APFOCS22, August 12th

p-norm regression with acceleration

A FEW STORIES

1) A NEW APPROACH TO NON-SMOOTH OPTIMIZATION?

2) ACCELERATION FOR NON-SMOOTH OPTIMIZATION?

3) FASTER FLOW ALGORITHMS

A FEW STORIES

1) A NEW APPROACH TO NON-SMOOTH OPTIMIZATION?
(LESS SMOOTH)

2) ACCELERATION FOR NON-SMOOTH OPTIMIZATION?
(LESS SMOOTH)

3) FASTER FLOW ALGORITHMS

OPTIMIZATION: One solution to all your problems?

Solution domain $D \subseteq \mathbb{R}^d$

Cost function $f : D \to \mathbb{R}$

$$\min_{\underline{x} \in D} f(x)$$

OPTIMIZATION: One solution to all your problems?

Solution domain $D \subseteq \mathbb{R}^d$

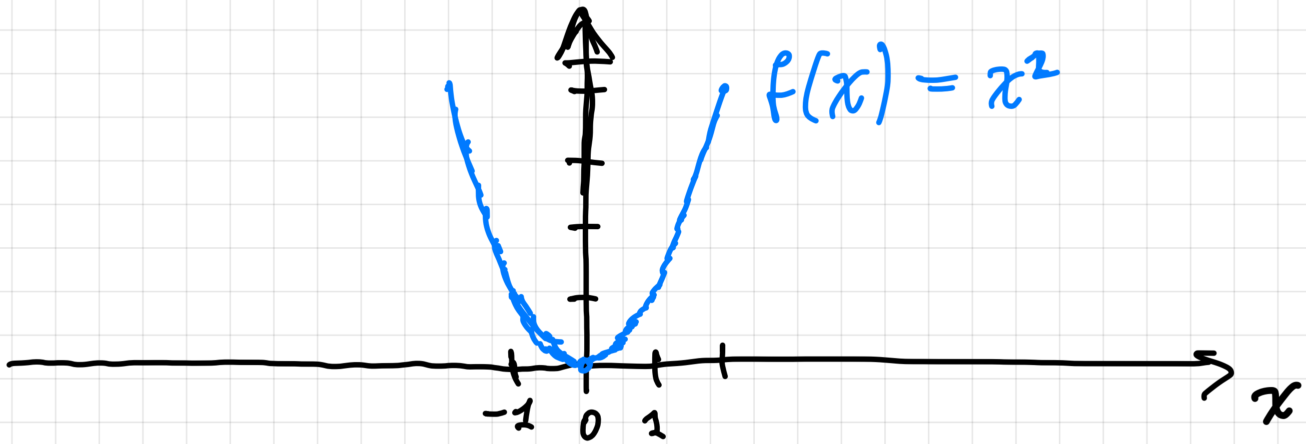
Cost function $f : D \to \mathbb{R}$

$$\min_{\underline{x} \in D} f(x)$$

IDEA: START with initial solution $\underline{x}_0$

SOMEHOW IMPROVE IT ??!

WHAT IS SMOOTH OPTIMIZATION?

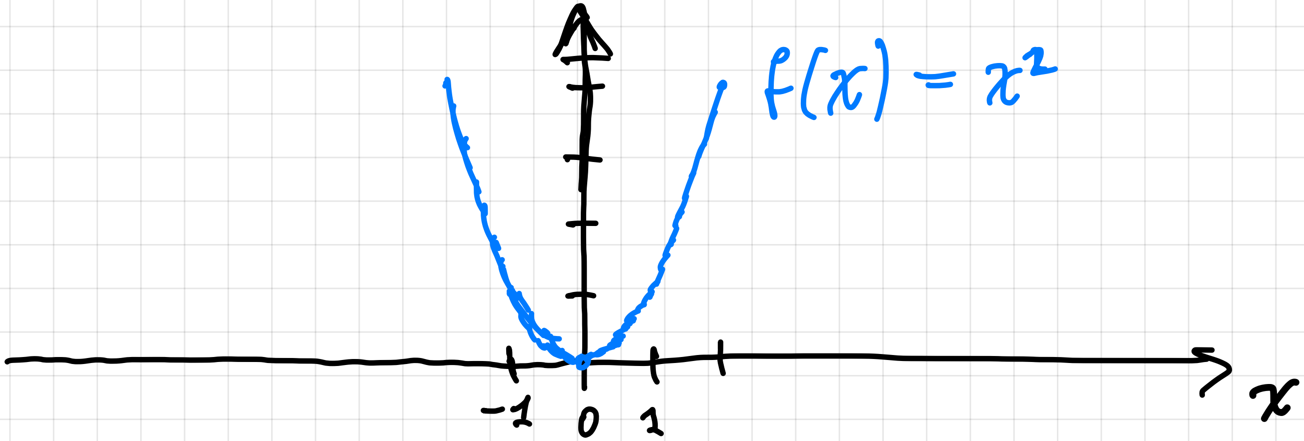


- Gradient of f exists
  and is not changing too quickly

- Taylor series gives good approximation

$$f(\underline{x}+\underline{\delta}) \simeq f(x) + f'(x)\cdot\delta + f''(x)\cdot\frac{\delta^2}{2} + O(\delta^3)$$

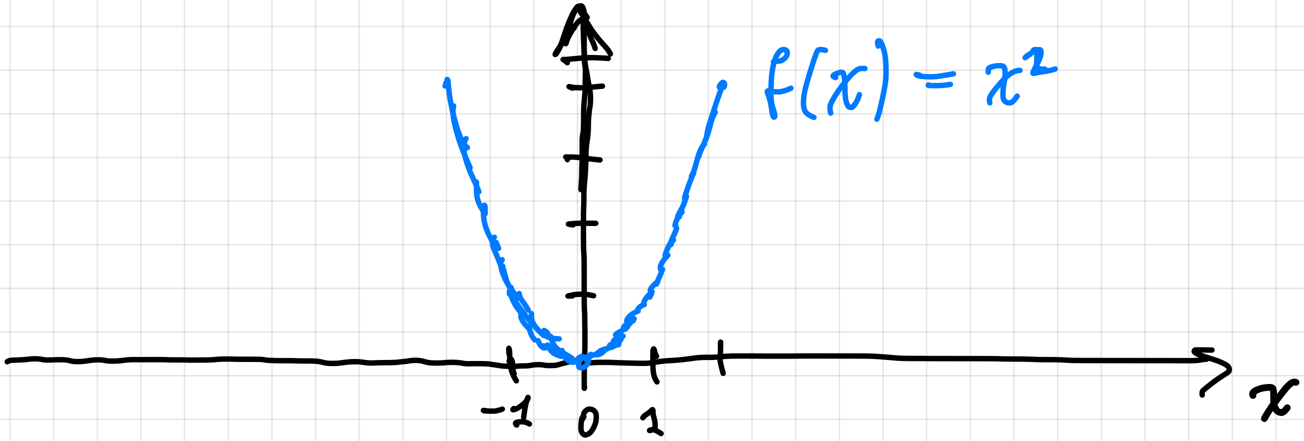
WHAT IS SMOOTH OPTIMIZATION?



- Gradient of f exists and is not changing too quickly
- Taylor series gives good approximation

$$f(\underline{x}+\underline{\delta}) \approx f(\underline{x}) + \sum_i \underline{\delta}(i) \frac{\partial f(\underline{x})}{\partial \underline{x}(i)} + \frac{1}{2}\sum_{i,j} \underline{\delta}(i)\,\underline{\delta}(j) \frac{\partial^2 f(\underline{x})}{\partial \underline{x}(i)\,\partial \underline{x}(j)} + O(\|\underline{\delta}\|^3).$$

WHAT IS SMOOTH OPTIMIZATION?



- Gradient of f exists and is not changing too quickly

- Taylor series gives good approximation

$$f(\underline{x}+\underline{\delta}) \approx f(\underline{x}) + \sum_i \underline{\delta}(i) \frac{\partial f(\underline{x})}{\partial \underline{x}(i)} \cancel{+ \frac{1}{2}\sum_{i,j} \underline{\delta}(i)\underline{\delta}(j) \frac{\partial^2 f(\underline{x})}{\partial \underline{x}(i)\, \partial \underline{x}(j)}} + O(\|\underline{\delta}\|^3).$$

- Suggests $\underline{\delta}(i) = -\frac{\partial f(\underline{x})}{\partial \underline{x}(i)} \cdot \varepsilon$ good for small ε?

GRADIENT DESCENT

$$f(\underline{x}+\underline{\delta}) \approx f(\underline{x}) + \sum_i \underline{\delta}(i) \frac{\partial f(\underline{x})}{\partial \underline{x}(i)} + \frac{1}{2}\sum_{i,j} \underline{\delta}(i)\underline{\delta}(j) \frac{\partial^2 f(\underline{x})}{\partial \underline{x}(i)\,\partial \underline{x}(j)} + O(\|x\|^3).$$

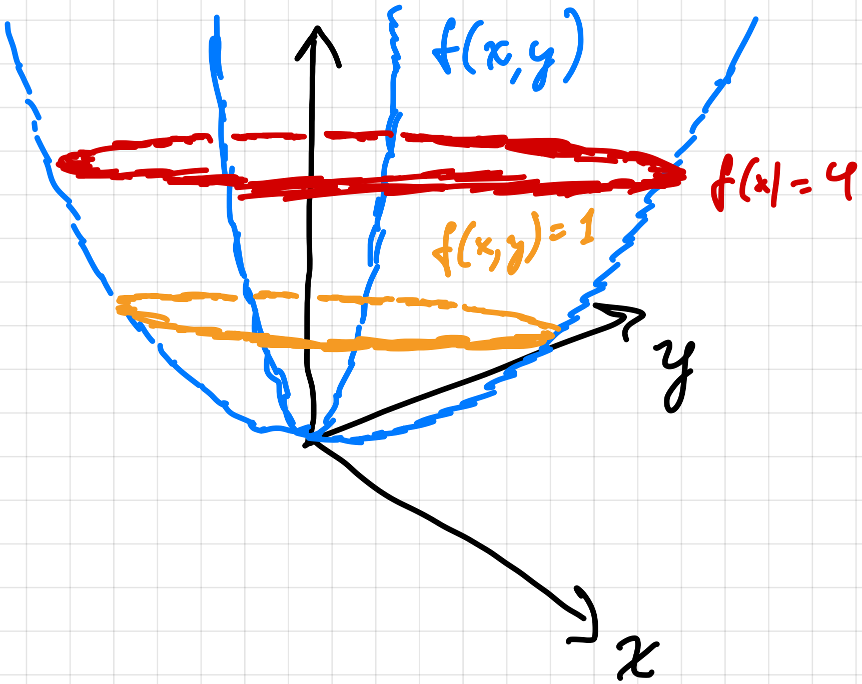
- Suggests $\underline{\delta}(i) = -\frac{\partial f(\underline{x})}{\partial \underline{x}(i)} \cdot \varepsilon$ good for small ε?

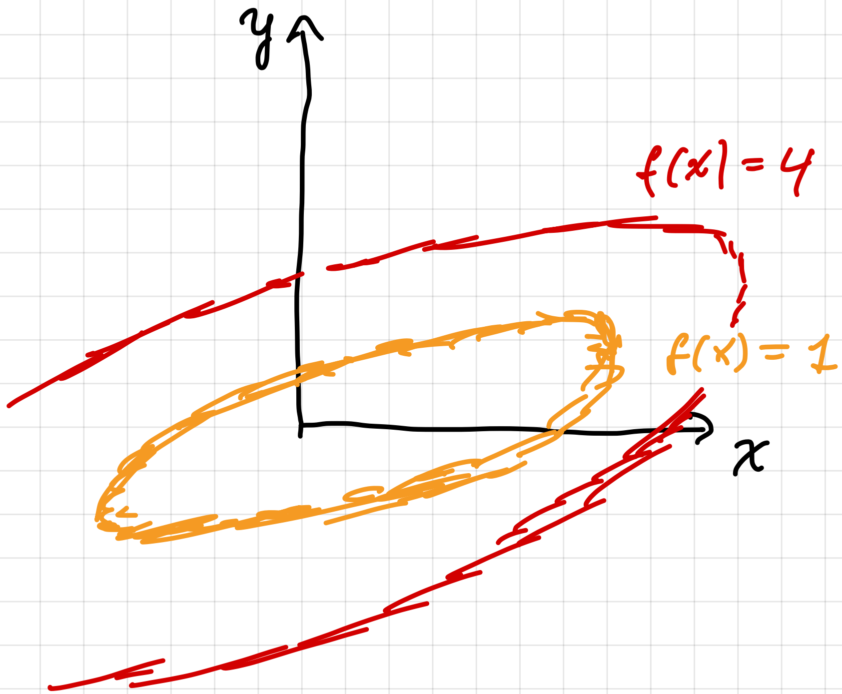
- $$\underline{\nabla} f(\underline{x}) = \begin{pmatrix} \frac{\partial f(\underline{x})}{\partial \underline{x}(1)} \\ \vdots \\ \frac{\partial f(\underline{x})}{\partial \underline{x}(d)} \end{pmatrix}$$

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

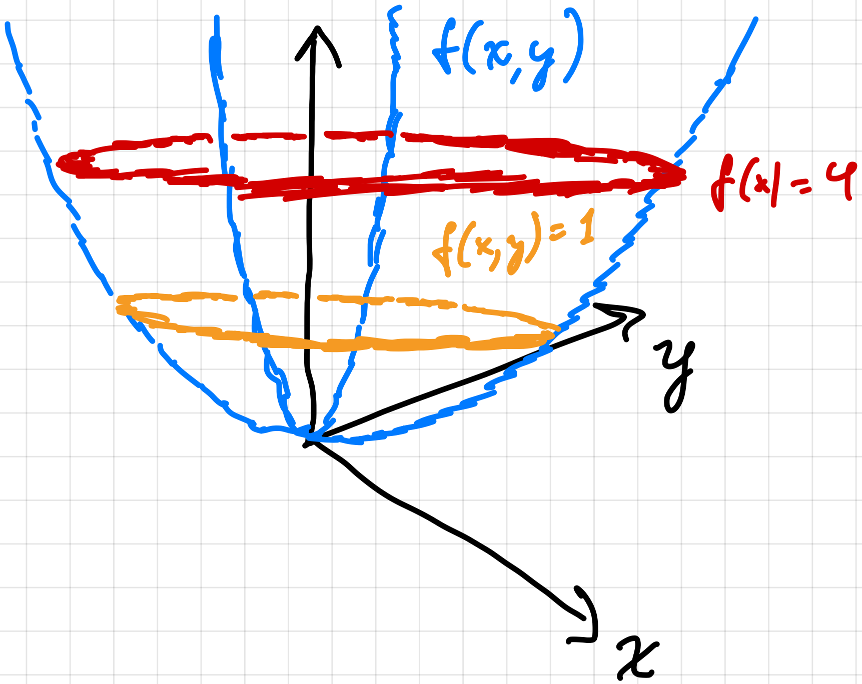
- $f(\underline{x}_{k+1}) \approx f(\underline{x}_k) - \varepsilon \|\underline{\nabla} f(\underline{x}_k)\|_2^2$ for small ε

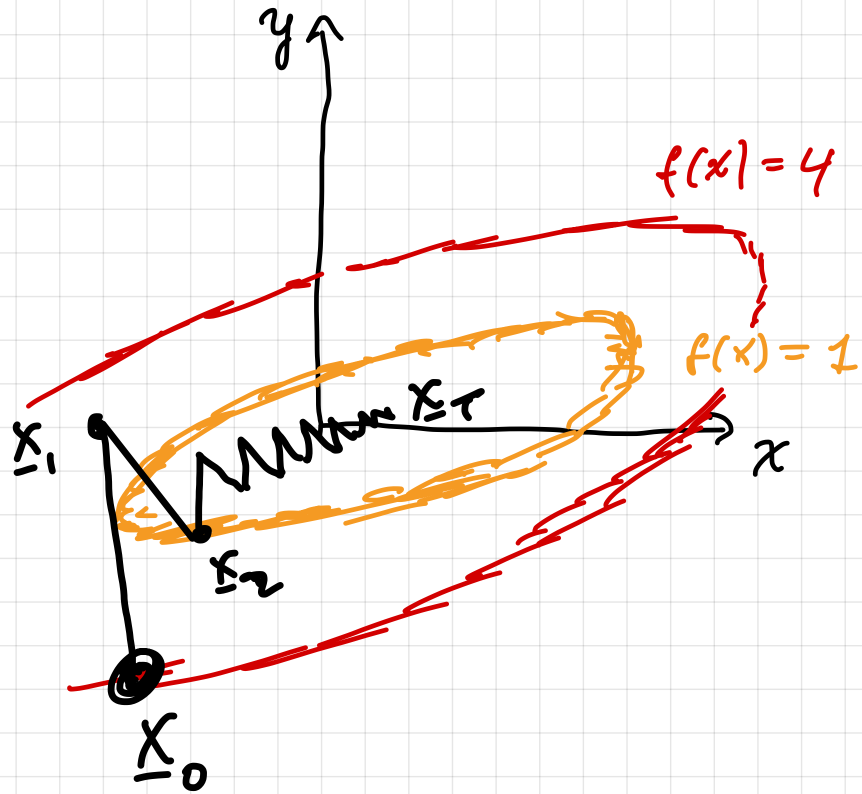
WHAT MAKES SMOOTH OPTIMIZATION HARD?





WHAT MAKES SMOOTH OPTIMIZATION HARD?

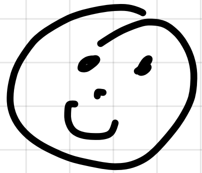


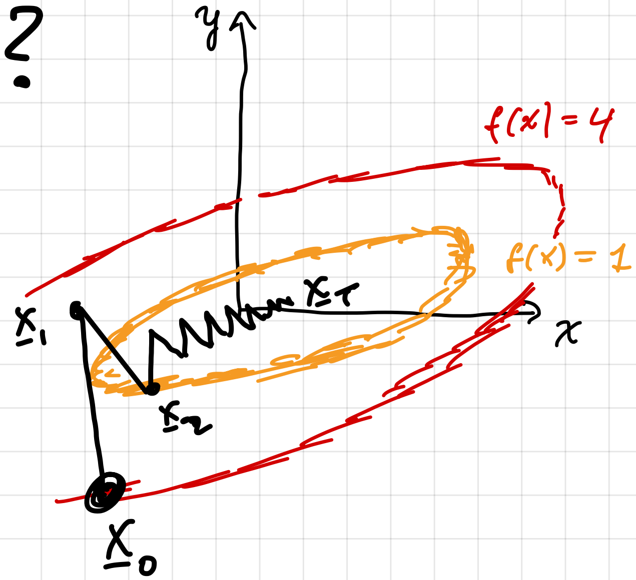


- Elongated shape → zig-zagging → slow ☹

WHAT MAKES SMOOTH OPTIMIZATION HARD?

- WHAT CAN WE DO ABOUT IT?
  - "Momentum" ⇒ acceleration

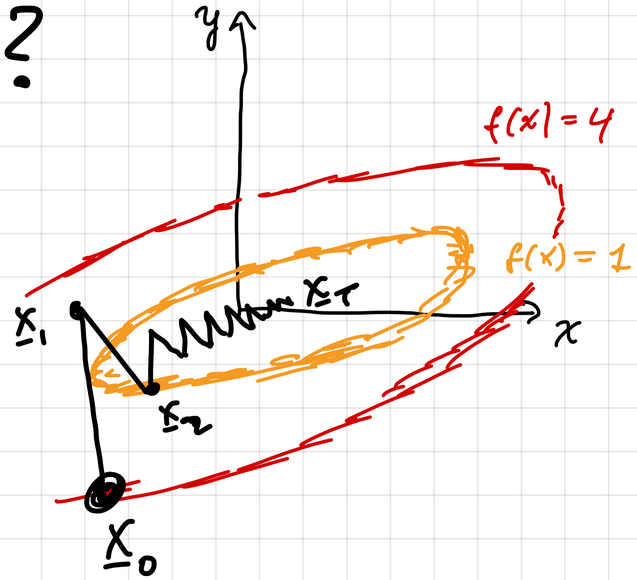




WHAT MAKES SMOOTH OPTIMIZATION HARD?

- WHAT CAN WE DO ABOUT IT?

→ Use the Hessian?



Newton step

$$f(\underline{x}+\underline{\delta}) \simeq f(\underline{x}) + \sum_i \underline{\delta}(i) \frac{\partial f(\underline{x})}{\partial \underline{x}(i)} + \sum_{i,j} \underline{\delta}(i)\underline{\delta}(j) \frac{\partial^2 f(\underline{x})}{\partial \underline{x}(i)\, \partial \underline{x}(j)} + O(\|\underline{\delta}\|^3).$$

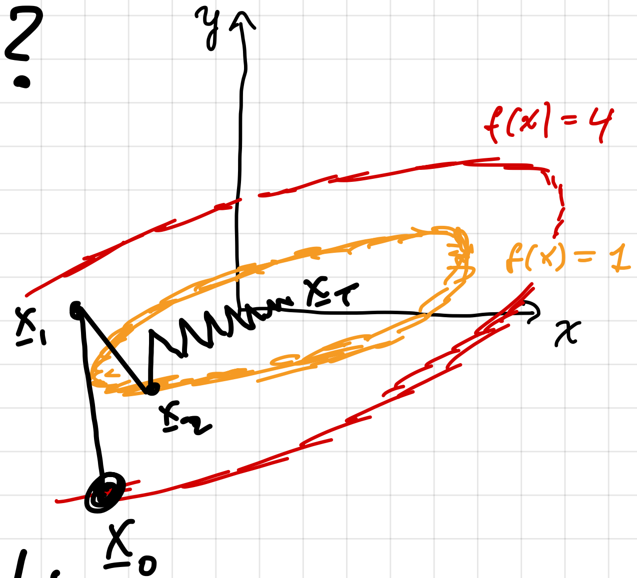
$$= f(\underline{x}) + \underbrace{\underline{\nabla} f(\underline{x})^T \underline{\delta} + \frac{1}{2}\underline{\delta}\, \underline{\underline{\nabla}}^2 f(\underline{x})\, \underline{\delta}}_{\text{Minimize wrt. } \underline{\delta}\text{? (And scale down?)}} + O(\|\underline{\delta}\|^3)$$

WHAT MAKES SMOOTH OPTIMIZATION HARD?

- WHAT CAN WE DO ABOUT IT?

→ Use the Hessian?

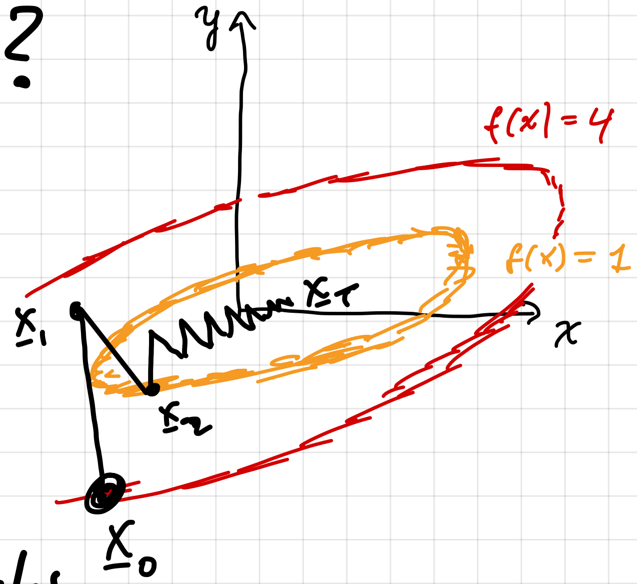
$$\underline{\underline{\nabla}}^2 f(\underline{x})\, \underline{\delta}^* = -\nabla f(\underline{x})$$



→ $\underline{\delta}^*$ by solving a linear equation

WHAT MAKES SMOOTH OPTIMIZATION HARD?

- WHAT CAN WE DO ABOUT IT?



→ Use the Hessian?

$$\underline{\underline{\nabla}}^2 f(\underline{x})\, \underline{\delta}^* = -\nabla f(\underline{x})$$

→ $\underline{\delta}^*$ by solving a linear equation

→ Solve w. Gaussian elimination "DIRECT METHODS"

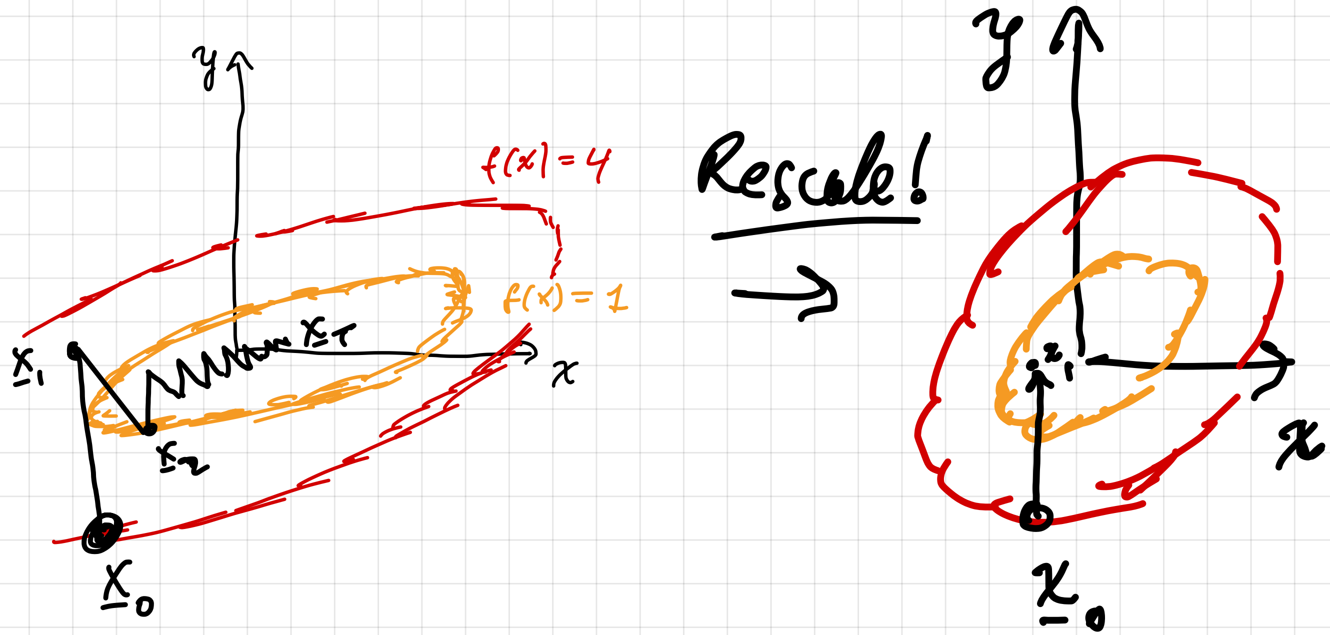
WHAT MAKES SMOOTH OPTIMIZATION HARD?

- WHAT CAN WE DO ABOUT IT?

→ Use the Hessian?

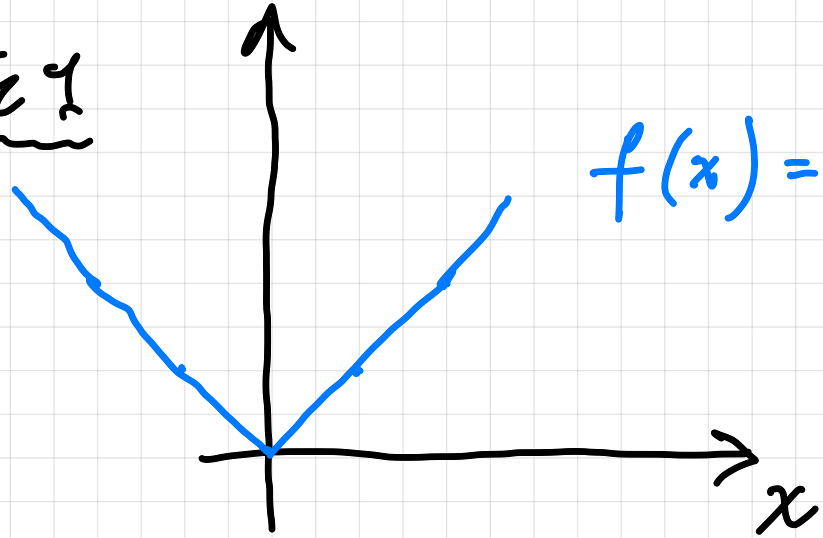
$$\underline{\underline{\nabla}}^2 f(\underline{x})\, \underline{\delta}^* = -\nabla f(\underline{x})$$

→ $\underline{\delta}^*$ by solving a linear equation



WHAT IS NON-SMOOTH OPTIMIZATION?

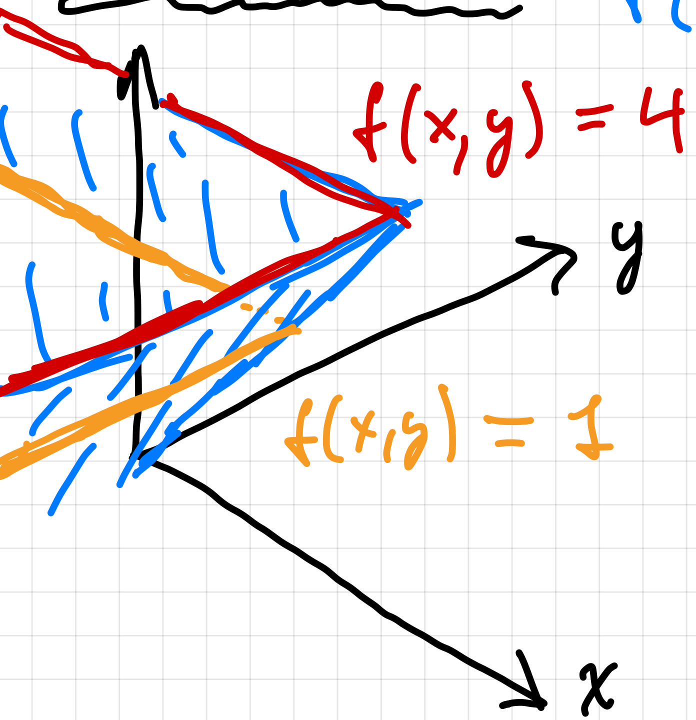
EXAMPLE 1

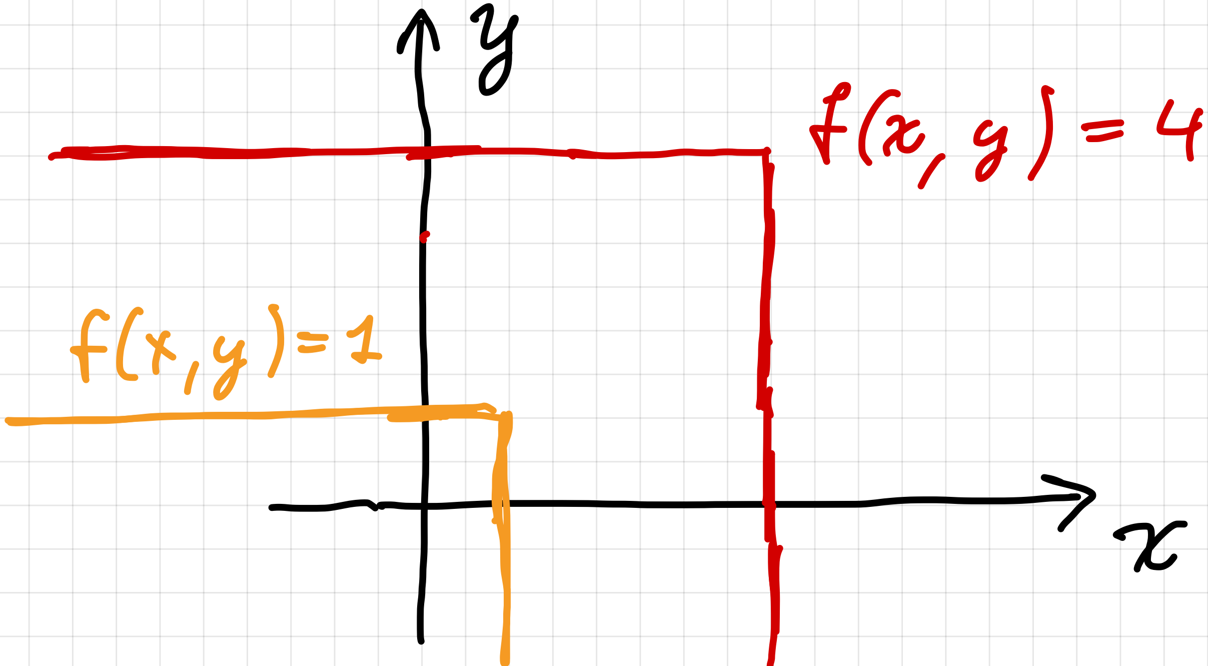


$f(x) = |x|$

EXAMPLE 2

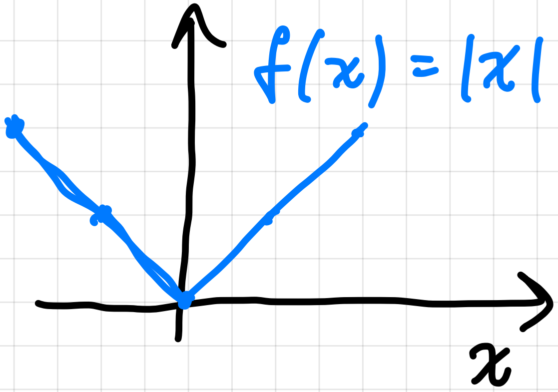
$f(x, y) = \max(x, y)$

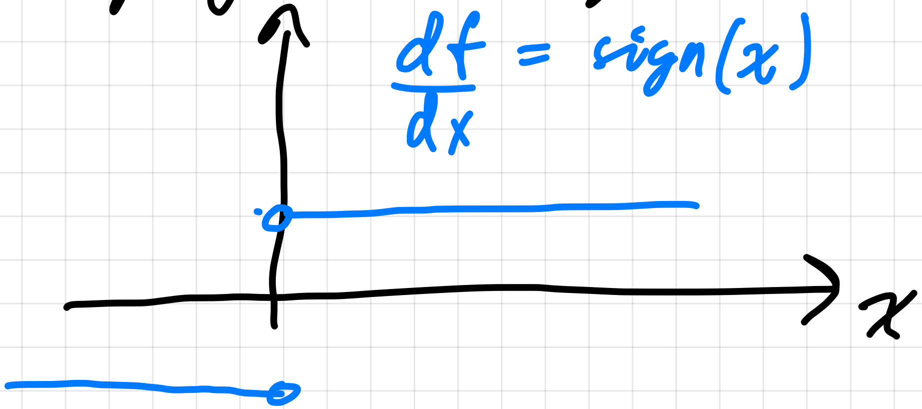




WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- Gradient changing suddenly





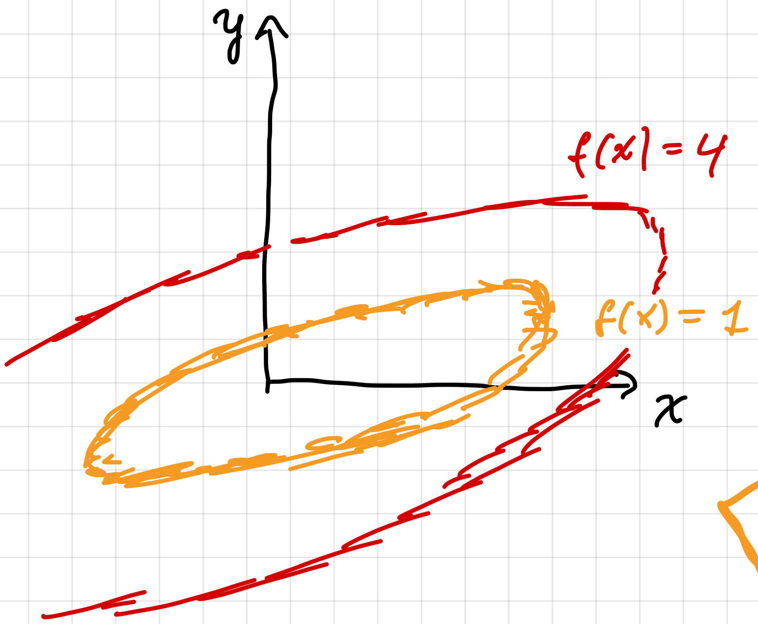
- Little information from gradient, esp. in high dim.

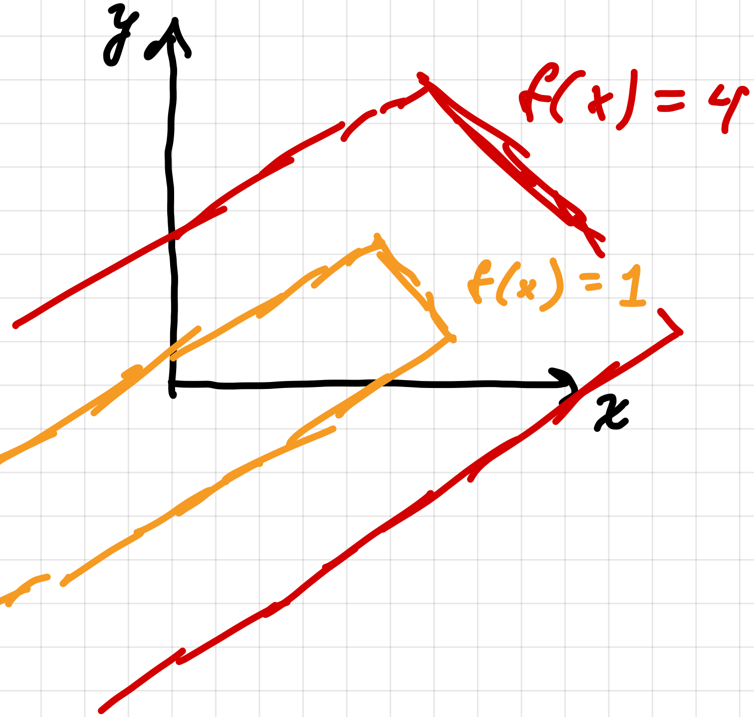
E.g. $f(\underline{x}) = \max_i \underline{x}(i)$

$\underline{\nabla} f(\underline{x})$ usually non-zero only in 1 coordinate.

WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

+ Elongation is still a problem (as for smooth case)

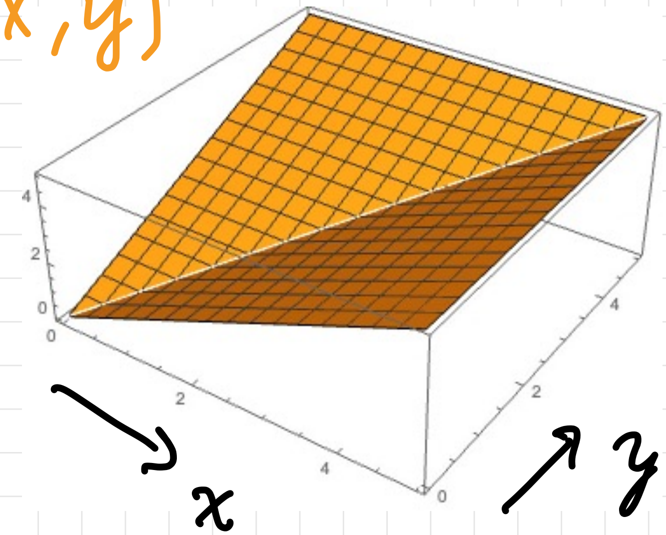




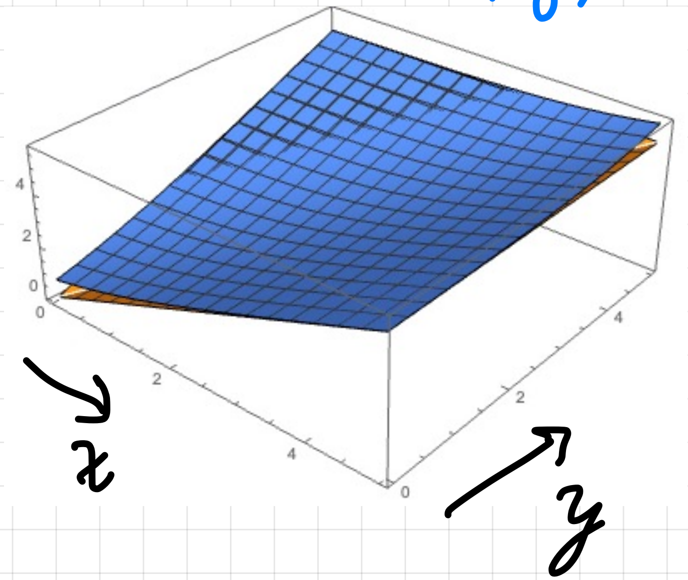
WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- How can we fix it?
- Standard approach #1: SMOOTHING

$\max(x, y)$



$\text{softmax}(x, y) = \log(e^x + e^y)$



WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- How can we fix it?
- Standard approach #1: SMOOTHING
- Smoothness/error trade-off gives low accuracy

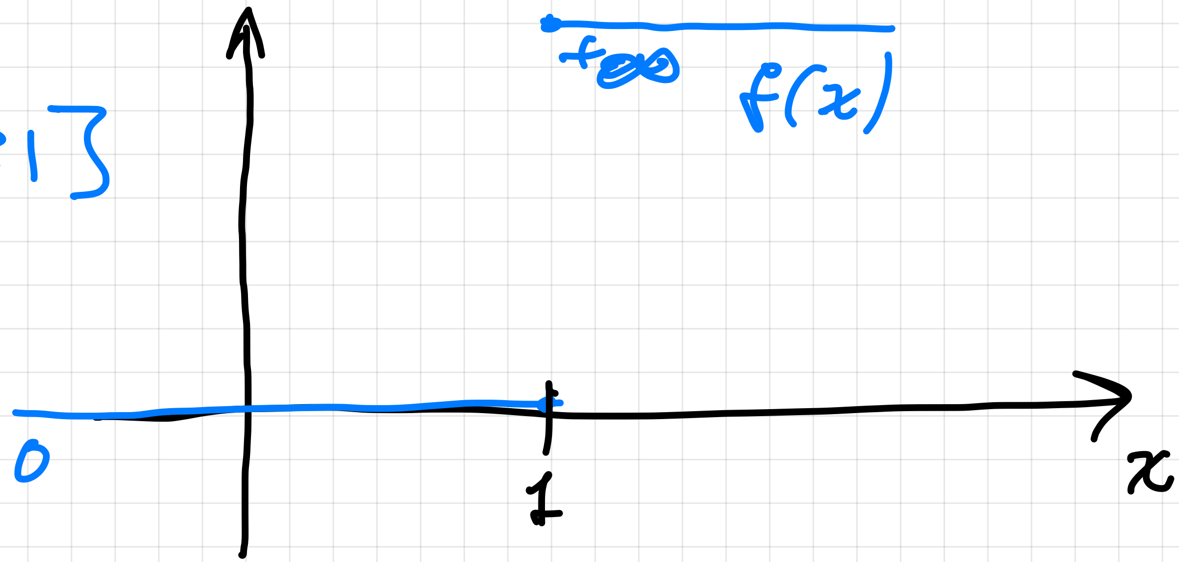
  $\Omega\left(\frac{1}{\varepsilon}\right)$ steps for ε error in value

WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- Standard approach # 2: HOMOTOPY

  $\approx$ weaker smoothing + transform the problem

$$f(x) = \infty \cdot 1_{[x \geq 1]}$$



WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

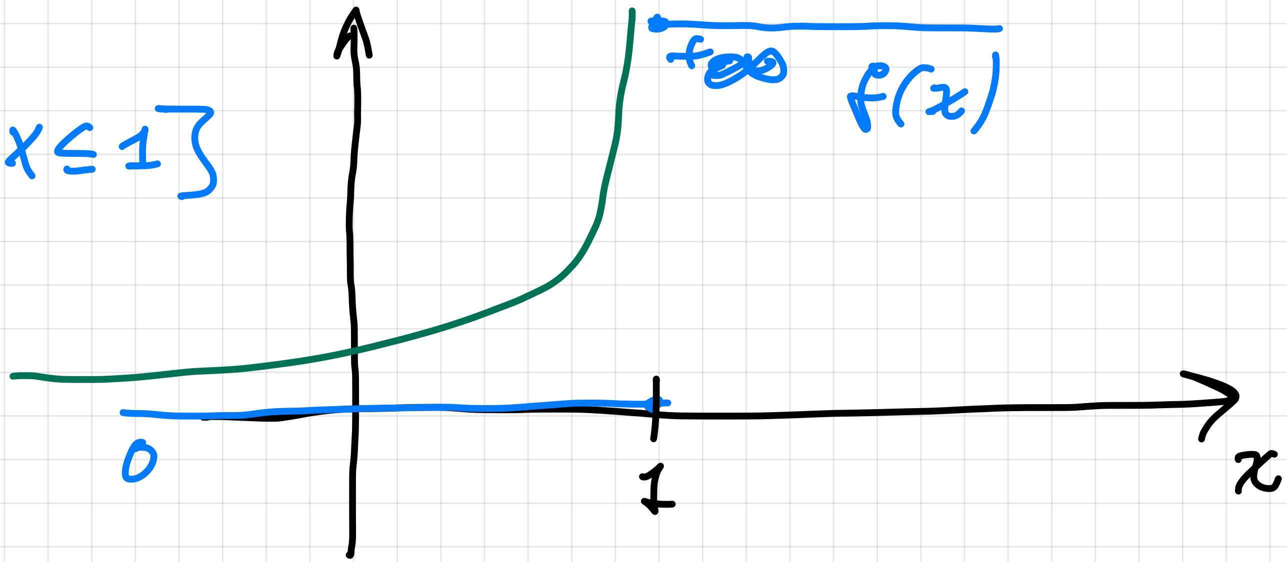
- Standard approach # 2: HOMOTOPY

  ≈ weaker smoothing + transform the problem

$$f(x) = \infty \cdot 1_{[x \leq 1]}$$

$$\approx c \log\left(\frac{1}{1-x}\right)$$

Make c small?



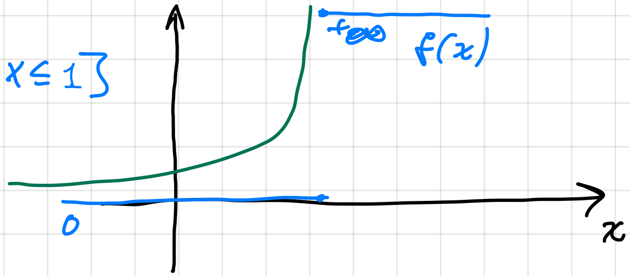
WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- Standard approach # 2: <u>HOMOTOPY</u>

$f(x) = \infty \cdot 1_{[x \leq 1]}$

$\approx c \log\left(\frac{1}{1-x}\right)$

make c small?



- Plan: to solve

$$\boxed{\begin{aligned} &\min_{x,y} \; -(x+y) \\ &x \leq 1 \\ &y \leq 1 \end{aligned}}$$

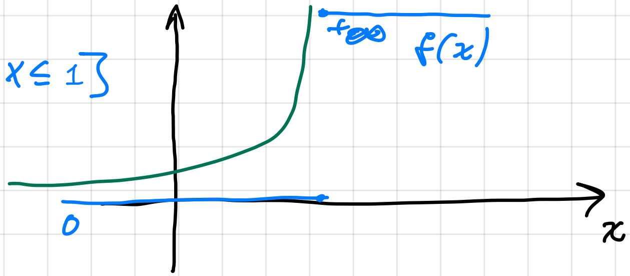
WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- Standard approach # 2: HOMOTOPY

$f(x) = \infty \cdot 1_{[x \le 1]}$

$\approx c \log\left(\frac{1}{1-x}\right)$

make c small?

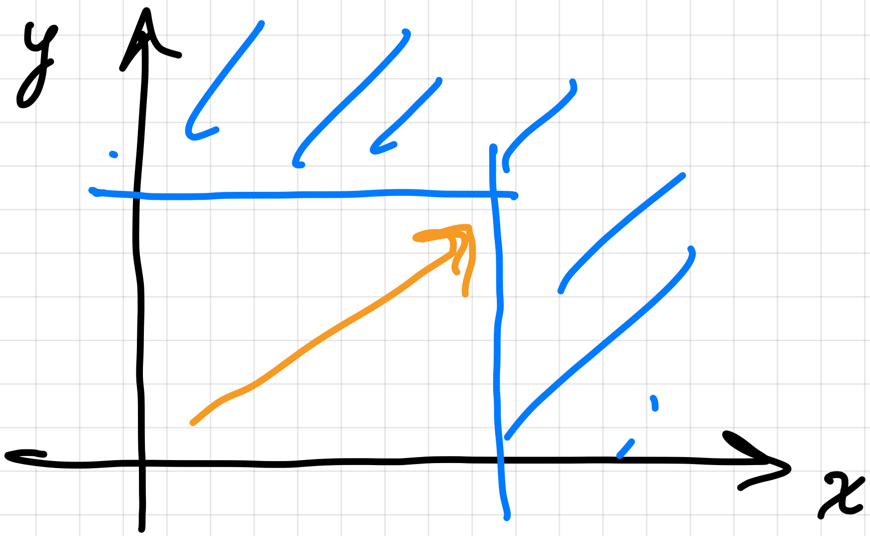


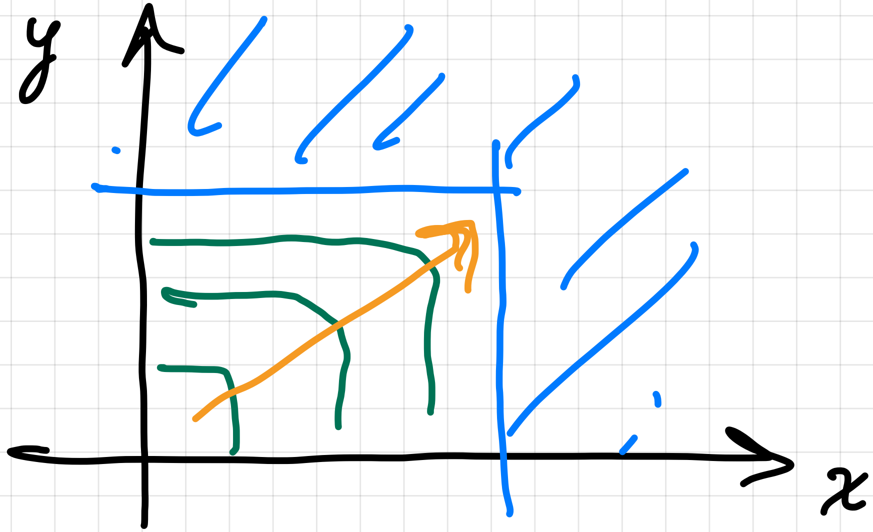
- Plan: to solve

$$\min_{x,y} \; -(x+y) \qquad x \le 1 \qquad y \le 1$$

convert $\Longrightarrow$

$$\min_{x,y} \; -(x+y) + c\log\left(\frac{1}{1-x}\right) + c\log\left(\frac{1}{1-y}\right)$$

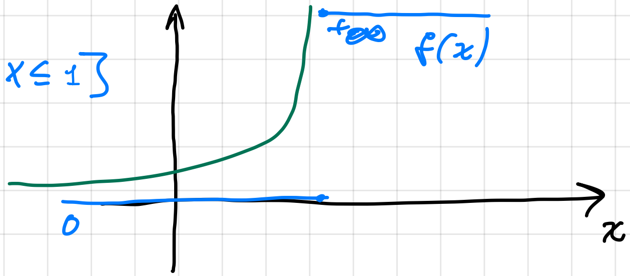




WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

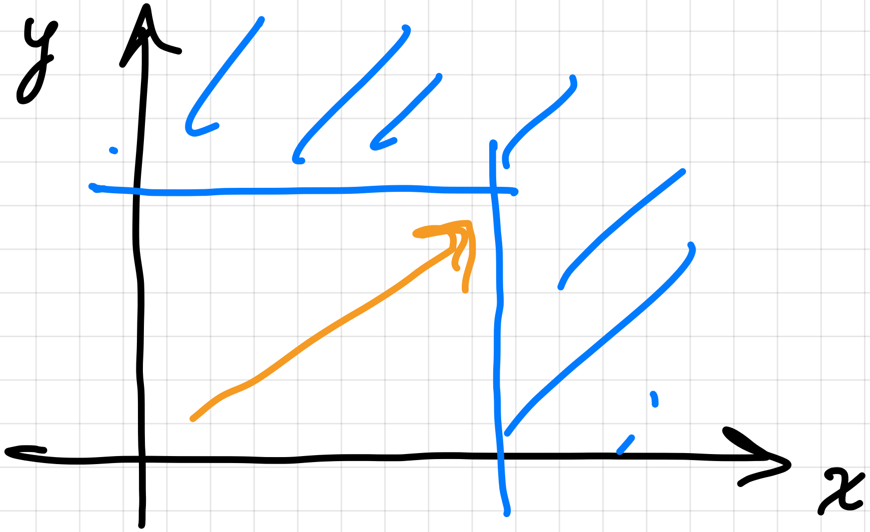
- Standard approach # 2: HOMOTOPY

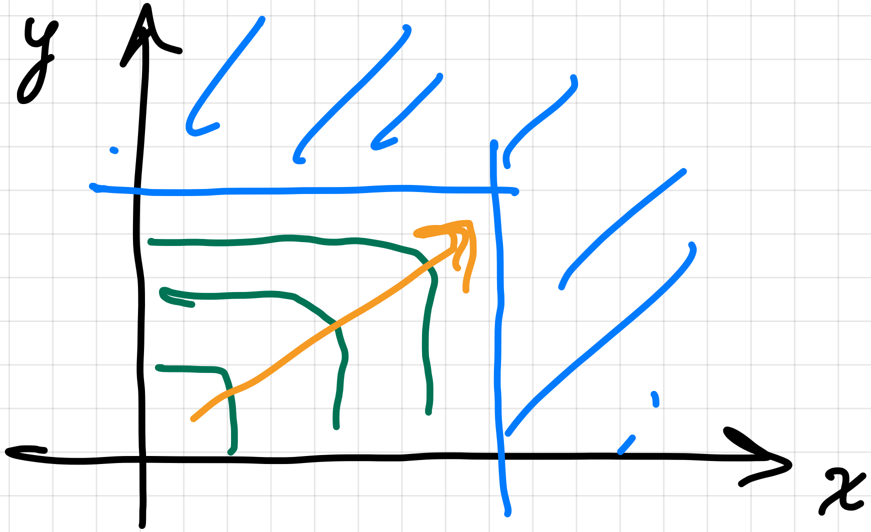
$f(x) = \infty \cdot 1_{[x \le 1]}$
$\approx c \log\left(\frac{1}{1-x}\right)$
make c small?



- Plan: to solve

$$\boxed{\begin{aligned} &\min_{x,y} \; -(x+y) \\ &x \le 1 \\ &y \le 1 \end{aligned}} \xrightarrow{\text{convert}} \underbrace{\boxed{\min_{x,y} \; -(x+y) + c\log\left(\frac{1}{1-x}\right) + c\log\left(\frac{1}{1-y}\right)}}_{\text{call this } f_c}$$





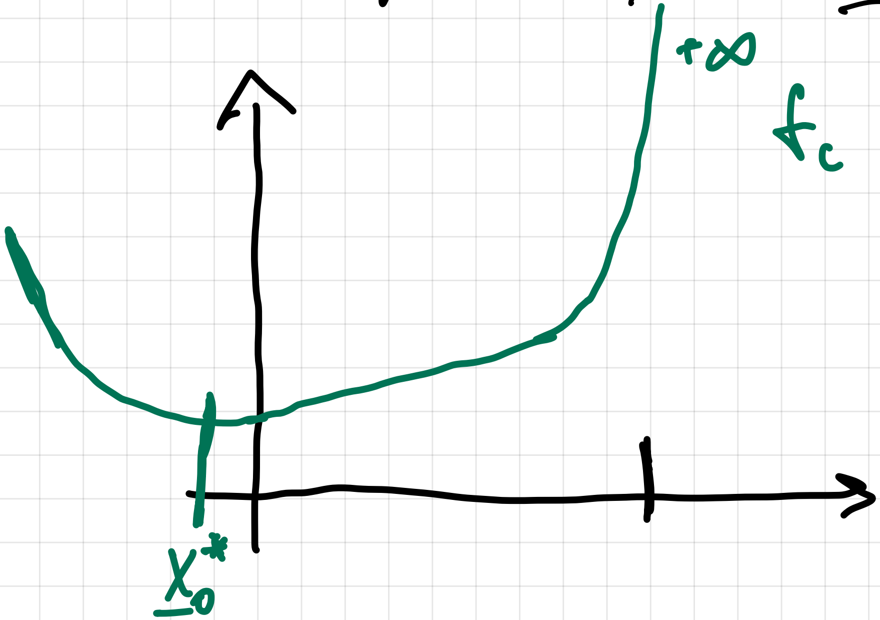
WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

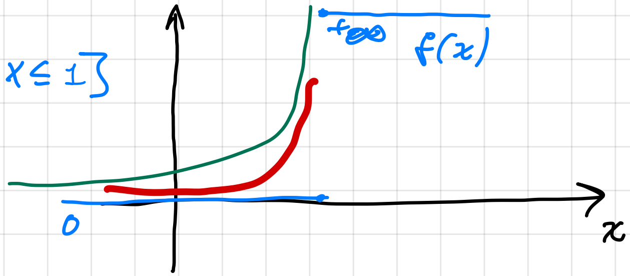
- Standard approach # 2: <u>HOMOTOPY</u>

$$f(x) = \infty \cdot 1_{[x \leq 1]}$$

$$\approx c \log\left(\frac{1}{1-x}\right)$$

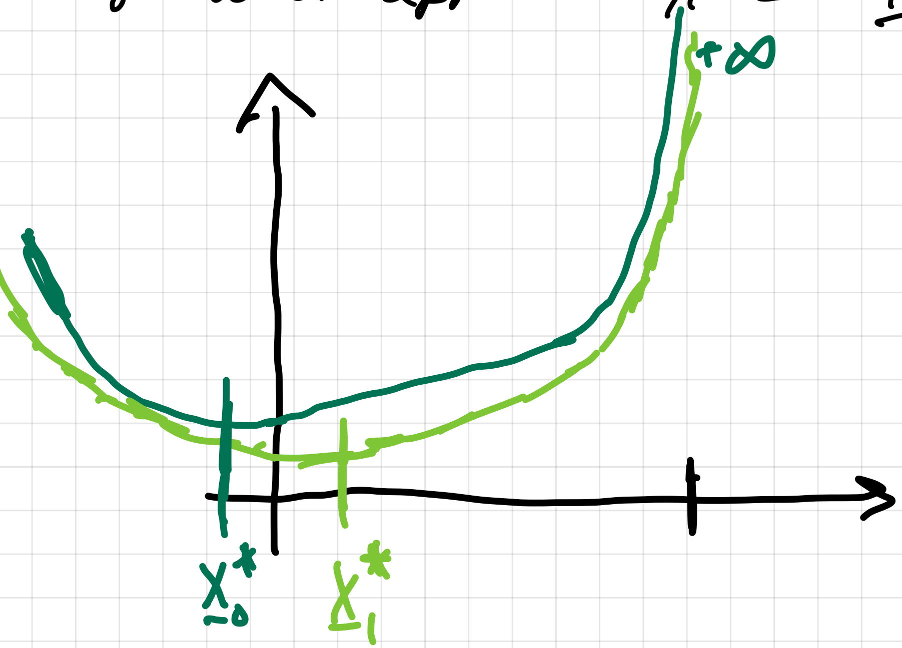
make c small?





WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

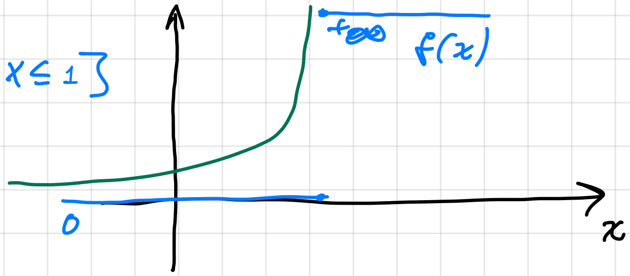
- Standard approach # 2: <u>HOMOTOPY</u>



$f(x) = \infty \cdot 1_{[x \leq 1]}$

$\approx c \log\left(\frac{1}{1-x}\right)$

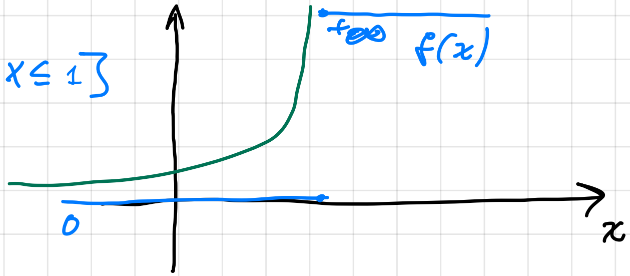
Make c small?

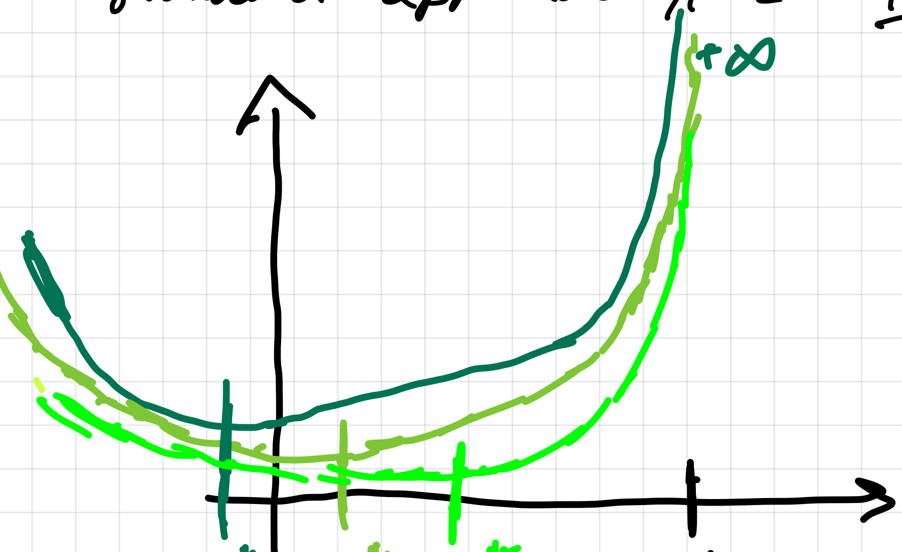


WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- Standard approach # 2: HOMOTOPY

$$f(x) = \infty \cdot 1_{[x \leq 1]}$$
$$\approx c \log\left(\frac{1}{1-x}\right)$$
make c small?



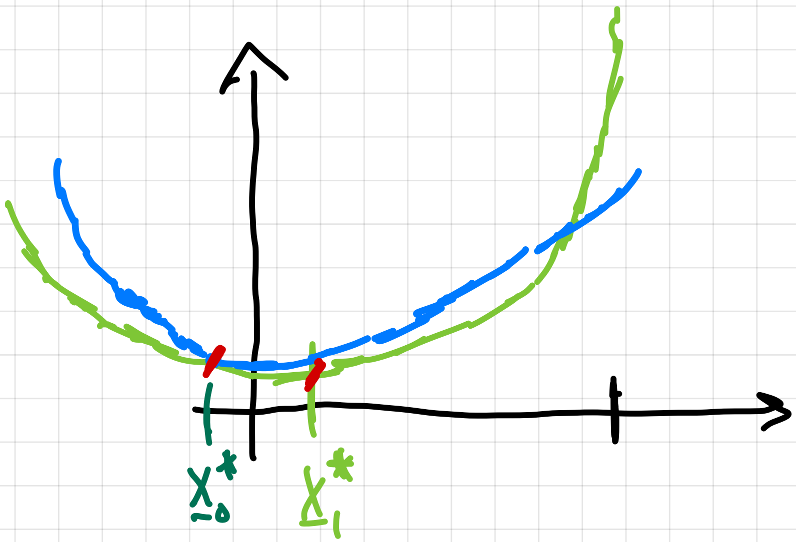


$x_0^* \quad x_1^* \quad x_2^* \cdots x_t^* \to x^*$ (true optimum)

WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- Standard approach # 2: HOMOTOPY
- How do we find each new point $\underline{x}^*_i$?
- LOCAL quadratic approximation

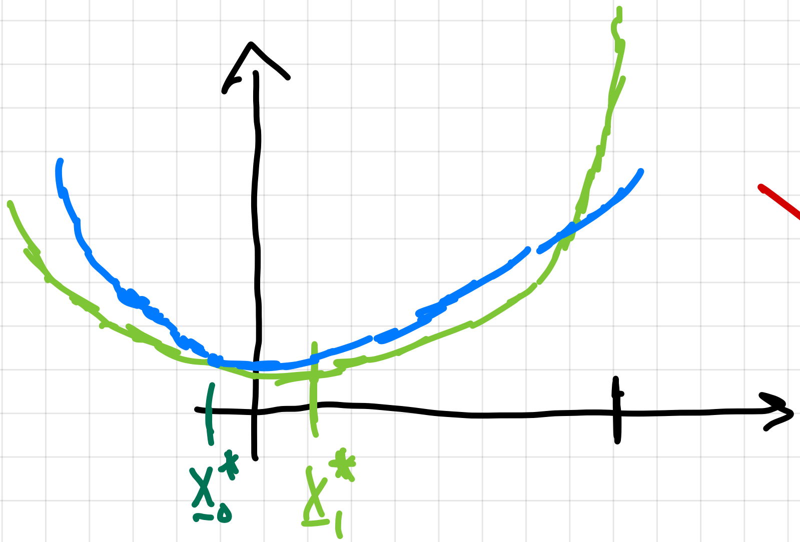
$$f(\underline{x}^*_0 + \underline{\delta}) \approx f(\underline{x}^*_0) + \underline{\nabla} f(\underline{x}^*_0) \cdot \underline{\delta} + \frac{1}{2} \underline{\delta} \cdot \nabla^2 f(\underline{x}^*_0)\, \underline{\delta}$$

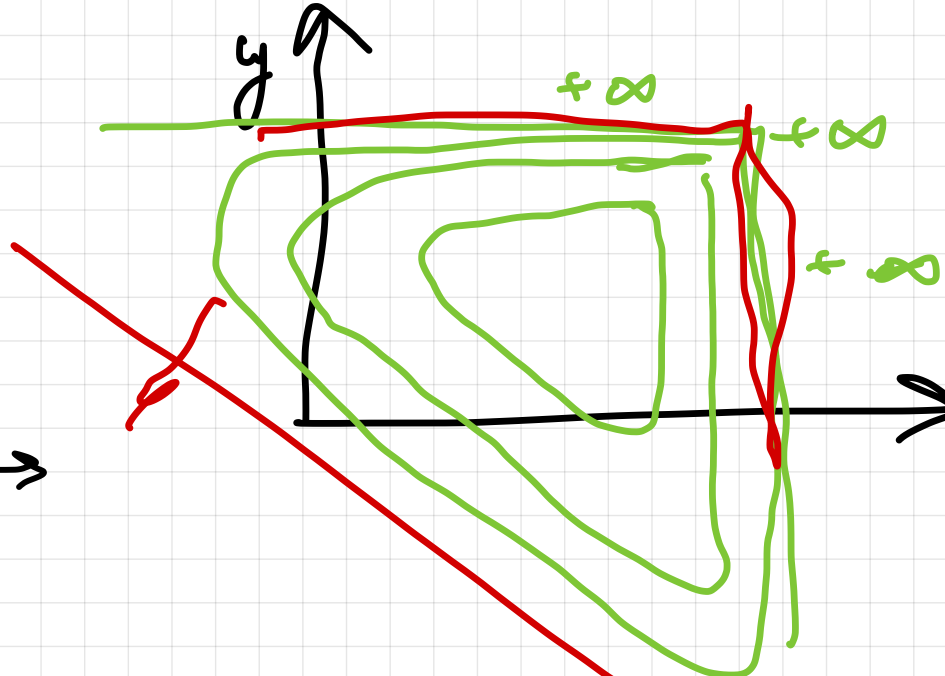


WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- Standard approach # 2: HOMOTOPY
- How do we find each new point $\underline{x}_i^*$?
- LOCAL quadratic approximation

$$f(\underline{x}_0^* + \underline{\delta}) \approx f(\underline{x}_0^*) + \underline{\nabla} f(\underline{x}_0^*) \cdot \underline{\delta} + \frac{1}{2} \underline{\delta} \, \nabla^2 f(\underline{x}_0^*) \, \underline{\delta}$$

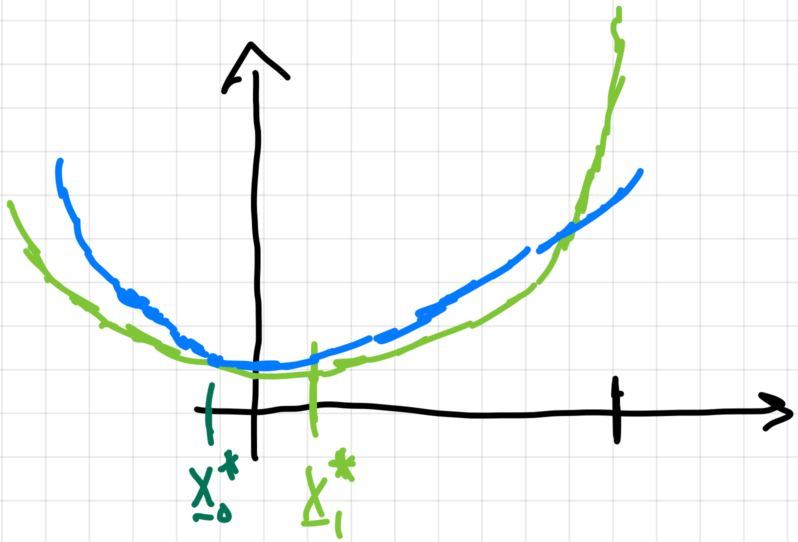


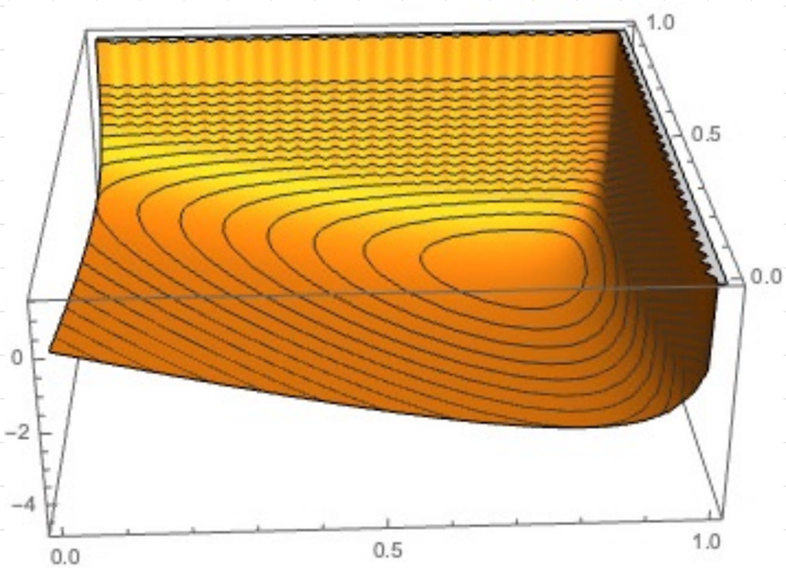


WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- Standard approach # 2: HOMOTOPY
- How do we find each new point $\underline{x}_i^*$?
- LOCAL quadratic approximation

$$f(\underline{x}_0^* + \underline{\delta}) \approx f(\underline{x}_0^*) + \underline{\nabla} f(\underline{x}_0^*) \cdot \underline{\delta} + \frac{1}{2}\, \underline{\delta}\, \nabla^2 f(\underline{x}_0^*)\, \underline{\delta}$$





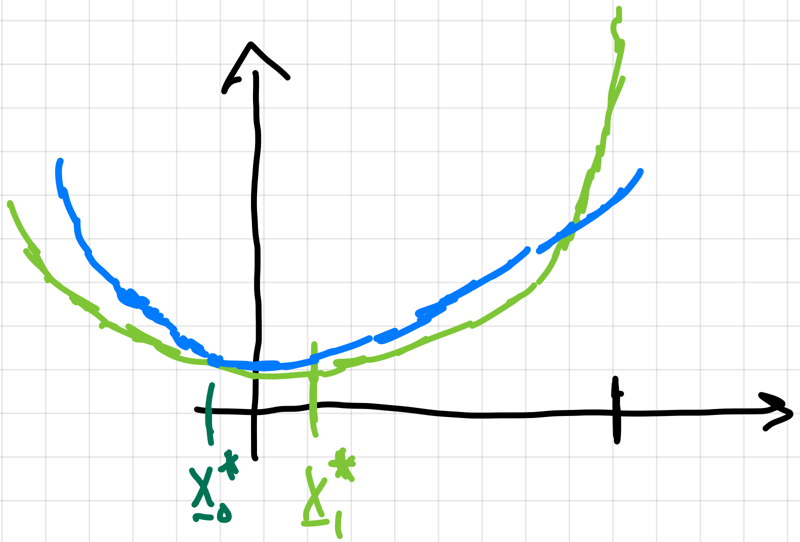
WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

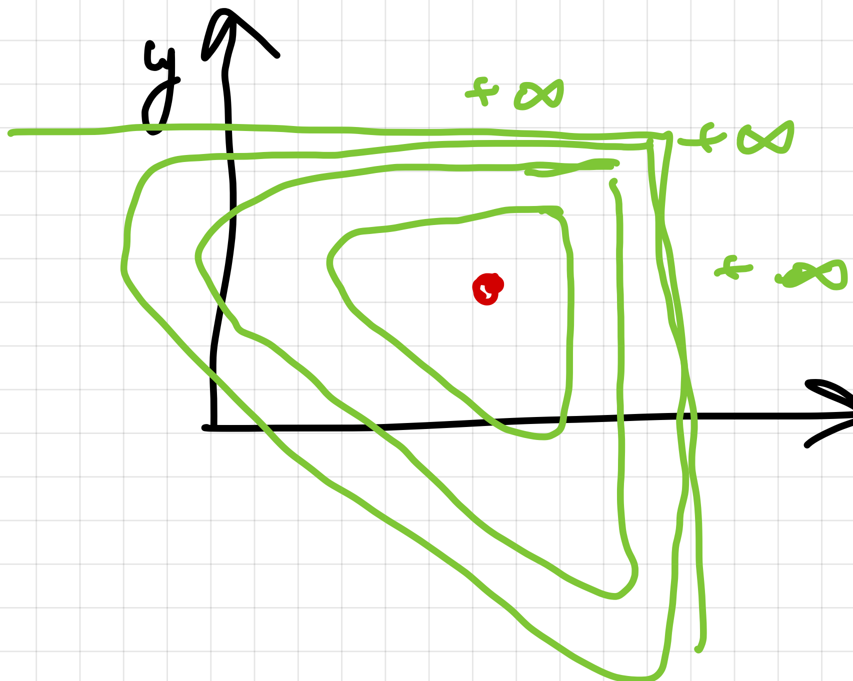
- Standard approach # 2: HOMOTOPY

- How do we find each new point $\underline{x}_i^*$?

- LOCAL quadratic approximation

$$f(\underline{x}_0^* + \underline{\delta}) \approx f(\underline{x}_0^*) + \underline{\nabla} f(\underline{x}_0^*) \cdot \underline{\delta} + \frac{1}{2} \underline{\delta}\, \nabla^2 f(\underline{x}_0^*)\, \underline{\delta}$$





$x+y$

s.t.

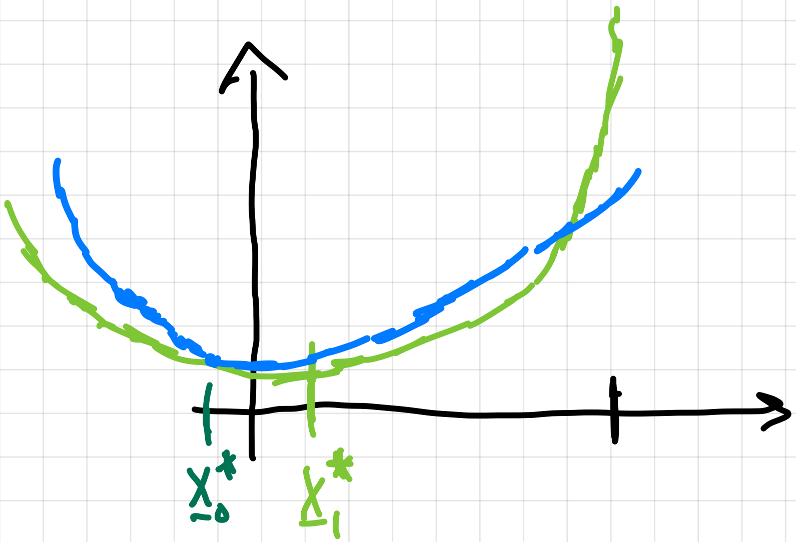
WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

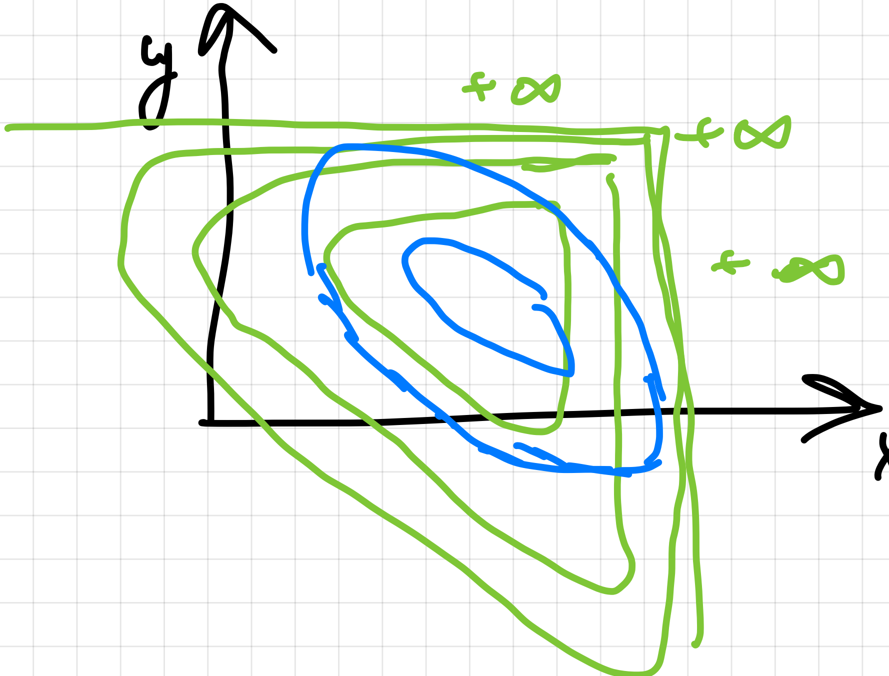
- Standard approach # 2: <u>HOMOTOPY</u>

- How do we find each new point $\underline{x}_i^*$?

- LOCAL quadratic approximation

$$f(\underline{x}_0^* + \underline{\delta}) \approx f(\underline{x}_0^*) + \underline{\nabla} f(\underline{x}_0^*) \cdot \underline{\delta} + \frac{1}{2}\, \underline{\delta}\, \nabla^2 f(\underline{x}_0^*)\, \underline{\delta}$$

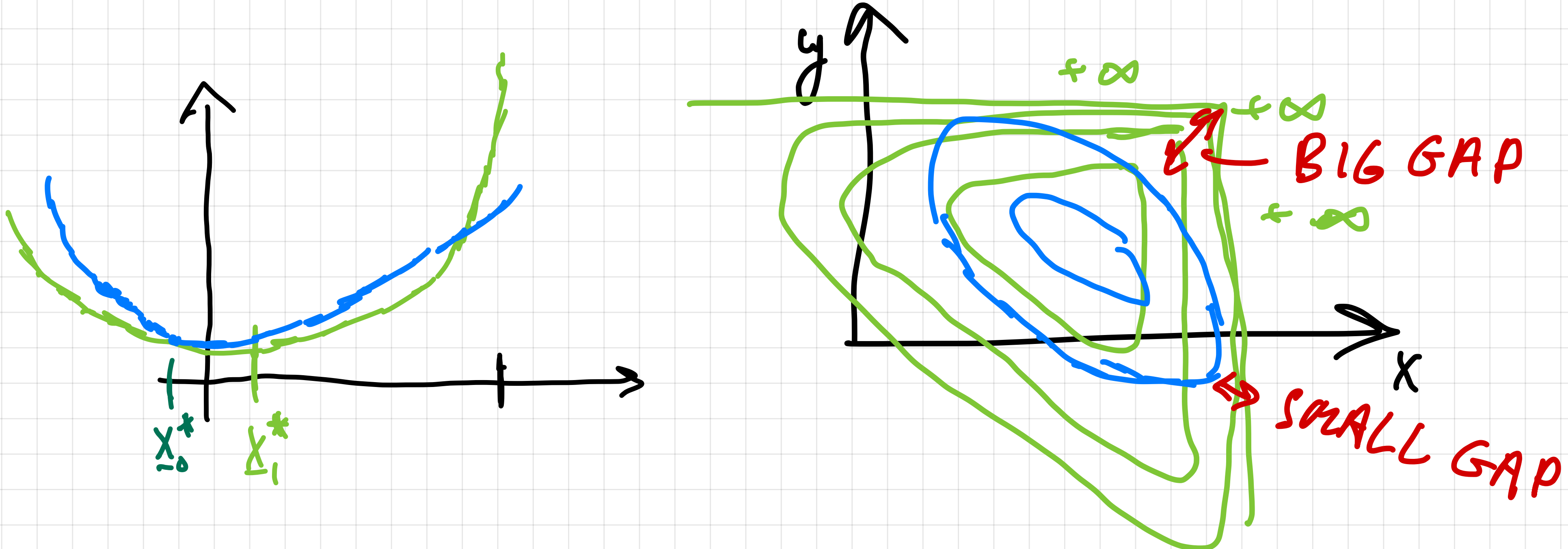




WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- Standard approach # 2: HOMOTOPY
- How do we find each new point $\underline{x}_i^*$?
- LOCAL quadratic approximation

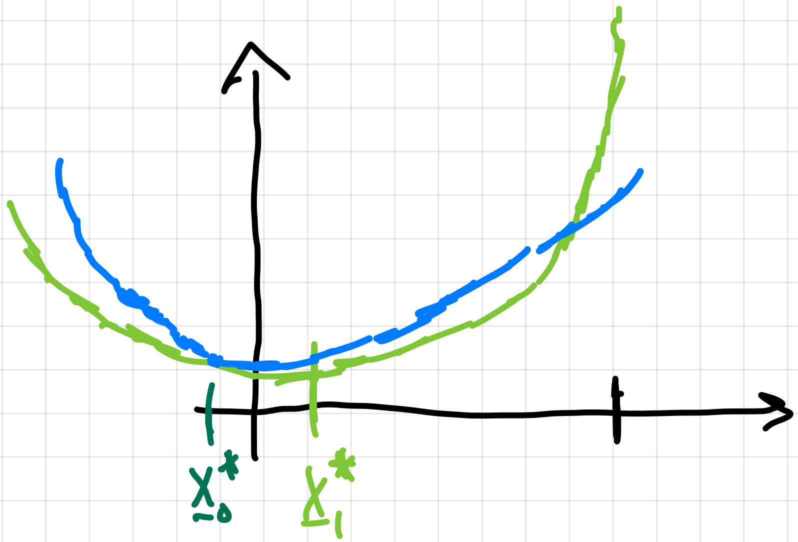
$$f(\underline{x}_0^* + \underline{\delta}) \approx f(\underline{x}_0^*) + \underline{\nabla} f(\underline{x}_0^*) \cdot \underline{\delta} + \frac{1}{2} \underline{\delta}\, \nabla^2 f(\underline{x}_0^*)\, \underline{\delta}$$

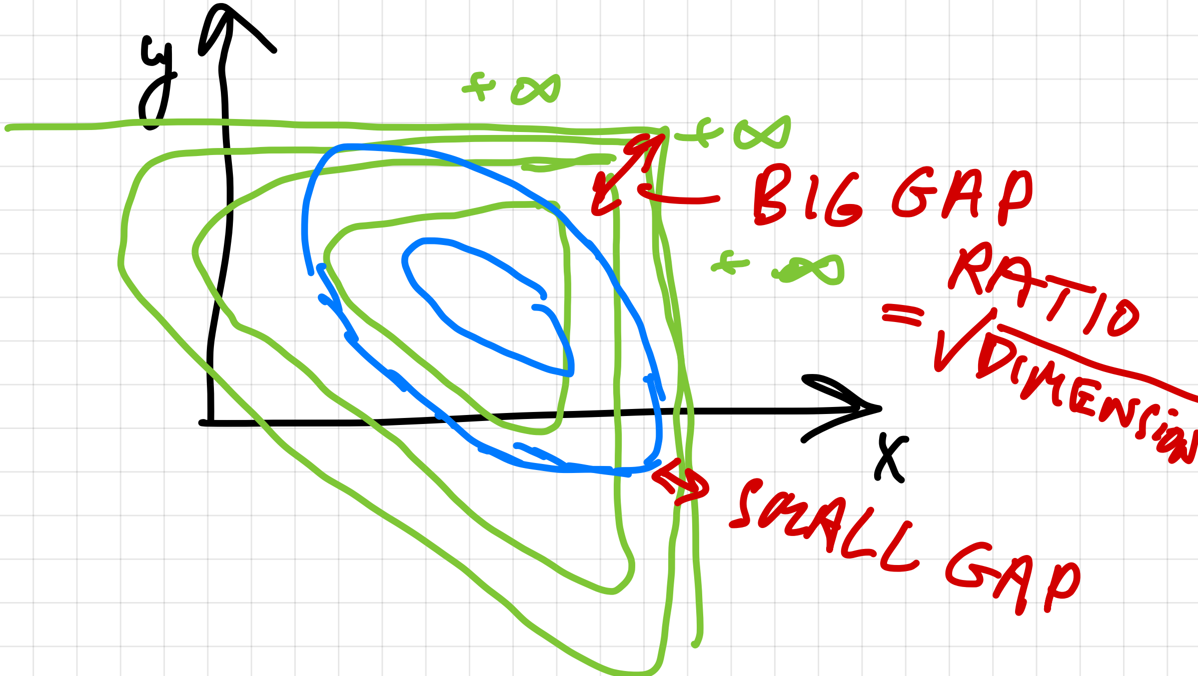


WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- Standard approach # 2: HOMOTOPY
- How do we find each new point $\underline{x}_i^*$?
- LOCAL quadratic approximation

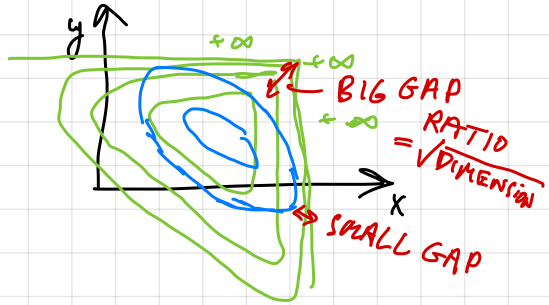
$$f(\underline{x}_0^* + \underline{\delta}) \approx f(\underline{x}_0^*) + \underline{\nabla} f(\underline{x}_0^*) \cdot \underline{\delta} + \frac{1}{2} \underline{\delta} \, \nabla^2 f(\underline{x}_0^*) \, \underline{\delta}$$





WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- Standard approach # 2: HOMOTOPY



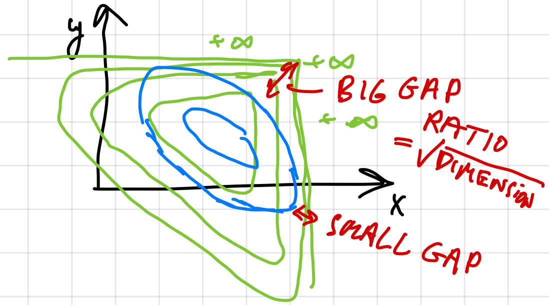
  - LOCAL quadratic approximation

$$f(\underline{x}_0^* + \underline{\delta}) \approx f(\underline{x}_0^*) + \underline{\nabla} f(\underline{x}_0^*) \cdot \underline{\delta} + \frac{1}{2} \underline{\delta}\, \nabla^2 f(\underline{x}_0^*)\, \underline{\delta}$$

  - Find $\underline{\delta}$? Solve a linear equation
  - $\underline{x}_1^* \approx \underline{x}_0^* + \underline{\delta}$

WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

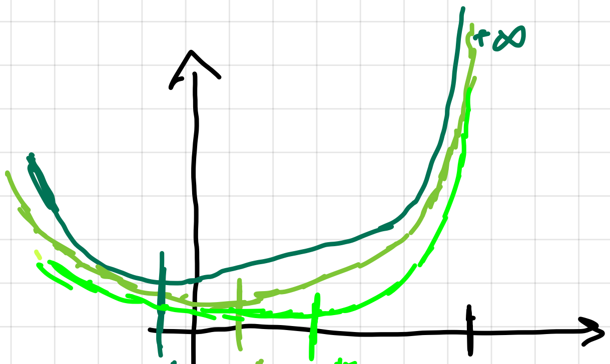
- Standard approach # 2: HOMOTOPY



  - LOCAL quadratic approximation

$$f(\underline{x}_0^* + \underline{\delta}) \approx f(\underline{x}_0^*) + \underline{\nabla} f(\underline{x}_0^*) \cdot \underline{\delta} + \frac{1}{2} \underline{\delta}\, \nabla^2 f(\underline{x}_0^*)\, \underline{\delta}$$

  - Find $\underline{\delta}$? Solve a linear equation
  - $\underline{x}_1^* \approx \underline{x}_0^* + \underline{\delta}$



$\underline{x}_0^*$ $\underline{x}_1^*$ $\underline{x}_2^*$ $\cdots$ $\underline{x}_t^* \to \underline{x}^*$ (true optimum)

- Need $t \approx \sqrt{\text{DIMENSION}}$ to get $\underline{x}_t^* \approx \underline{x}^*$

WHAT MAKES NON-SMOOTH OPTIMIZATION HARD?

- Standard approach # 2: <u>HOMOTOPY</u>

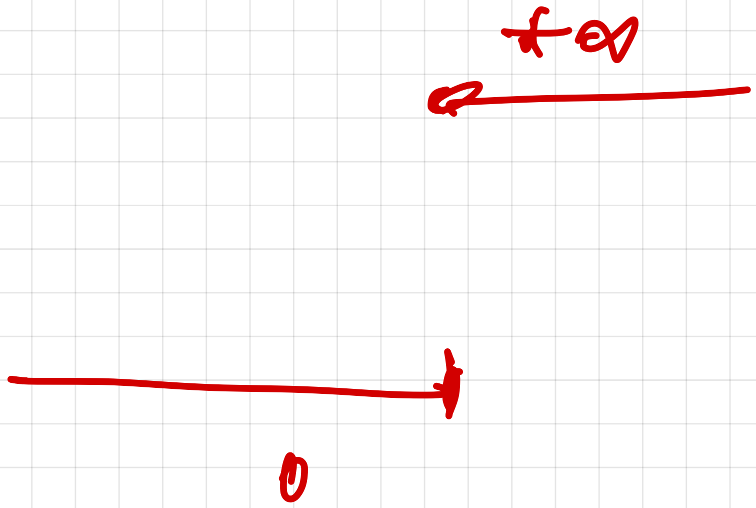
- Need $\approx \sqrt{\text{DIMENSION}}$ linear eq. solves to get $\underline{x}^*$

→ Renegar '86: Linear programs

→ Nesterov & Nemirovski '94

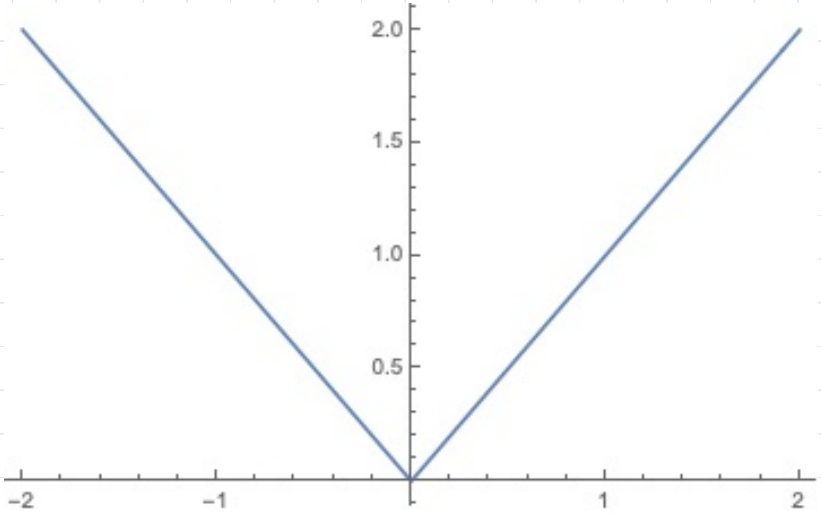
Almost anything convex

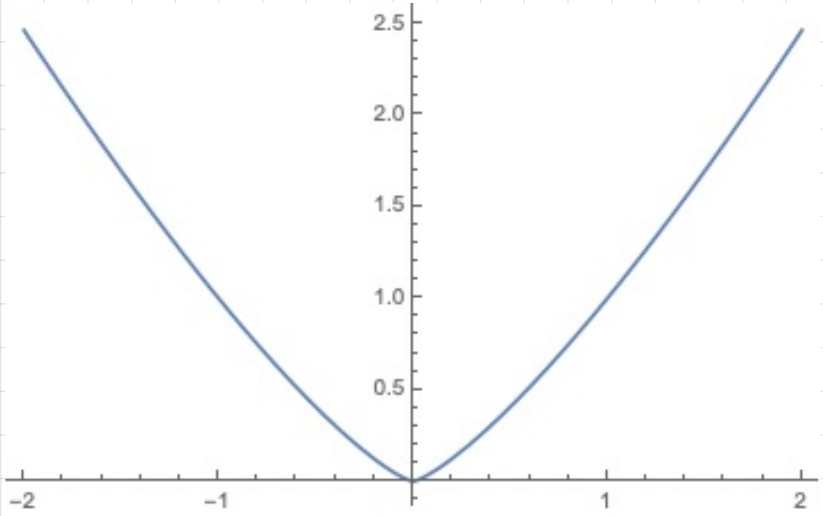
"Compile to hard constraints"



WHAT ABOUT LESS SMOOTH OPTIMIZATION?

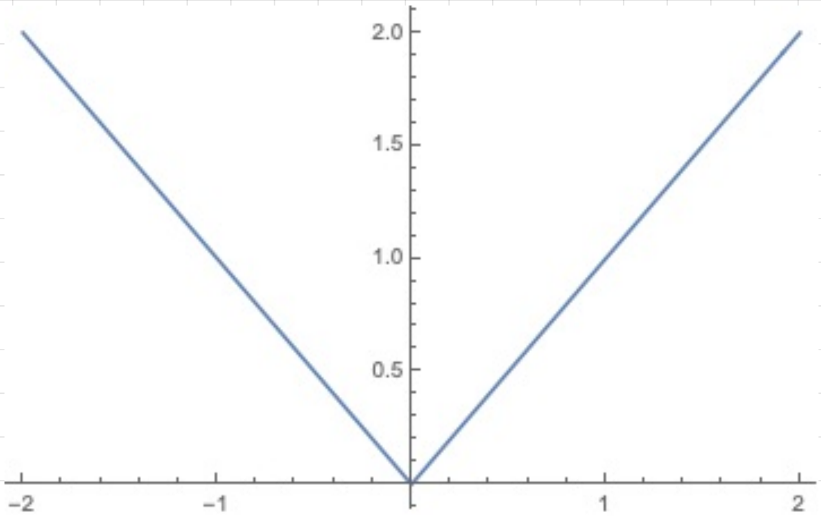
$f(x) = |x|$ vs $f(x) = |x|^{1.1}$

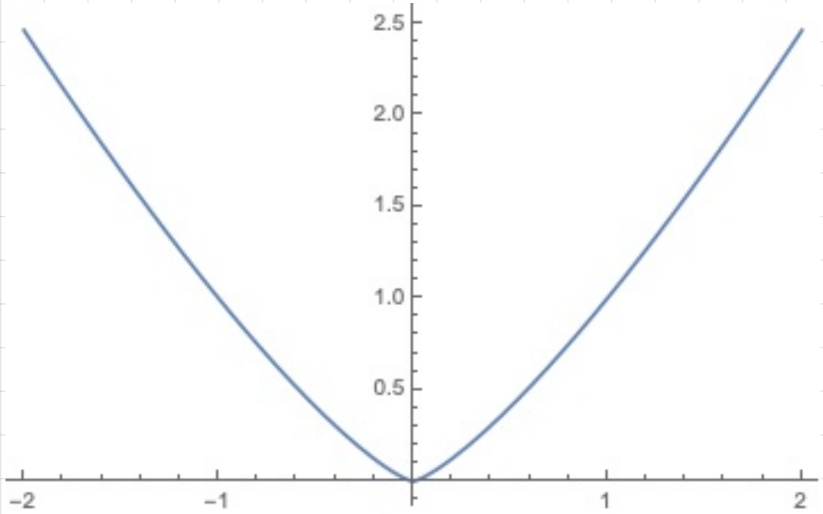




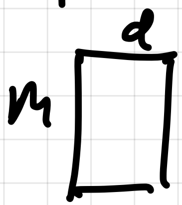
WHAT ABOUT LESS SMOOTH OPTIMIZATION?

$f(x) = |x|$ vs $f(x) = |x|^{1.1}$





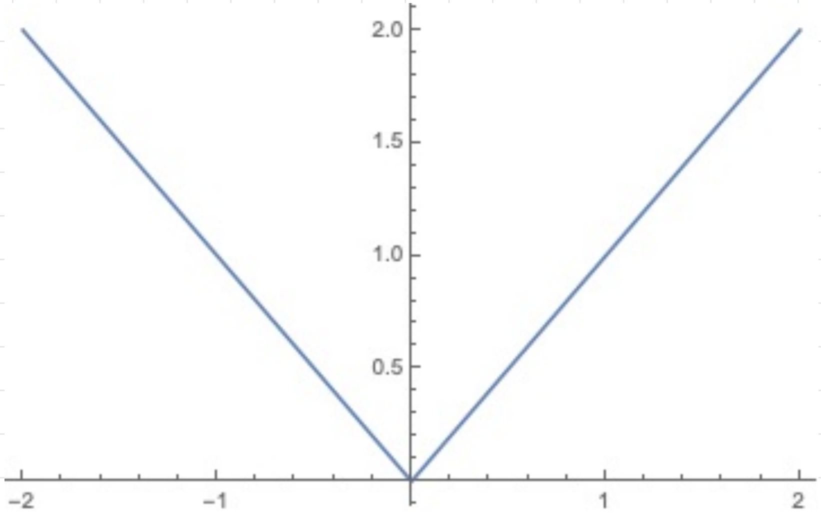
Nesterov & Nemirovski '94: $f(\underline{x}) = \|A\underline{x} - \underline{b}\|_p^p$

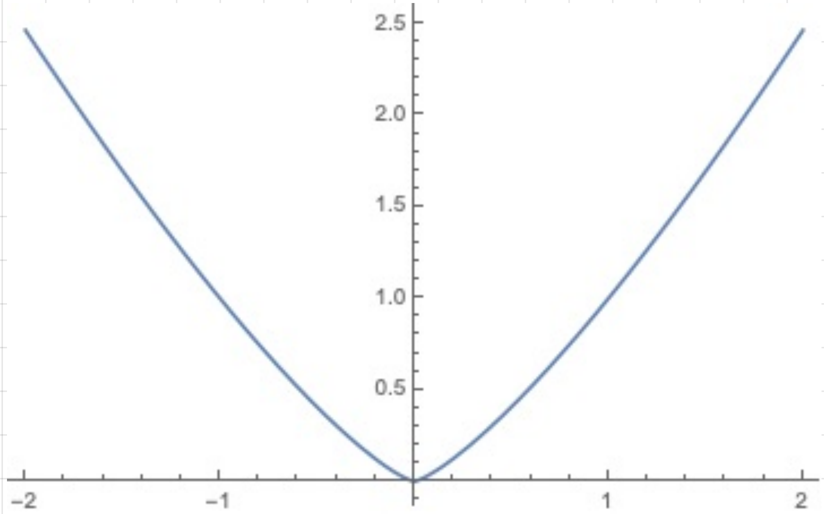


$1 < p < \infty$

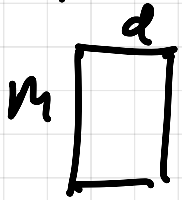
WHAT ABOUT LESS SMOOTH OPTIMIZATION?

$f(x) = |x|$ vs $f(x) = |x|^{1.1}$





Nesterov & Nemirovski '94: $f(\underline{x}) = \|A\underline{x} - \underline{b}\|_p^p$



in $\approx \sqrt{m}$ linear equations using homotopy.

ITERATIVE REFINEMENT

- If we can solve $A\underline{x} = \underline{b}$ approximately, <u>then</u> we can solve it to high accuracy:

$$\underline{b}_1 = \underline{b} - Ax_0 \,, \quad \text{solve } A\underline{x}' = \underline{b}_1$$

$$\underline{b}_2 = \underline{b} - A(\underline{x}_0 + \underline{x}_1)$$

$$\vdots$$

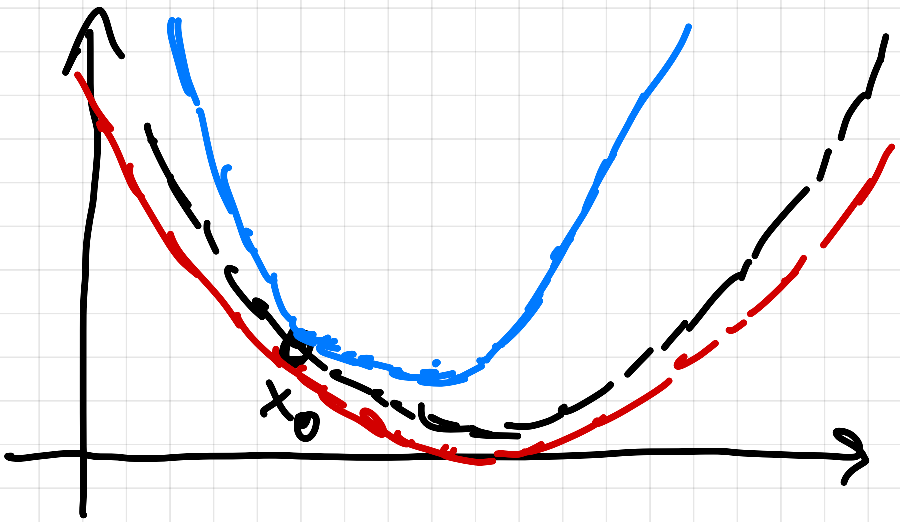
- Converges to ε error exteremely quickly:

$$\log\left(\frac{1}{\varepsilon}\right) \text{ iterations}$$

ITERATIVE REFINEMENT

- View as a quadratic function

$$f(\underline{x}) = \|A\underline{x} - \underline{b}\|_2^2 = f(\underline{x}_0) + \nabla f(\underline{x}_0)^T(\underline{x} - \underline{x}_0) + \frac{1}{2}(\underline{x} - \underline{x}_0)\nabla^2 f(\underline{x}_0)(\underline{x} - \underline{x}_0)$$



DESPITE ERROR, compare w. upper bound and lower bound

ITERATIVE REFINEMENT

$$f(\underline{x}) = \|A\underline{x} - \underline{b}\|_2^2 = f(\underline{x}_0) + \nabla f(\underline{x}_0)^T(\underline{x} - \underline{x}_0) + \frac{1}{2}(\underline{x} - \underline{x}_0)\nabla^2 f(\underline{x}_0)(\underline{x} - \underline{x}_0)$$

$$f(\underline{x}) = \|A\underline{x} - b\|_p^p = f(\underline{x}_0) + \nabla f(\underline{x}_0)^T(\underline{x} - \underline{x}_0) + \frac{1}{2}(\underline{x} - \underline{x}_0)\nabla^2 f(\underline{x}_0)(\underline{x} - \underline{x}_0)$$

$$+ \cdots + \frac{1}{p!}\nabla^p f(\underline{x}_0)[\underline{x} - \underline{x}_0; \cdots]$$

Simplify:

$$\min_{\underline{x}} \|A\underline{x} - \underline{b}\|_p^p \longrightarrow \min_{C\underline{y} = \underline{d}} \|\underline{y}\|_p^p$$

GOAL $\min_{C\underline{y}=\underline{d}} \|\underline{y}\|_p^p$

$$\|\underline{y}\|_p^p = \sum_i |y(i)|^p$$

Update $\underline{y}$? $\underline{y}_0 \rightarrow \underline{y}_0 + \underline{\delta}$

$$C\underline{y}_0 = \underline{d}, \quad C(\underline{y}_0 + \underline{\delta}) = \underline{d} \qquad C\underline{\delta} = \underline{0}$$

Case $p = 2$

$$\min_{C\underline{y}=\underline{d}} \|\underline{y}\|_2^2 \longrightarrow \min_{C\underline{\delta}=\underline{0}} \|\underline{y}_0 + \underline{\delta}\|_2^2$$

$$\|\underline{y}_0 + \underline{\delta}\|_2^2 = \|\underline{y}_0\|_2^2 + \underbrace{2\,\underline{y}_0 \cdot \underline{\delta} + \|\underline{\delta}\|_2^2}_{\substack{\text{if we can do apx min} \\ \text{w.r.t. } \underline{\delta} \text{ w. } (C\underline{\delta}=\underline{0}) \\ \text{then we recover} \\ \text{constant fraction of} \\ \text{PROGRESS}}}$$

SINGLE COORD

$$(y_0+\delta)^2 = y_0^2 + \underbrace{2y_0\delta + \delta^2}_{\text{const apx}}$$

$$(y_0+\delta)^p = y_0^p + \underbrace{p\,y_0^{p-1}\delta + \binom{p}{2} y_0^{p-2}\delta^2 + \dots + \delta^p}_{\text{const apx ?}}$$

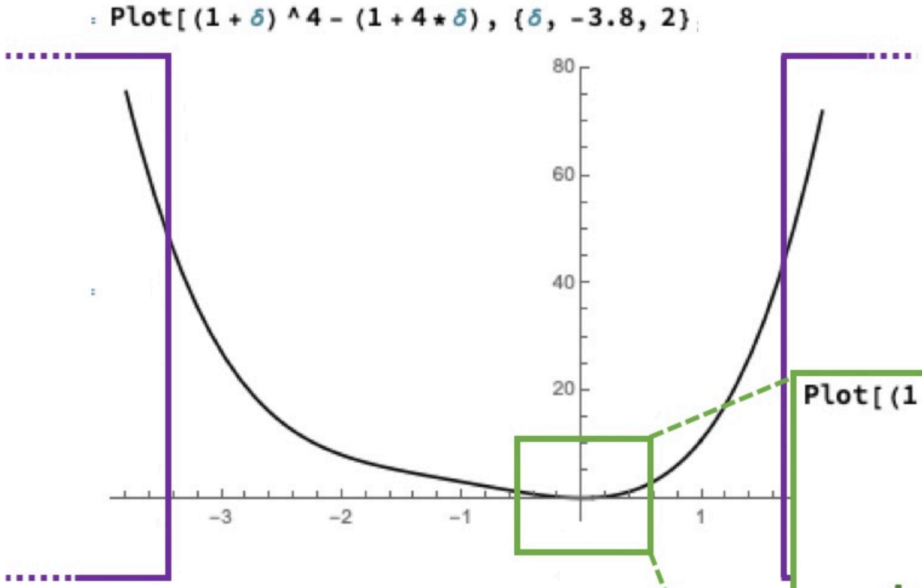
ITERATIVE REFINEMENT

$y_0 = 1$ then

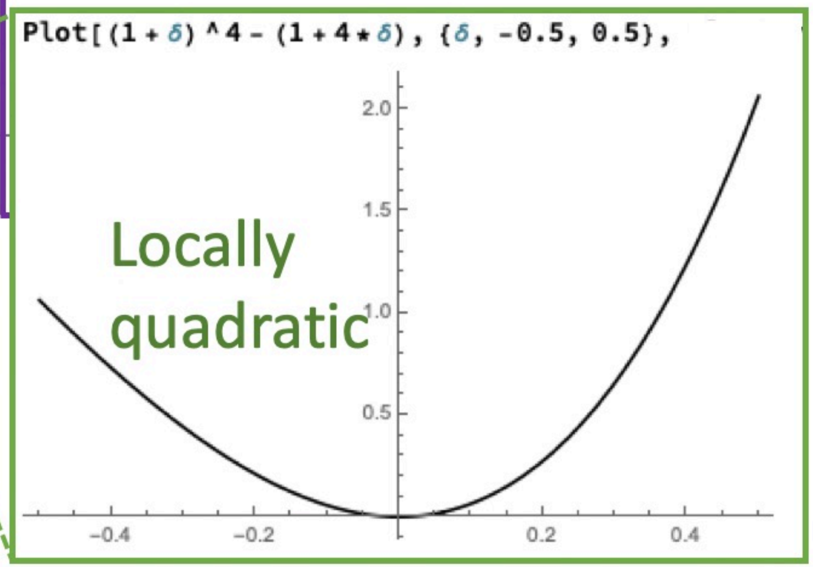
$$(1+\delta)^p = \underbrace{1 + p\delta}_{\text{linear approximation}} + \underbrace{\binom{p}{2}\delta^2}_{\text{local curvature}} + \binom{p}{3}\delta^3 + \dots + \underbrace{\delta^p}_{\text{long range behavior}}$$

ITERATIVE REFINEMENT

$$(1+\delta)^p = \underbrace{1+p\delta}_{\text{linear approximation}} + \underbrace{\binom{p}{2}\delta^2}_{\text{local curvature}} + \binom{p}{3}\delta^3 + \dots + \underbrace{\delta^p}_{\text{long range behavior}}$$



Long range:
4th power

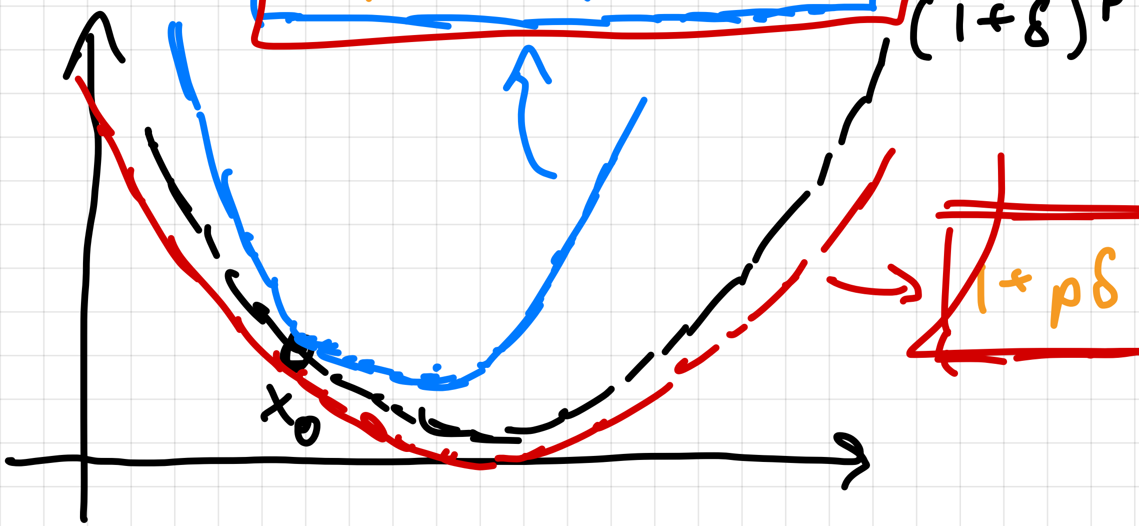


$$(1+\delta)^4 - (1+4\delta) \approx \delta^2 + \delta^4$$

ITERATIVE REFINEMENT

$$(1+\delta)^p = \underbrace{1 + p\delta}_{\text{linear approximation}} + \underbrace{\binom{p}{2}\delta^2}_{\text{local curvature}} + \binom{p}{3}\delta^3 + \dots + \underbrace{\delta^p}_{\text{long range behavior}}$$

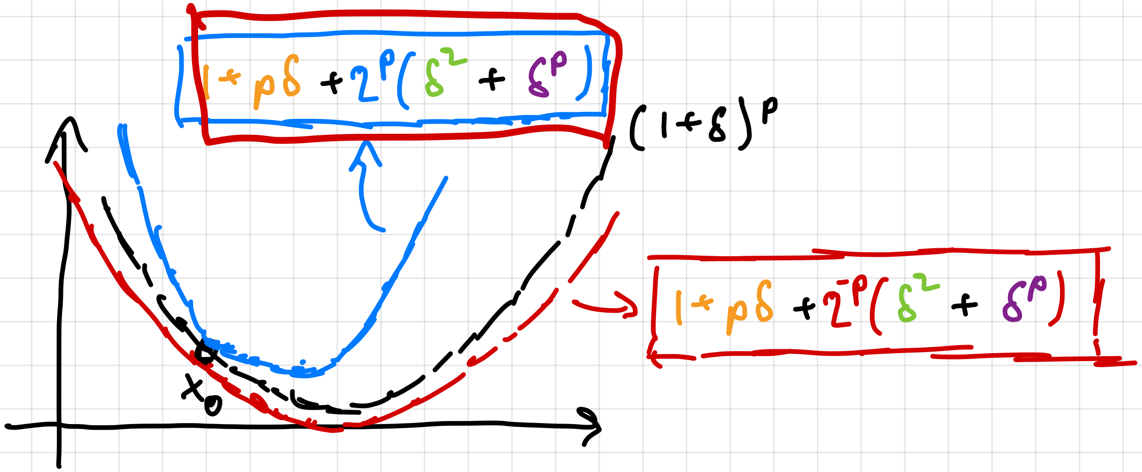
$$\boxed{1 + p\delta + 2^p(\delta^2 + \delta^p)}$$



$$\boxed{1 + p\delta + 2^{-p}(\delta^2 + \delta^p)}$$

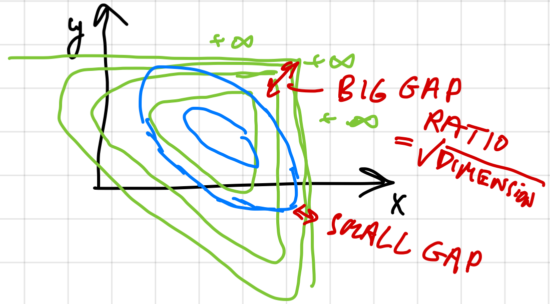
GLOBAL APPROXIMATION!

ITERATIVE REFINEMENT



GLOBAL APPROXIMATION!

HOMOTOPY



- LOCAL quadratic approximation

ITERATIVE REFINEMENT

$$(1+\delta)^p \approx \boxed{1+p\delta+2^p(\delta^2+\delta^p)}$$ GLOBAL APPROXIMATION

What does this buy us?

→ If we can do APPROXIMATE MIN of

LINEAR + QUADRATIC + POWER p

then $\boxed{\text{we can solve } \|Ax-b\|_p^p \text{ to high accuracy! in } O\left(2^p \log\left(\frac{1}{\varepsilon}\right)\right) \text{ iterations!}}$

[Adil Kyng Sachdeva Peng SODA 2019]

ITERATIVE REFINEMENT

NEED APPROXIMATE MIN OF

LINEAR + QUADRATIC + POWER p

→ NEW MULTIPLICATIVE WEIGHT METHOD

– WIDTH REDUCTION!

≈ ACCELERATION

→ Solve in $m^{c(p)}$ linear equation solves

$$\|Ax - b\|_p^p \qquad m\,\overset{d}{\boxed{A}}$$

→ MWU width reduction based on

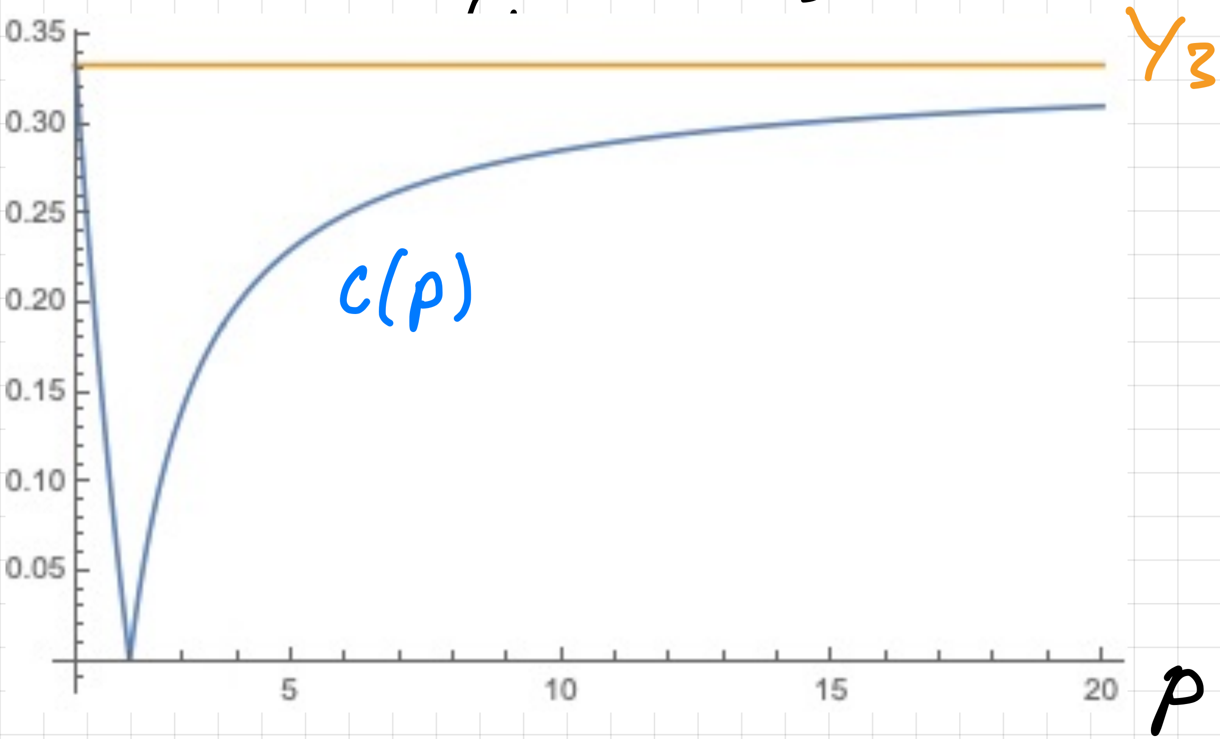
Christiano-Kelner-Madry-Spielman-Teng '11

ITERATIVE REFINEMENT

NEED APPROXIMATE MIN of

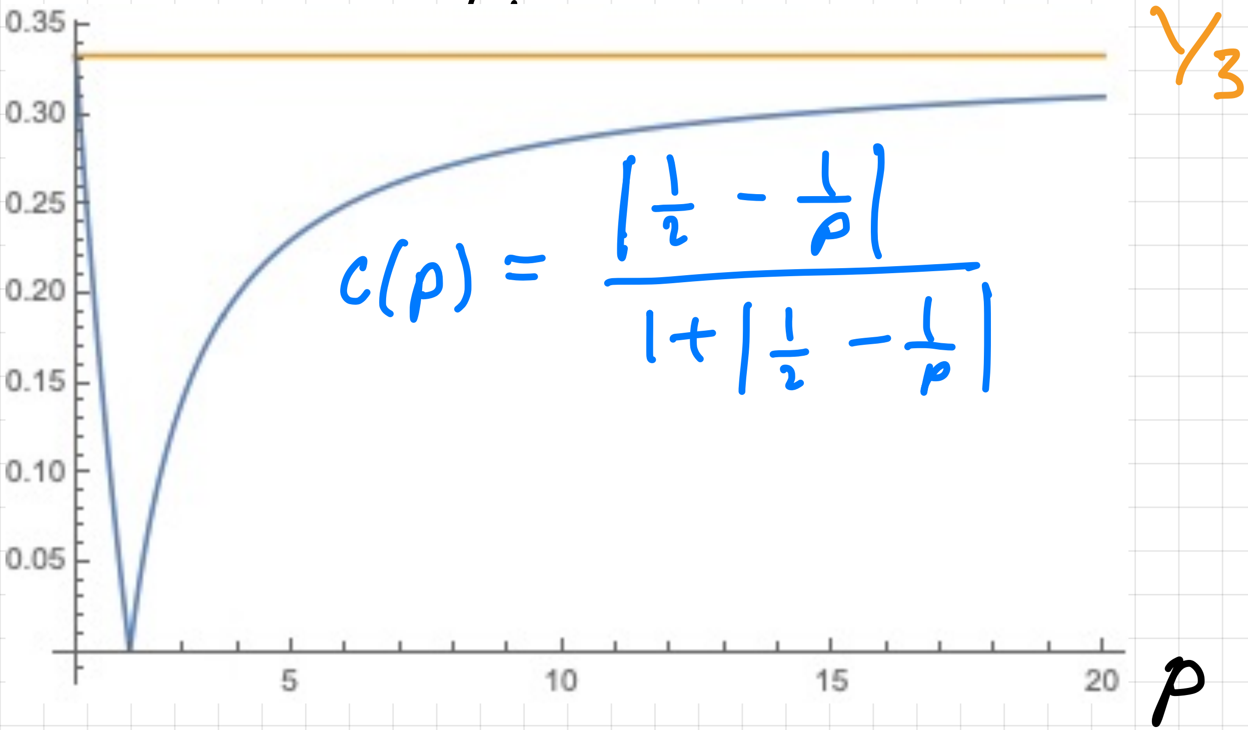
LINEAR + QUADRATIC + POWER p

→ Solve in $n^{c(p)}$ linear equation solves



ACCELERATION

NEED APPROXIMATE MIN of
LINEAR + QUADRATIC + POWER p
→ Solve in $n^{c(p)}$ linear equation solves



$$c(p) = \frac{\left|\frac{1}{2} - \frac{1}{p}\right|}{1 + \left|\frac{1}{2} - \frac{1}{p}\right|}$$

$\sqrt[3]{\text{DIMENSION}}$ instead of $\sqrt{\text{DIMENSION}}$?!

AKPS '19 again

ACCELERATION

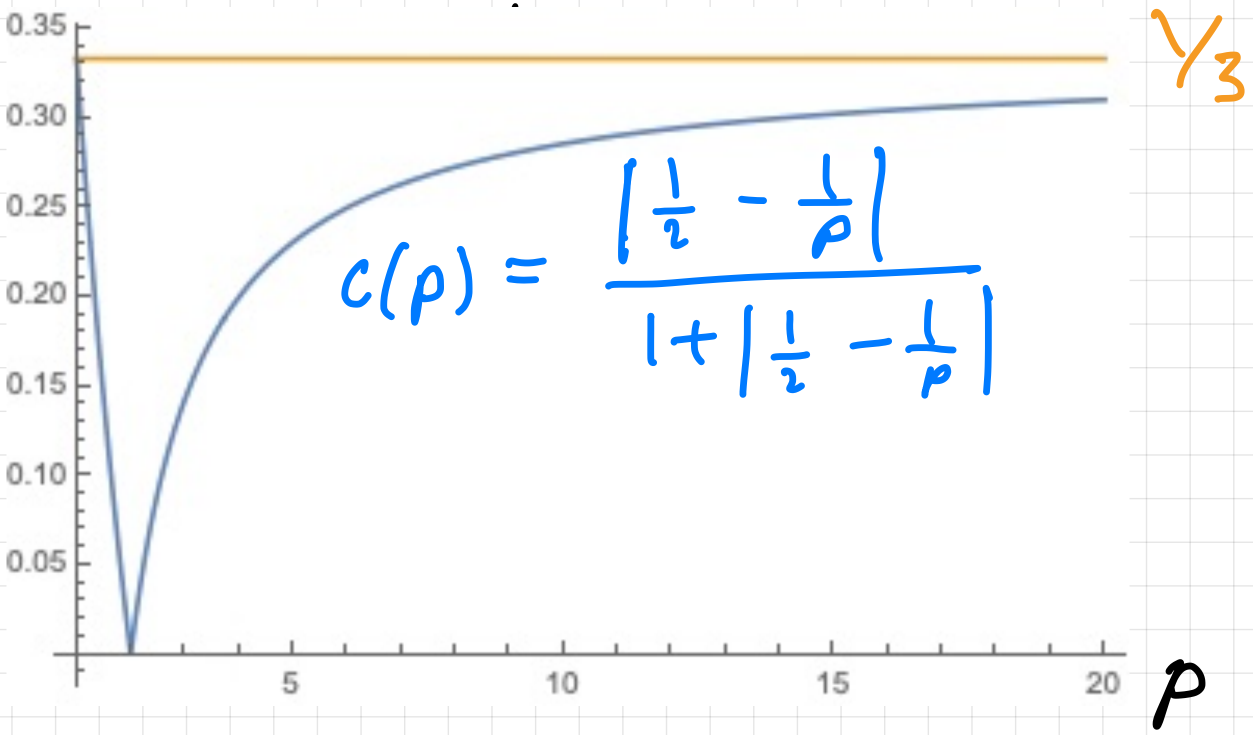
$\sqrt[3]{\text{DIMENSION}}$ instead of $\sqrt{\text{DIMENSION}}$?!

AKPS '19 again

Important prior work: Christiano et al. STOC '11

Bubeck et al. STOC '18

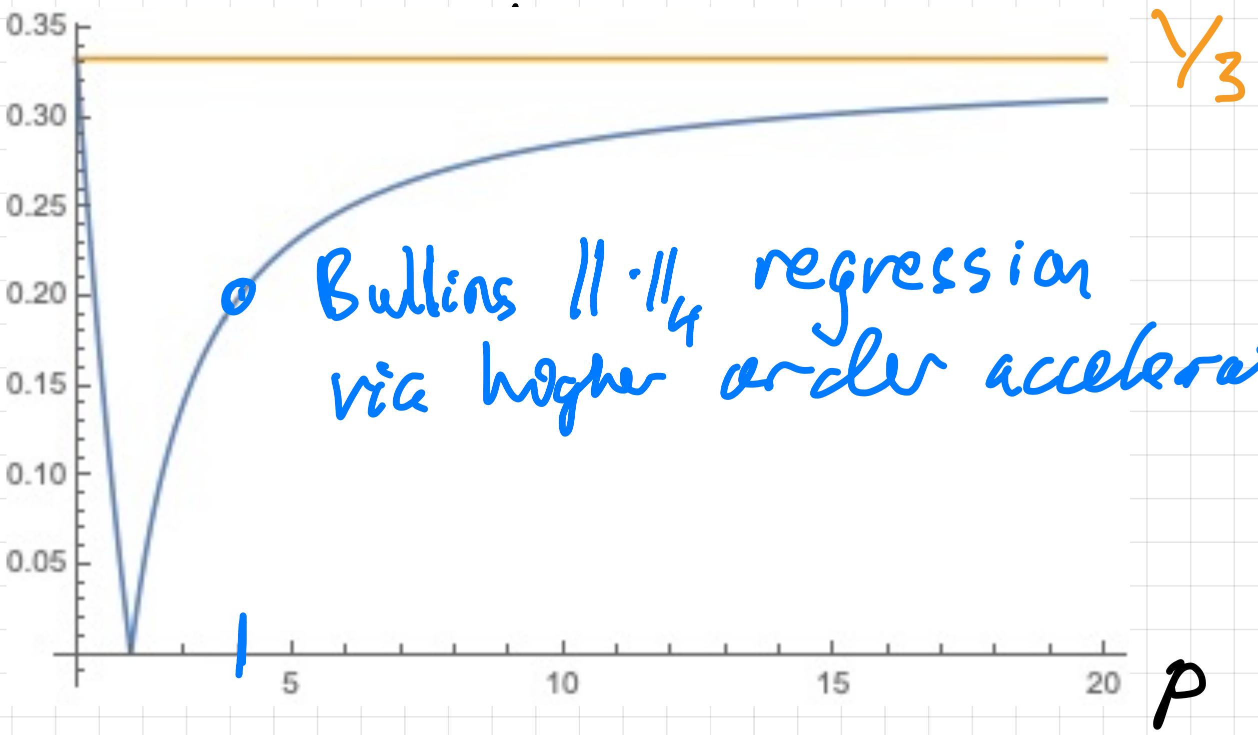
ACCELERATION



$$c(p) = \frac{\left|\frac{1}{2} - \frac{1}{p}\right|}{1 + \left|\frac{1}{2} - \frac{1}{p}\right|}$$

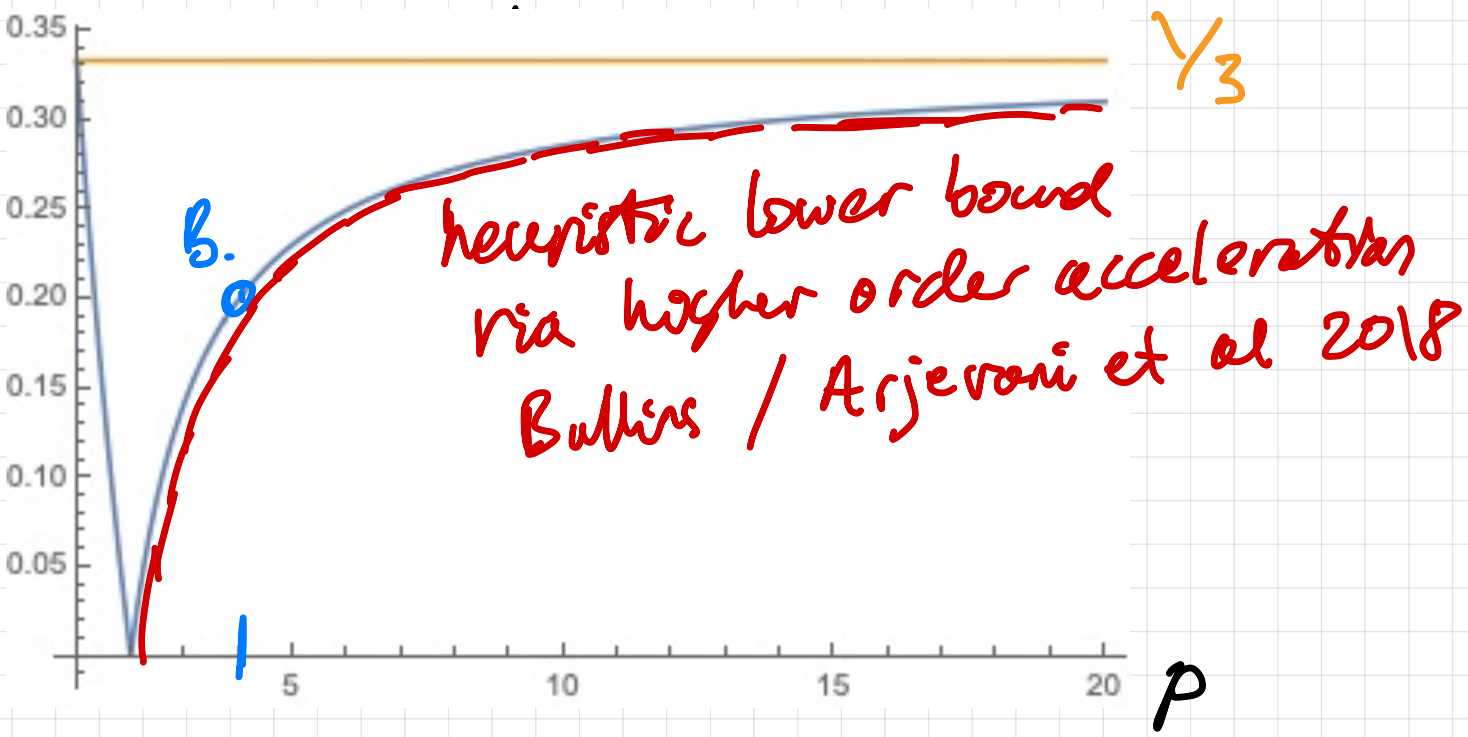
What is going on here?

ACCELERATION

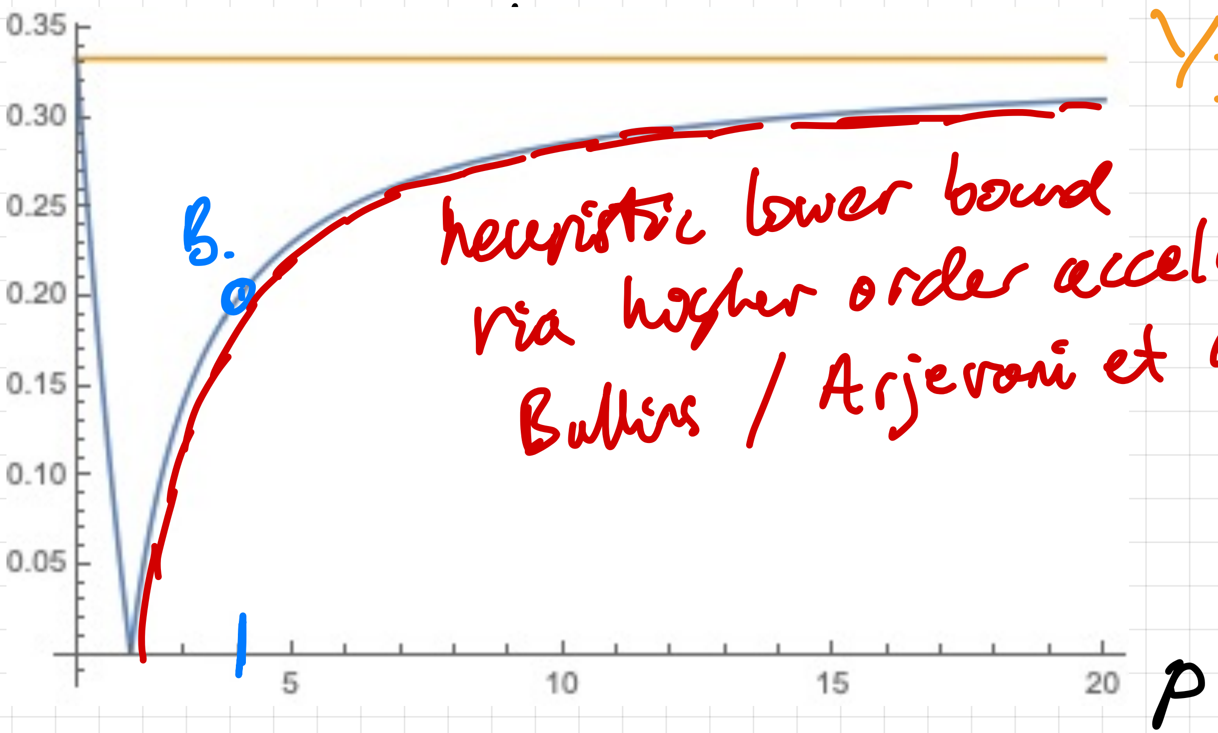


- Bullins $\|\cdot\|_4$ regression via higher order acceleration

ACCELERATION



ACCELERATION



heuristic lower bound via higher order acceleration Bullins / Arjevani et al 2018

• 1/2 Renegar '86

1/3 heuristic lower bound for linear programming

Spielman Madry

$p = \infty$

ACCELERATION

The Christiano et al. scheme

$$\min \|\underline{f}\|_\infty$$
$$B\underline{f} = \underline{d}$$

- $\|f\|_2$ min for $\|f\|_\infty$ min?
- Use multiplicative weight method
- Each iteration:

1) $\sum_e p_e |f_e| \leq (1+\varepsilon) \|f^*\|_\infty$

2) $\|f\|_\infty \leq \rho$

- Repeat ρ_{ε^c} times for ε error

ACCELERATION

The Christiano et al. scheme

$$\min \|\underline{f}\|_\infty$$
$$B\underline{f} = \underline{d}$$

- $\|f\|_2$ min for $\|f\|_\infty$ min?
- Use multiplicative weight method
- Each iteration:

1) $\sum_e p_e |f_e| \leq (1+\varepsilon) \|f^*\|_\infty$

2) $\|f\|_\infty \leq \rho$

- Repeat ρ_{ε^c} times for ε error

$\rho = \sqrt{m}$

for

$f \in \operatorname{argmin} \underline{f}^T R \underline{f}$

$B\underline{f} = \underline{d}$

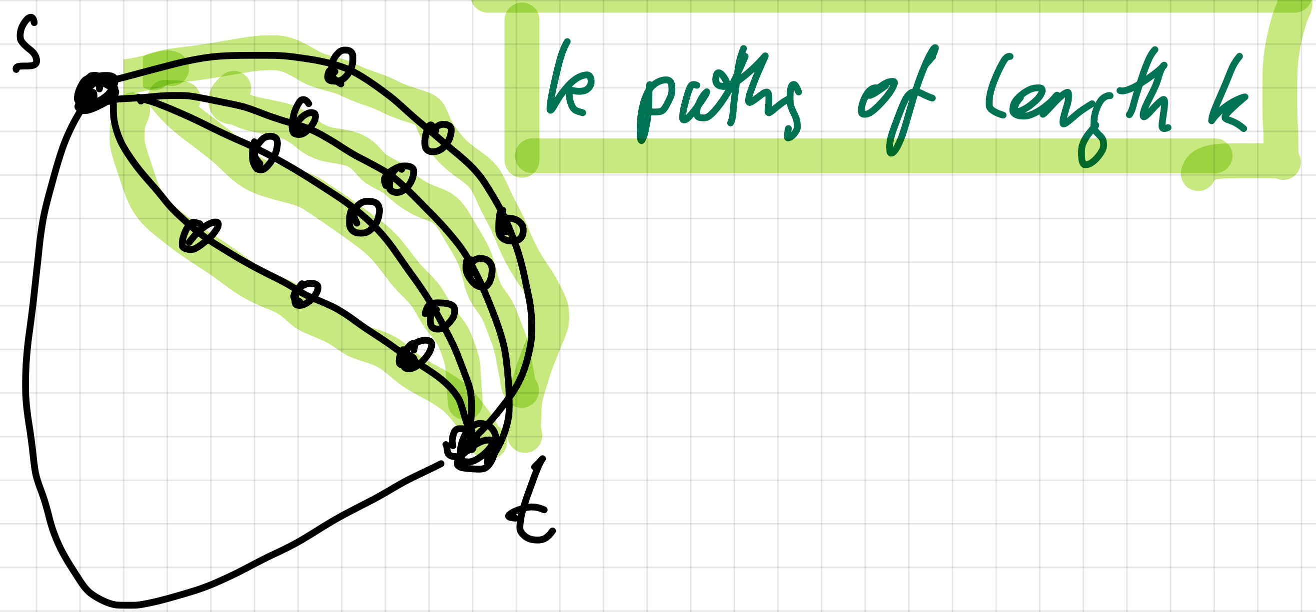
$R = \frac{1}{m} I + \operatorname{diag}(\underline{p})$

ACCELERATION

The Christiano et al. scheme

$$\min \|\underline{f}\|_\infty \quad B\underline{f} = \underline{d}$$

- $\|f\|_2$ min for $\|f\|_\infty$ min?
- Use multiplicative weight method



$\rho = \sqrt{m}$

for

$$f \in \operatorname{argmin}_{B\underline{f} = \underline{d}} \underline{f}^T R \underline{f}$$

$$R = \frac{1}{m} I + \operatorname{diag}(\underline{\rho})$$

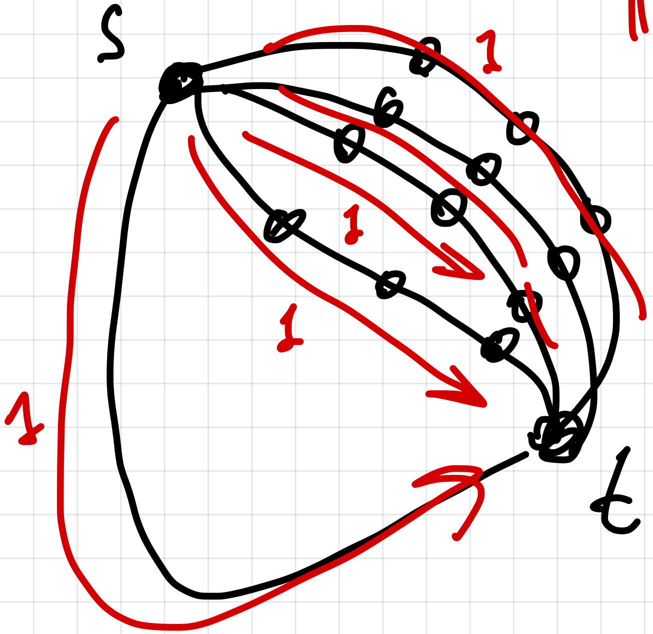
ACCELERATION

The Christiano et al. scheme

$$\min \|\underline{f}\|_\infty \quad B\underline{f} = \underline{d}$$

- $\|f\|_2$ min for $\|f\|_\infty$ min?
- Use multiplicative weight method

$\|f\|_\infty$ min: $\|f\|_\infty \le 1$



$k+1$ units

$\underline{\rho} = \sqrt{m}$

for

$\underline{f} \in \operatorname{argmin} \underline{f}^T R \underline{f}$

$B\underline{f} = \underline{d}$

$R = \frac{1}{m} I + \operatorname{diag}(\underline{\rho})$

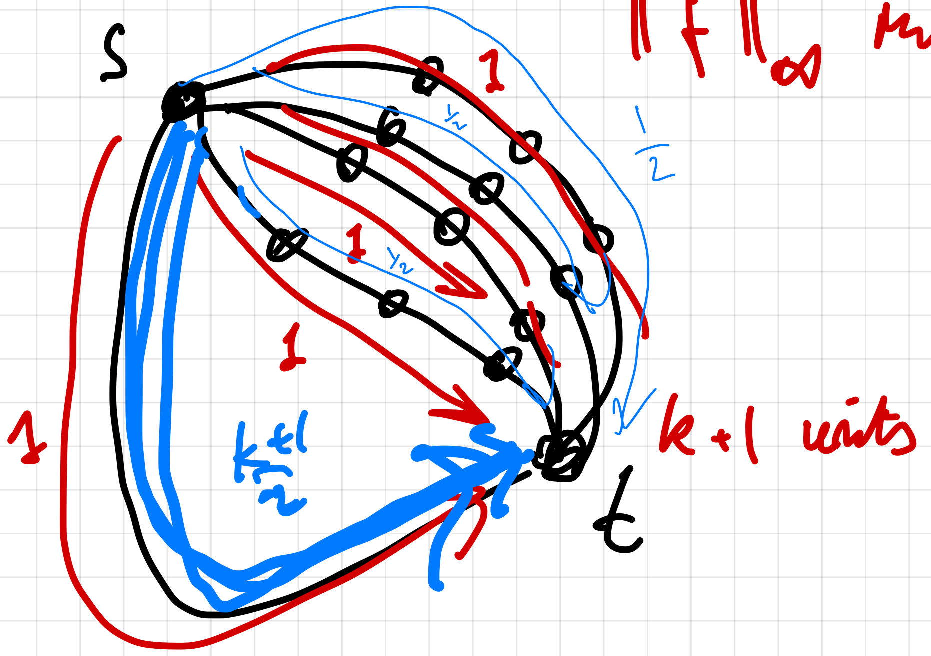
ACCELERATION

The Christiano et al. scheme

$$\min \|\underline{f}\|_\infty \quad B\underline{f} = \underline{d}$$

- $\|f\|_2$ min for $\|f\|_\infty$ min?
- Use multiplicative weight method

$\|f\|_\infty$ min : $\|f\|_\infty \leq 1$



$\rho = \sqrt{m}$

for

$\underline{f}_2 \leftarrow \arg\min \underline{f}^T R \underline{f}$

$B\underline{f} = \underline{d}$

$R = \frac{1}{n} I + \mathrm{diag}(\underline{\rho})$

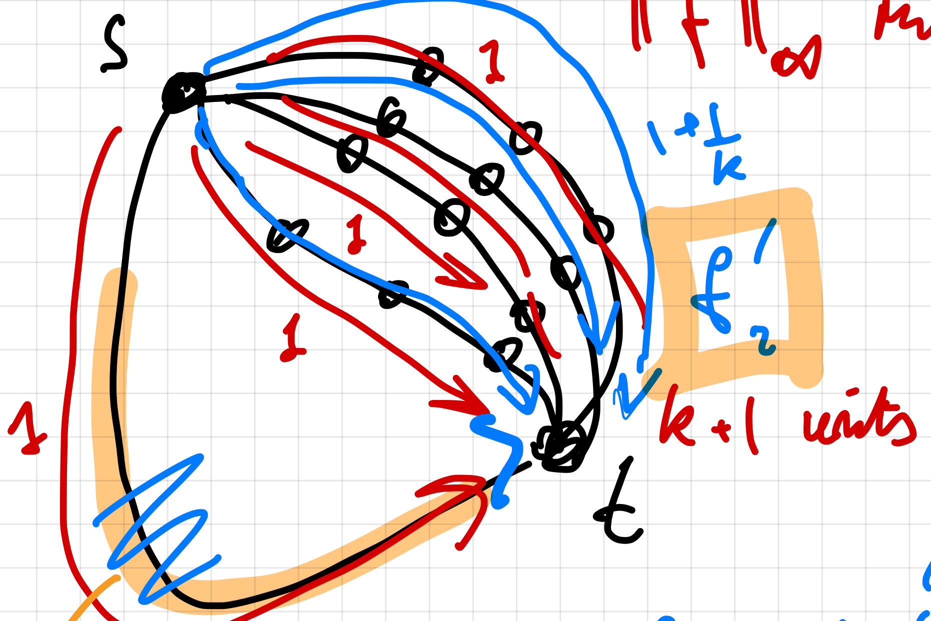
ACCELERATION

The Christiano et al. scheme

$$\min \|\underline{f}\|_\infty \quad B\underline{f} = \underline{d}$$

- $\|f\|_2$ min for $\|f\|_\infty$ min?
- Use multiplicative weight method

$\|f\|_\infty$ min: $\|f\|_\infty \le 1$



$$\|f_2'\|_\infty \le \frac{k+1}{k} \approx 1$$

delete edge from electrical flow

$\rho \approx \sqrt{m}$

for

$$f_2 \in \operatorname{argmin} \underline{f}^T R \underline{f} \quad B\underline{f} = \underline{d}$$

$$R = \frac{1}{m} I + \operatorname{diag}(\underline{\rho})$$

ACCELERATION

The Christiano et al. scheme

$$\min \|\underline{f}\|_\infty \quad B\underline{f} = \underline{d}$$

- $\|f\|_2$ min for $\|f\|_\infty$ min?
- Use multiplicative weight method

  $\|f\|_2$ with deletion of high flow edges:

  $$\|f\|_\infty \leq m^{1/3}$$

ACCELERATION

The Christiano et al. scheme

$$\min \|\underline{f}\|_\infty$$
$$B\underline{f} = \underline{d}$$

- $\|f\|_2$ min for $\|f\|_\infty$ min?
- Use multiplicative weight method

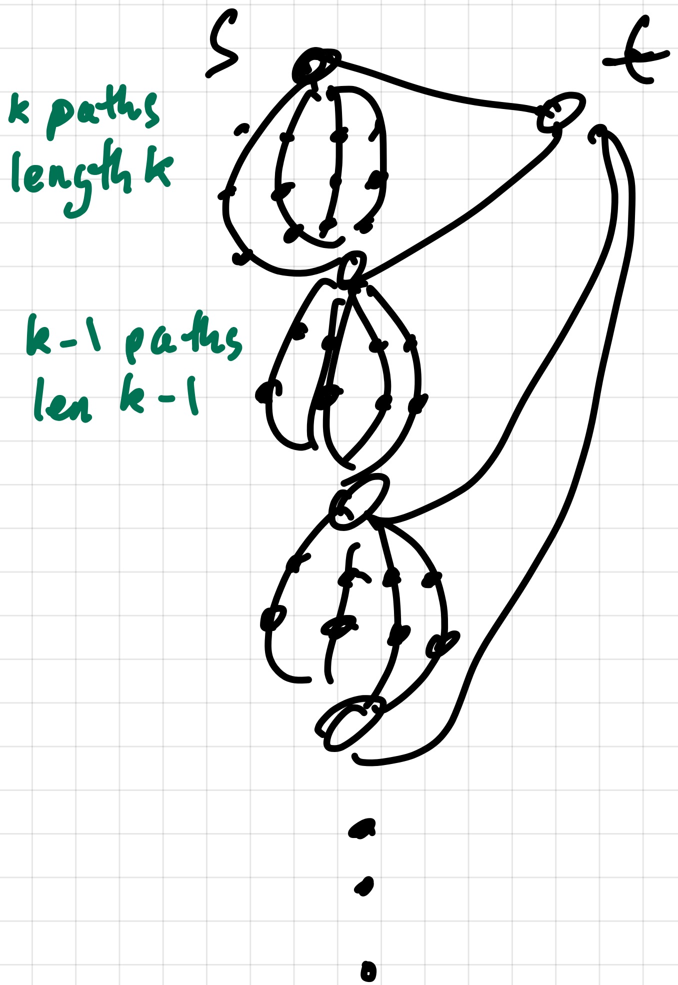
$\|f\|_2$ with deletion of high flow edges:

$$\|f\|_\infty \leq m^{1/3}$$

and this analysis is tight.

ACCELERATION

$m^{1/3}$ lower bound for flow + deletion



$\|f\|_\infty$ min ≤ 1

w. $k+1$ units s-t

f $\|\cdot\|_2$ min w. deletions

$\|f\|_\infty \geq \frac{k}{10}$ always

Madry / Spielman

ACCELERATION

- Analyzing the deletion scheme?

$$\boxed{\|f^*\|_\infty \le 1} \Rightarrow \boxed{\sum_e \left(\rho_e + \frac{1}{m}\right) f_e^2 \le m} \implies \boxed{|f_e| \le \sqrt{m}}$$

→ Delete edge w. $|f_e| \ge \rho$

→ Increase cost by $\approx \left(1 + \frac{\rho^2}{m}\right)$ factor

→ Repeat T times : $e^{\frac{T\rho^2}{m}}$

→ Deletion & $\|f\|_\infty$: small increase

→ Hence $T\frac{\rho^2}{m} \gtrsim \log n \Rightarrow$ contradiction

ACCELERATION

- Analyzing the deletion scheme?

$\boxed{\|f^*\|_\infty \leq 1} \Rightarrow \boxed{\sum_e \left(\rho_e + \frac{1}{m}\right) f_e^2 \leq m} \Rightarrow \boxed{|f_e| \leq \sqrt{m}}$

→ Delete edge w. $|f_e| \geq \rho$

→ Increase cost by $\approx \left(1 + \frac{\rho^2}{m}\right)$ factor

→ Repeat T times : $e^{\frac{T\rho^2}{m}}$

→ Deletion & $\|f\|_\infty$: small increase

SKETCH $T\frac{\rho^2}{m} \gtrsim 1$, min $T + \rho$? $T = \rho$, $\rho = m^{1/3}$

ACCELERATION

- Analyzing the deletion scheme?

$\boxed{\|f^*\|_\infty \leq 1} \Rightarrow \boxed{\sum_e \left(\rho_e + \frac{1}{m}\right) f_e^2 \leq m} \Rightarrow \boxed{|f_e| \leq \sqrt{m}}$

→ Delete edge w. $|f_e| \geq \rho$

→ Mult increase in cost by $\approx \left(1 + \frac{\rho^2}{m}\right)$

→ Repeat T times: $e^{\frac{T\rho^2}{m}}$

→ Deletion & $\|f\|_\infty$: small increase

→ Hence $T\frac{\rho^2}{m} \gtrsim \log n \Rightarrow$ contradiction

ACCELERATION

OUR OBJECTIVE

$$\min_{C\underline{\delta}=\underline{0}} \underline{g}^T \underline{\delta} + \underline{\delta}^T H \underline{\delta} + \|\delta\|_p^p$$

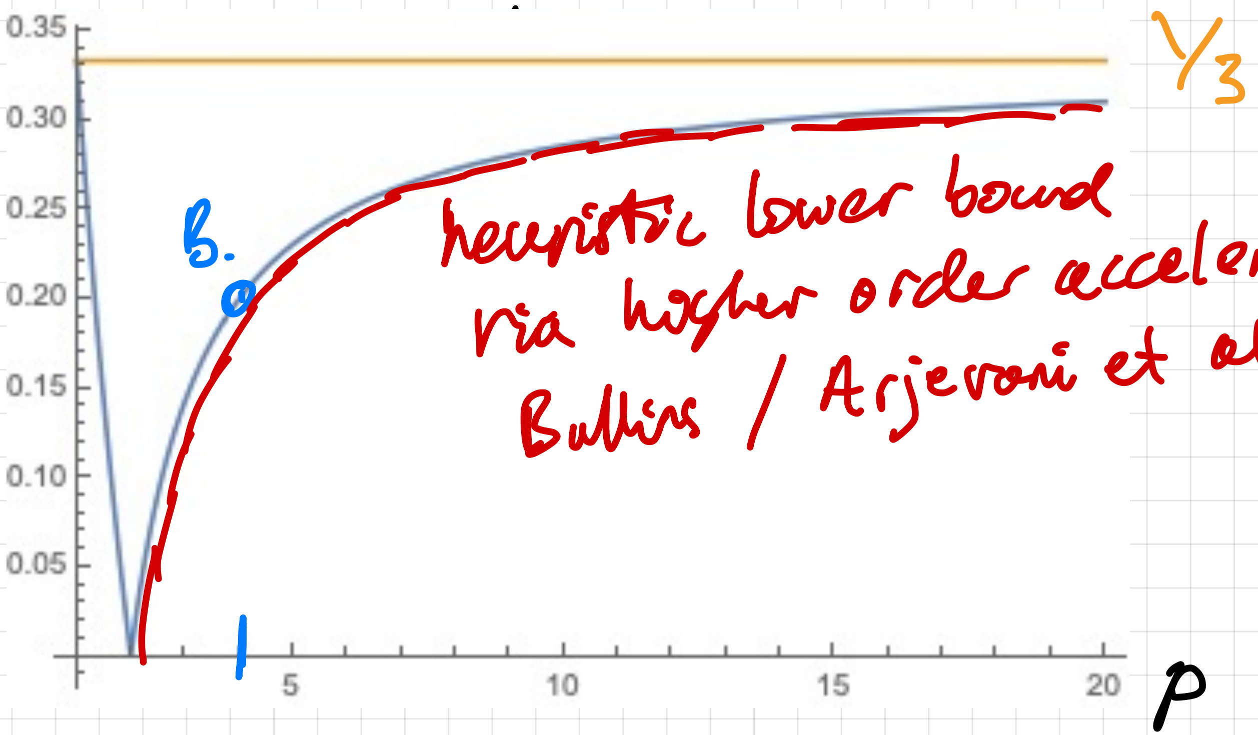
$\downarrow$

$$\min_{\substack{C\underline{\delta}=\underline{0}\\ \underline{g}^T\delta=\Theta}} \underline{\delta}^T H \underline{\delta} + \|\delta\|_p^p$$

Apply MWU to solve w. acceleration.

Keep $\|H^{1/2}\underline{\delta}\|_2$ and $\|\underline{\delta}\|_p$ small simultaneously.

ACCELERATION



• 1/2 Renegar '86

1/3 heuristic lower bound for linear programming

Spielman Madry

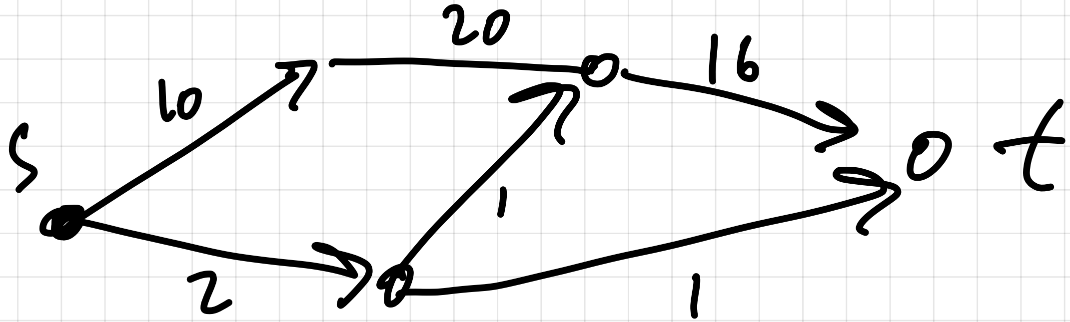
$p = \infty$

OPEN QUESTIONS

- iterative refinement for all less smooth convex functions?
  → Adil et. al recent progress
- linear programming in $\sqrt[3]{\text{DIMENSION}}$ linear eq. solves?
- formal lower bounds against randomised algorithms?

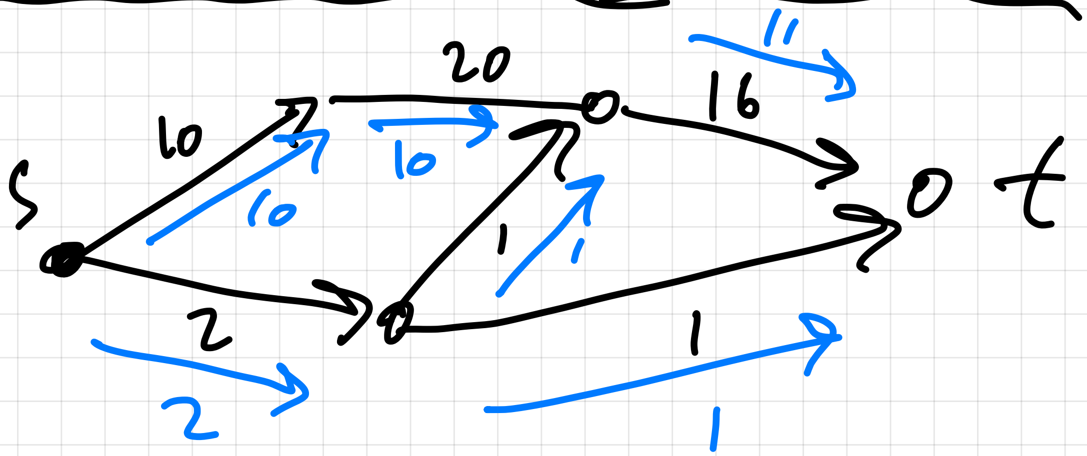
GOING FURTHER FOR FLOW PROBLEMS

MAXIMUM FLOW



GOING FURTHER FOR FLOW PROBLEMS

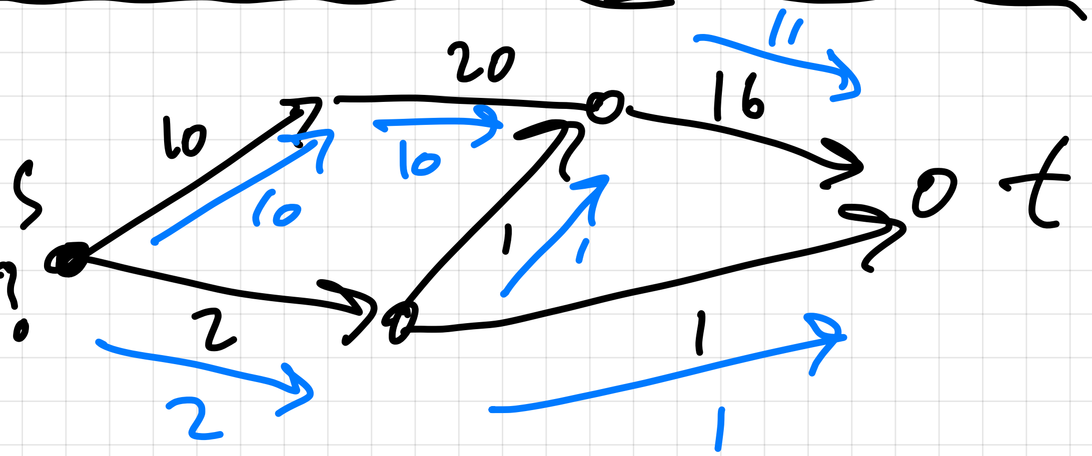
MAXIMUM FLOW



GOING FURTHER FOR FLOW PROBLEMS

MAXIMUM FLOW

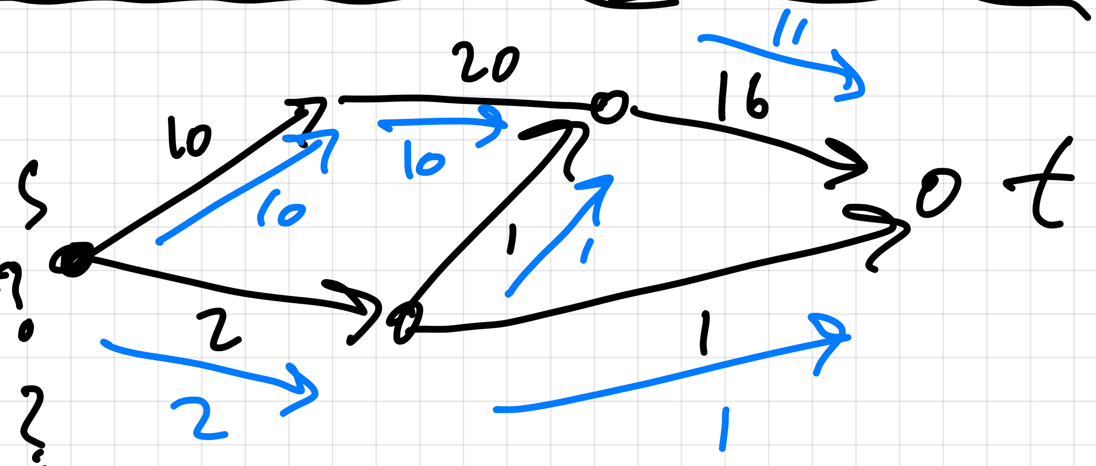
- solve with homotopy?



GOING FURTHER FOR FLOW PROBLEMS

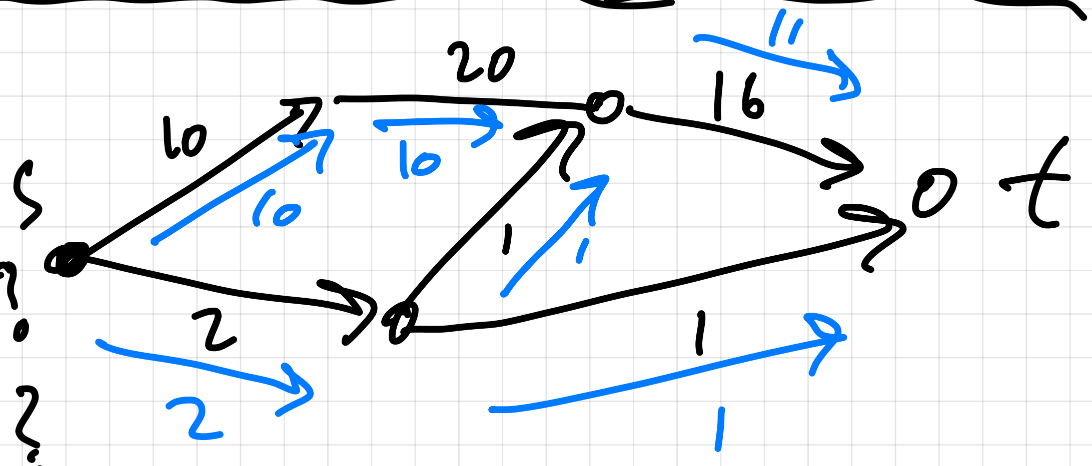
MAXIMUM FLOW

- solve with homotopy?
- linear equations?



GOING FURTHER FOR FLOW PROBLEMS

MAXIMUM FLOW



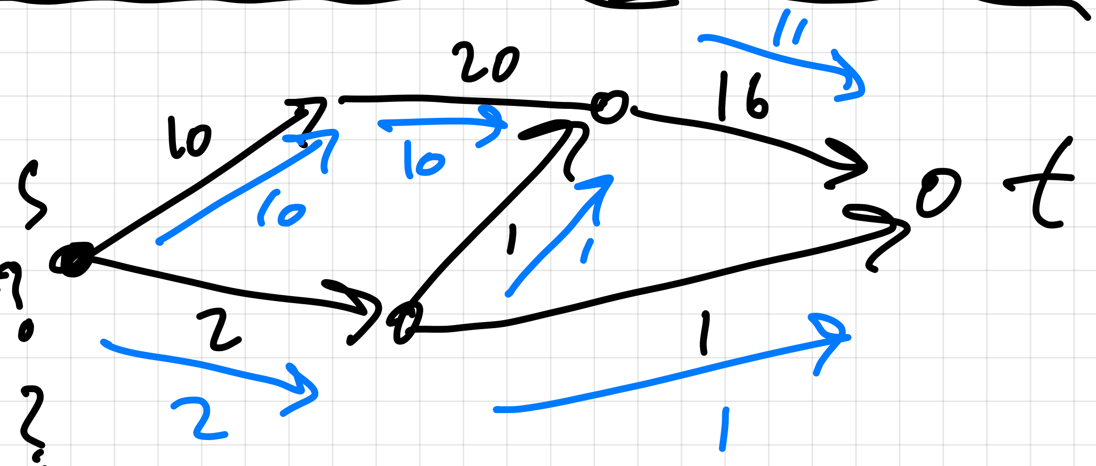
- solve with homotopy?
- linear equations?

$\Updownarrow$

- electrical flow

GOING FURTHER FOR FLOW PROBLEMS

MAXIMUM FLOW



- solve with homotopy?
- linear equations?

$\Updownarrow$

- electrical flow → fast algorithm due to Spielman & Teng

GOING FURTHER FOR FLOW PROBLEMS

- "Electrical flow"

$\Updownarrow$

- $Lx = b$ In nearly linear time

  Spielman Teng 2004

Key ingredients:

1) iterative refinement
2) sampling
3) Gaussian elimination

GOING FURTHER FOR FLOW PROBLEMS

Key ingredients:

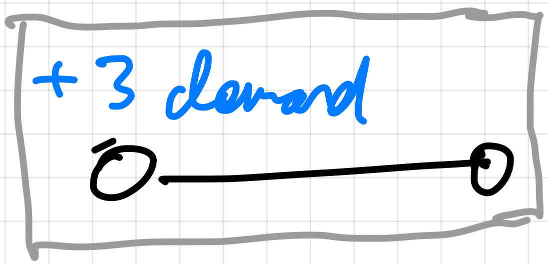
1) iterative refinement
2) sampling
3) Gaussian elimination

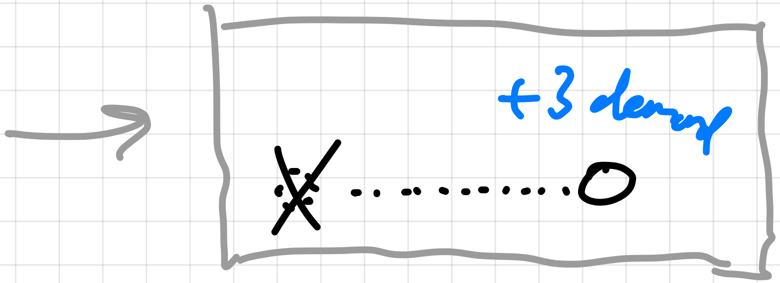
- Kyng, Peng, Sachdeva, Wang STOC 2019
  Almost linear time 2, p flow
  in unit capacity graphs
- Liu-Sidford/Kathuria STOC 2020
  Maximum flow in $\approx |E|^{1.33}$ time in unit capacity graphs.
  via 2, p-norm based interior point method
- Adil Bullins Kyng Sachdeva ICALP 2021
  Nearly linear time p-norm flow in slightly dense graphs

How to do MIXED Gaussian elimination?

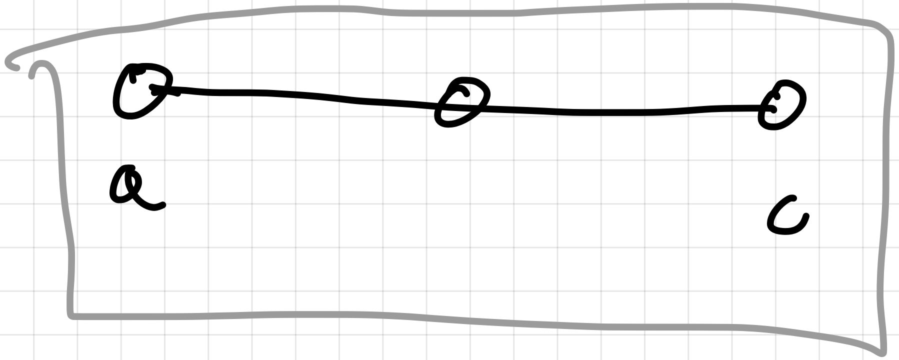
Recall: eliminate variable $\Leftrightarrow$ optimize over variable

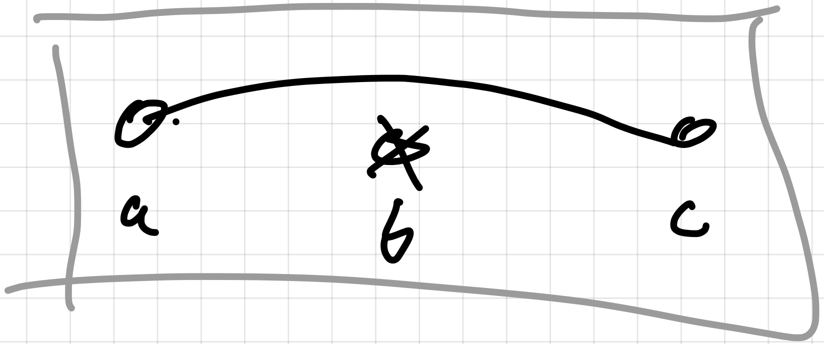
Degree 1 vertices: flow routing is trivial





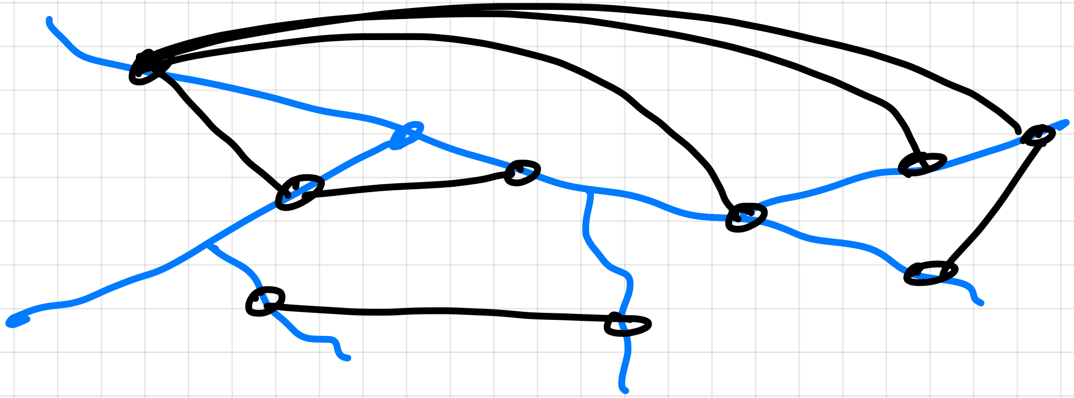
Degree 2 vertices w. ZERO DEMAND



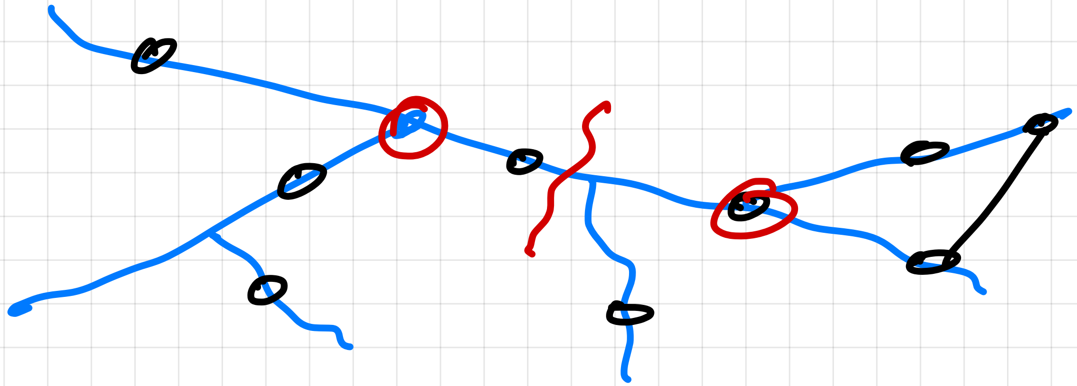


How to get many degree 1 & 2 nodes?

Rerouting:



Move edges along tree path to terminals



Goal: 1) all edges move a short distance

2) each tree edge is not "used" too much

How to get many degree 1 & 2 nodes?

Spielman-Teng '04 : Ultrasparsifiers ⇒ vertex elim for ℓ_2

Räcke '08 : oblivious routing

Madry '10 : j-trees

Sherman & Kelner-Lee-Orecchia-Sidford

efficient j trees ⇒ vertex elim for ℓ_∞

Key ingredient : low-stretch trees

Kyng-Peng-Wang-Zhang '15 : $g^T \underline{f} + f^T R f + \|f\|_p^p$

⟶ simultaneous vertex elim for weighted ℓ_2 & unweighted ℓ_∞ + handle gradients

GOING FURTHER FOR FLOW PROBLEMS

Key ingredients:

✓ 1) iterative refinement

2) sampling

✓ 3) Gaussian elimination

Simultaneous graph sparsification for $g^T f + f^T R f + \|Sf\|_p^p$

approach: 1) bucketing edges to enforce uniform wts.

2) expander decompose & enforce "nice" gradient

3) argue sampling preserves optimal solutions approximately on expanders