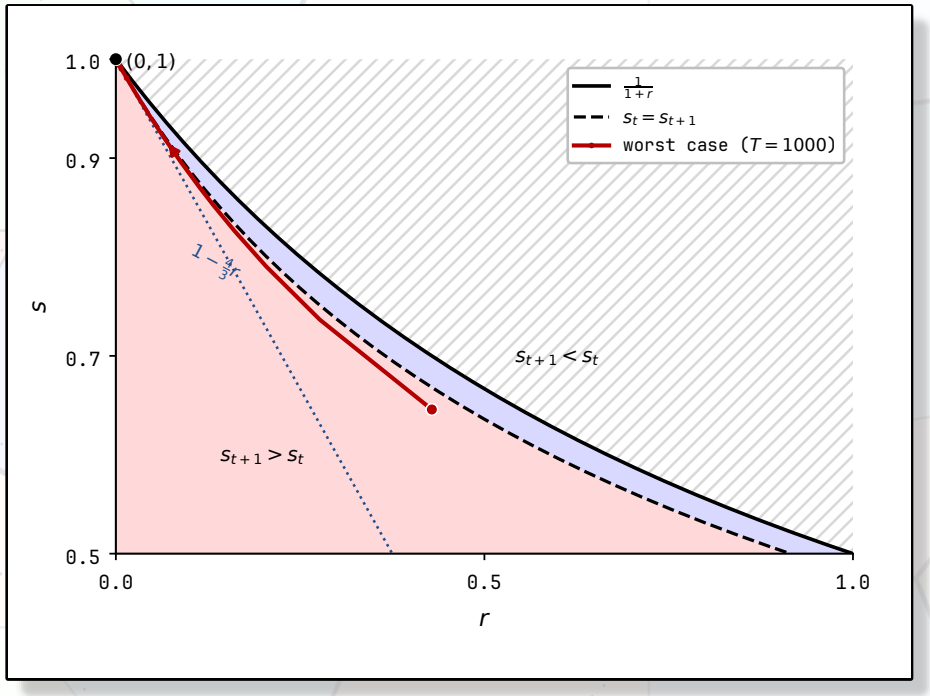
Backward Reconstruction

Lower Bound

Rolling the dynamics backwards.

1. Start near $(r^*, s^*) = (0, 1)$ and use $s = 1 - \frac{4}{3}r \Rightarrow (\varepsilon, 1 - \frac{4}{3}\varepsilon)$ via Taylor
2. No need to do expensive forward search
3. Endpoint gives the worst-case initialization
4. Need to work a little bit to stay in the slow regime



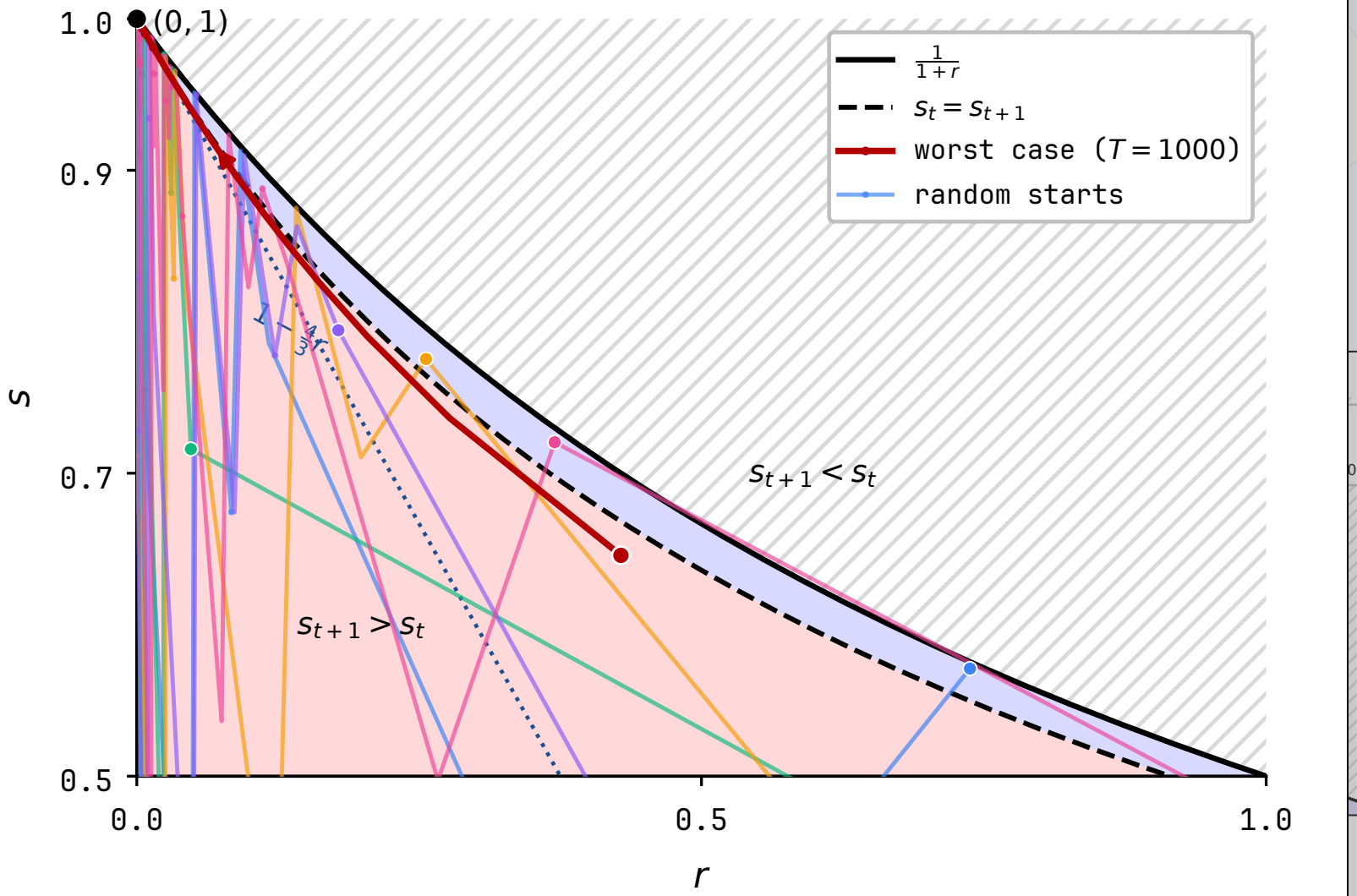


Backward Reconstruction

Lower Bound

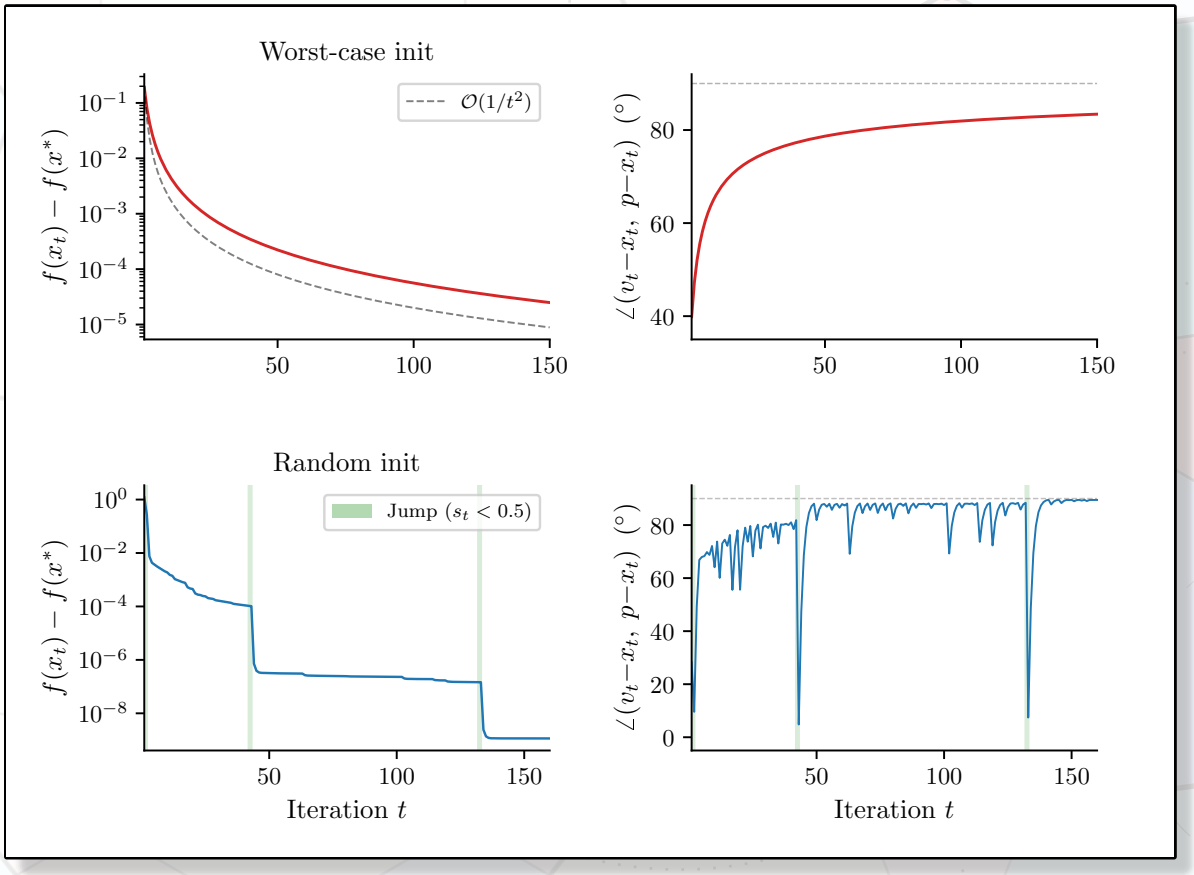
Rolli

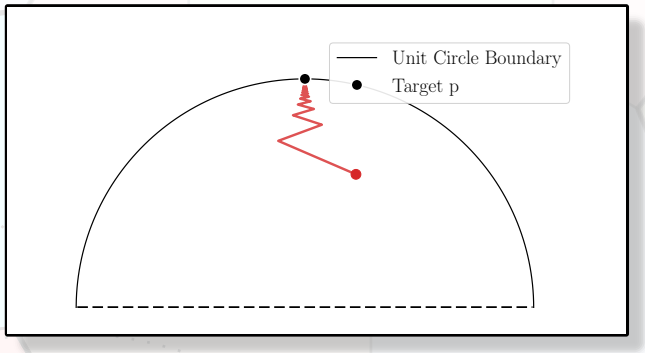
1. S
2. N
3. E
4. N

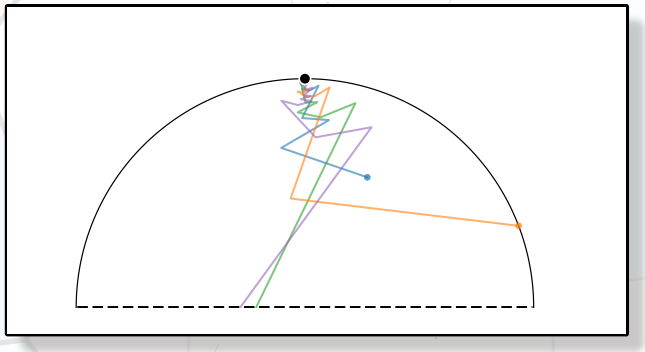


Constructed Worst Cases: Does it work?

Lower Bound



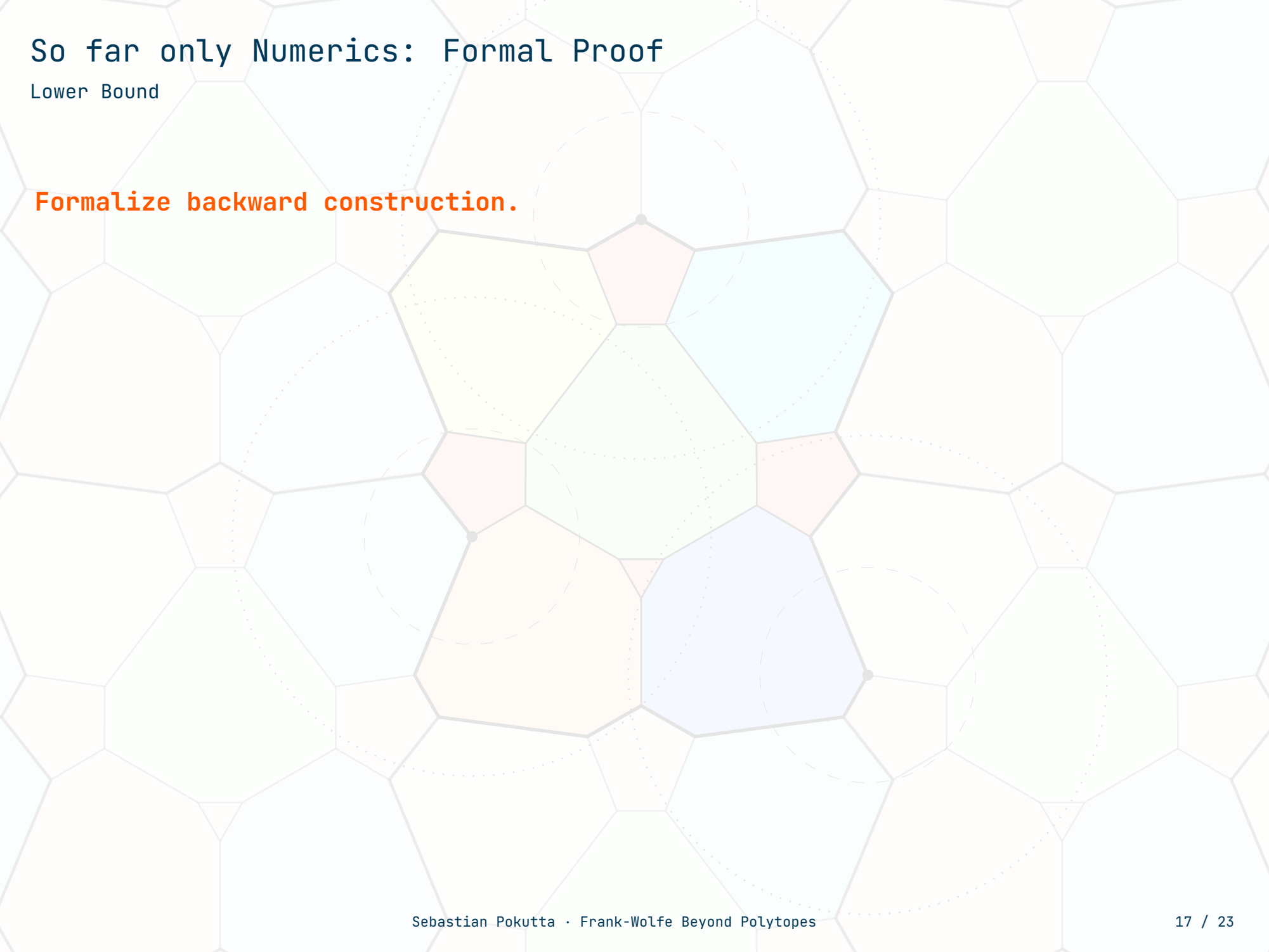




So far only Numerics: Formal Proof

Lower Bound

Formalize backward construction.



So far only Numerics: Formal Proof

Lower Bound

Formalize backward construction.

Show $s_t \geq 1 - \frac{4}{3} r_t$ and $s_{t+1} > s_t$. This then implies $r_t = \Omega(1/t)$ and hence

$$f(x_t) - f(x^*) = \Omega(1/t^2).$$

So far only Numerics: Formal Proof

Lower Bound

Formalize backward construction.

Show $s_t \geq 1 - \frac{4}{3} r_t$ and $s_{t+1} > s_t$. This then implies $r_t = \Omega(1/t)$ and hence

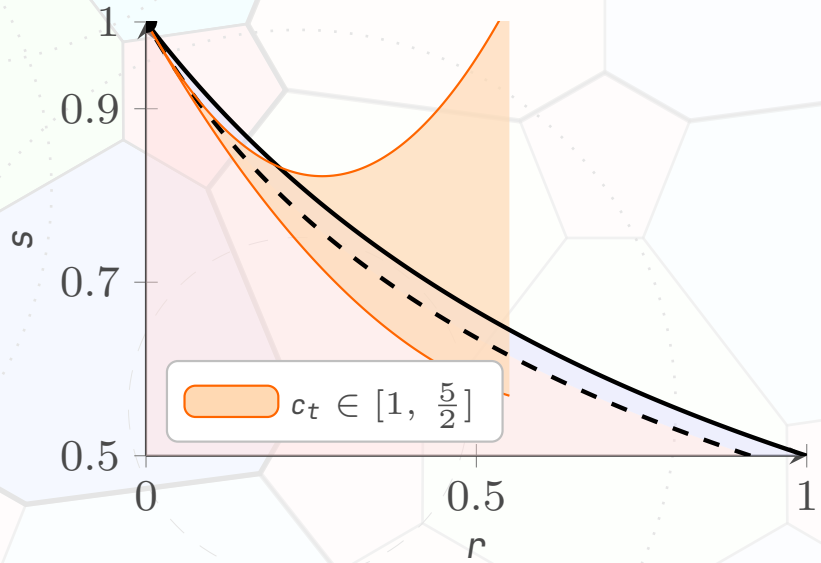
$$f(x_t) - f(x^*) = \Omega(1/t^2).$$

Add a quadratic Taylor correction

$$s_t = 1 - \tfrac{4}{3} r_t + c_t r_t^2$$

Invariant band. One backward step preserves

$$c_t \in [1, \tfrac{5}{2}] \implies c_{t-1} \in [1, \tfrac{5}{2}]$$



Main Result

Lower Bound

Theorem (Lower Bound). There exists a compact, smooth, and strongly convex set $\mathcal{X} \subset \mathbb{R}^d$, a smooth and strongly convex function $f : \mathcal{X} \to \mathbb{R}$, such that for all $T \in \mathbb{N}$ there exists a starting point $x_0 \in \mathcal{X}$ such that the FW iterates (with short step or exact line search) satisfy

$$f(x_t) - f(x^*) = \Omega\left(\frac{1}{t^2}\right)$$

for all $t = 1, \ldots, T$. Equivalently, $T = \Omega(1/\sqrt{\varepsilon})$. [Halbey et al., 2026a]

Main Result

Lower Bound

Theorem (Lower Bound). There exists a compact, smooth, and strongly convex set $\mathcal{X} \subset \mathbb{R}^d$, a smooth and strongly convex function $f : \mathcal{X} \to \mathbb{R}$, such that for all $T \in \mathbb{N}$ there exists a starting point $x_0 \in \mathcal{X}$ such that the FW iterates (with short step or exact line search) satisfy

$$f(x_t) - f(x^*) = \Omega\left(\frac{1}{t^2}\right)$$

for all $t = 1, \ldots, T$. Equivalently, $T = \Omega(1/\sqrt{\varepsilon})$. [Halbey et al., 2026a]

Key points.

1. New proof technique
2. Without assumptions on position of x^*, $\mathcal{O}(1/\sqrt{\varepsilon})$ is *tight* [Garber and Hazan, 2015]
3. Smoothness of $\mathcal{X}$ does not help as our set if smooth
4. Dimension-independent as iterates live in 2D
5. Recent work obtains $\Omega(1/\sqrt{\varepsilon})$ for $d < T$ for *any* LMO-based algorithm but set is not smooth and large-scale regime only [Grimmer and Liu, 2026]

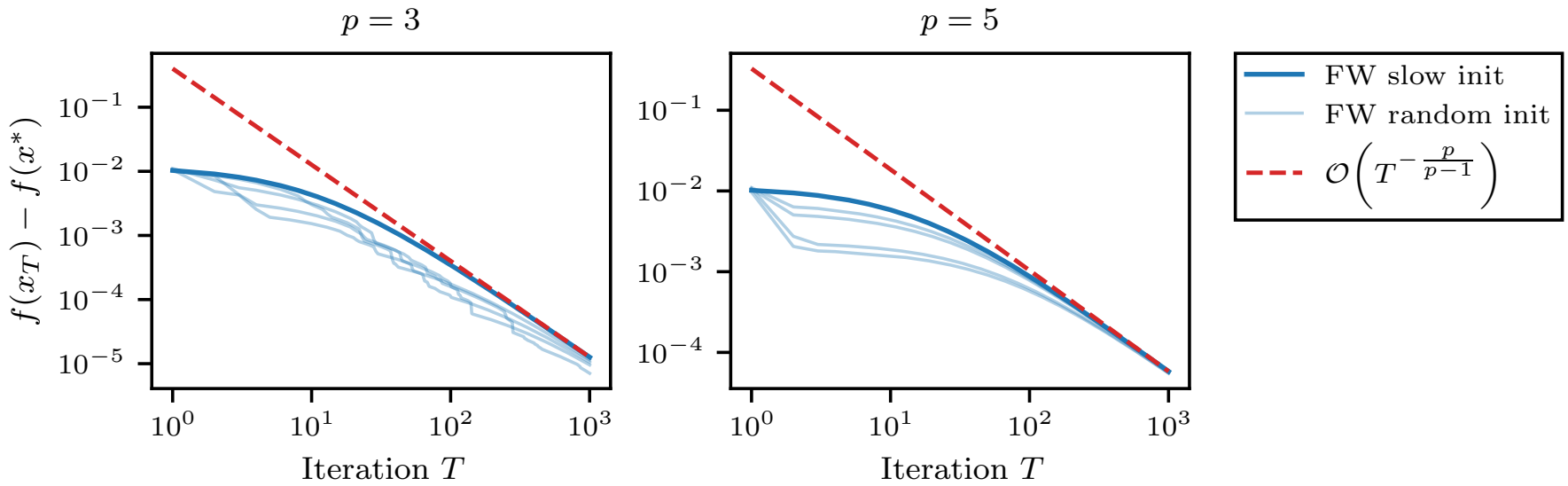
How Curvature Impacts the Rate

Lower Bounds on Uniformly Convex Sets

Theorem. For every $p \geq 3$, the quadratic $f(x) = \|x - e_1\|_2^2$ on the 2D unit ℓ_p ball admits an explicit, fixed starting point with

$$f(x_T) - f^\star = \Omega\left(T^{-p/(p-1)}\right)$$

for exact line search and short steps. This matches the upper bound of Kerdreux et al. [2021].



[Halbey et al., 2026b]

Curvature Alone Beats $1/t$

What if the Objective is only Smooth and Convex?

Theorem. For smooth convex f on a compact uniformly convex set $\mathcal{X}$, vanilla Frank-Wolfe achieves

$$f(x_t) - f^\star = o(1/t).$$

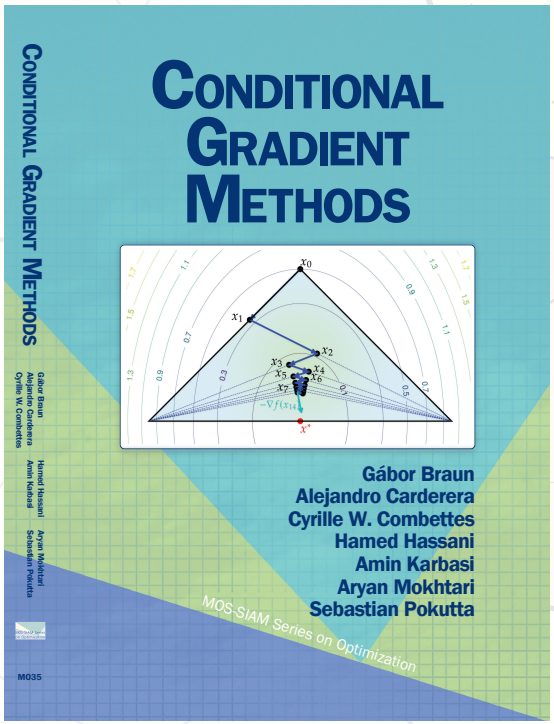
This holds for global short steps, exact line search, and classical open-loop steps $\gamma_t = 2/(t+2)$.

No strong convexity of f is needed. More generally, the result holds under local dual sharpness near the minimizers, a geometric condition on the feasible set and its linear minimization oracle.

[Pokutta, 2026]

If you want to learn more...

Thank you!



Conditional Gradient Methods

Gábor Braun, Alejandro Carderera, Cyrille W Combettes, Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Sebastian Pokutta

https://conditional-gradients.org/
https://arxiv.org/abs/2211.14103

MOS-SIAM Series on Optimization

References I

F. Bach. On the effectiveness of Richardson extrapolation in machine learning. *arXiv preprint 2002.02835v3*, July 2020.

G. Braun, S. Pokutta, and D. Zink. Lazifying Conditional Gradient Algorithms. *Proceedings of the International Conference on Machine Learning (ICML)*, 70:566–575, 2017.

G. Braun, S. Pokutta, D. Tu, and S. Wright. Blended Conditional Gradients: the unconditioning of conditional gradients. *Proceedings of ICML*, 2019a.

G. Braun, S. Pokutta, and D. Zink. Lazifying Conditional Gradient Algorithms. *Journal of Machine Learning Research (JMLR)*, 20(71):1–42, 2019b.

G. Braun, A. Carderera, C. W. Combettes, H. Hassani, A. Karbasi, A. Mokhtari, and S. Pokutta. *Conditional Gradient Methods*. MOS-SIAM Series on Optimization, 1 2025. ISBN 978-1-61197-855-1.

A. Carderera, J. Diakonikolas, C. Y. Lin, and S. Pokutta. Parameter-free Locally Accelerated Conditional Gradients. *Proceedings of ICML*, 2 2021.

C. W. Combettes, C. Spiegel, and S. Pokutta. Projection-Free Adaptive Gradients for Large-Scale Optimization. *preprint*, 10 2020.

J. Diakonikolas, A. Carderera, and S. Pokutta. Locally Accelerated Conditional Gradients. *Proceedings of AISTATS*, 108:1737–1747, 2020.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

R. M. Freund, P. Grigas, and R. Mazumder. An extended Frank-Wolfe method with “in-face” directions, and its application to low-rank matrix completion. *SIAM Journal on Optimization*, 27(1):319–346, 2017.

D. Garber and E. Hazan. Playing non-linear games with linear oracles. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 420–428. IEEE, 2013.

D. Garber and E. Hazan. Faster rates for the Frank-Wolfe method over strongly-convex sets. In *32nd International Conference on Machine Learning, ICML 2015*, 2015.

D. Garber and O. Meshi. Linear-memory and decomposition-invariant linearly convergent conditional gradient algorithm for structured polytopes. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1001–1009. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6115-linear-memory-and-decomposition-invariant-linearly-convergent-conditional-gradient-algorithm-for-structured-polytopes.pdf.

B. Grimmer and N. Liu. Lower bounds for linear minimization oracle methods optimizing over strongly convex sets. *arXiv preprint arXiv:2602.22608*, 2026.

J. Guélat and P. Marcotte. Some comments on Wolfe’s ‘away step’. *Mathematical Programming*, 35(1):110–119, 1986.

J. Halbey, D. Deza, M. Zimmer, C. Roux, B. Stellato, and S. Pokutta. Lower Bounds for Frank-Wolfe on Strongly Convex Sets. *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 306:39468–39494, 5 2026a. doi: 10.48550/arXiv.2602.04378.

J. Halbey, C. Roux, and S. Pokutta. Curvature-Dependent Lower Bounds for Frank-Wolfe. *to appear in Proceedings of NeurIPS*, 9 2026b. doi: 10.48550/arXiv.2605.10595.

References II

E. Hazan and S. Kale. Projection-free online learning. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

E. Hazan and H. Luo. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pages 1263–1271, 2016.

M. Jaggi. Revisiting Frank-Wolfe: projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 427–435, 2013.

T. Kerdreux, A. d'Aspremont, and S. Pokutta. Restarting Frank-Wolfe. *Proceedings of AISTATS*, 2019.

T. Kerdreux, A. d'Aspremont, and S. Pokutta. Projection-Free Optimization on Uniformly Convex Sets. *Proceedings of AISTATS*, 1 2021.

T. Kerdreux, A. d'Aspremont, and S. Pokutta. Local and Global Uniform Convexity Conditions. In *Data Science and Optimization*, volume 91 of *Fields Institute Communications*, pages 85–120. Springer, 1 2026. doi: 10.1007/978-3-032-03844-9_4.

S. Lacoste-Julien. Convergence rate of Frank-Wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 496–504. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5925-on-the-global-linear-convergence-of-frank-wolfe-optimization-variants.pdf.

G. Lan. The complexity of large-scale convex programming under a linear optimization oracle. *arXiv preprint arXiv:1309.5550*, 2013.

G. Lan and Y. Zhou. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016.

E. S. Levitin and B. T. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6(5):1–50, 1966.

A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, 1983.

S. Pokutta. Frank-Wolfe Beyond 1/t Convergence. *to appear in Proceedings of NeurIPS*, 9 2026. doi: 10.48550/arXiv.2604.28006.

S. J. Reddi, S. Sra, B. Póczos, and A. Smola. Stochastic Frank-Wolfe methods for nonconvex optimization. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1244–1251. IEEE, 2016.

P. Wolfe. Convergence theory in nonlinear programming. In *Integer and Nonlinear Programming*, pages 1–36. North-Holland, Amsterdam, 1970.