

ORF522 – Linear and Nonlinear Optimization

10. Interior-point methods for linear optimization

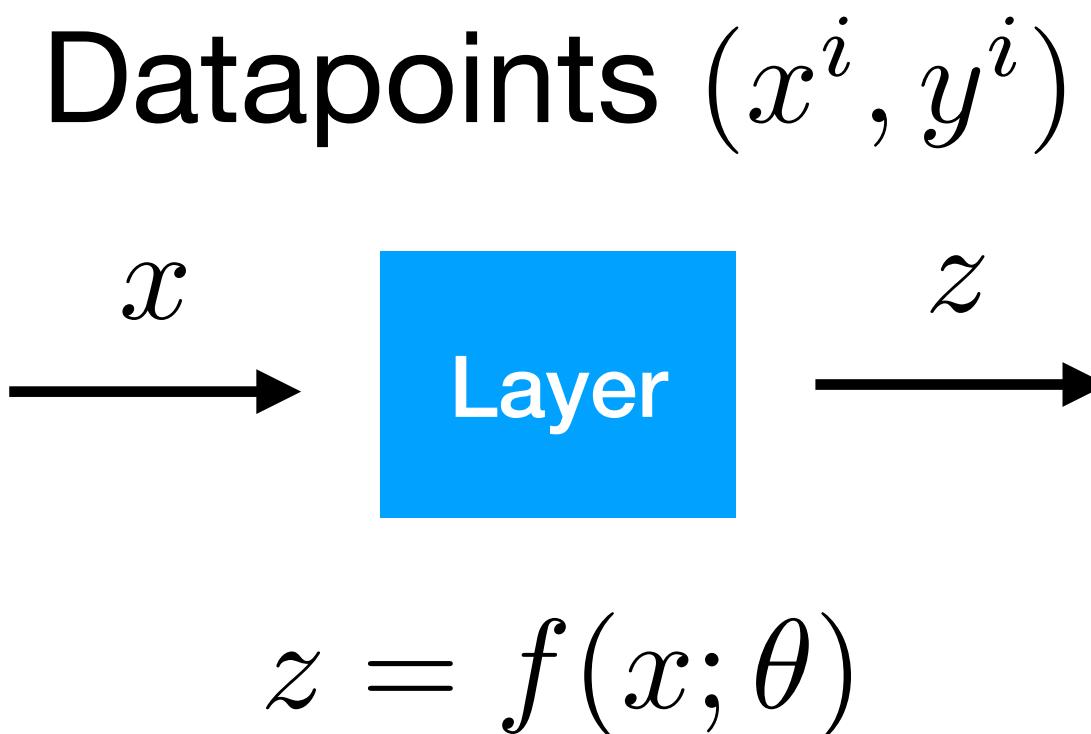
Ed forum

- how do we usually solve for x in $f(x) = 0$, where f is already given?
- we performed a global sensitivity analysis when the b vector is perturbed, but I do not think we covered the case when the constraint matrix A is changed. Can we extract information on the sensitivities of x^* and y^* with respect to such changes?
- The Sudoku problem is a discrete constraint satisfaction problem, while we have been looking at linear programs. What was the connection between this discrete version and the linear optimization layer? It appears to be a relaxation of the discrete case, since the constraints are relaxed. Did we discuss how to connect the solution to this relaxed problem back to the discrete case?
- if we want to add both a new variable and a new constraint, should we just add them sequentially using the above approach? Or is there a better way to do this?

Recap

Training a neural network

Single layer model



Training

minimize $\mathcal{L}(\theta) = \sum_{i=1}^n \ell(z^i, y^i)$

Gradient descent (more on this later)

$$\theta \leftarrow \theta - t \nabla_{\theta} \mathcal{L}(\theta)$$

Sensitivity

$$\nabla_{\theta} \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial \theta} \right)^T = \left(\frac{\partial \ell}{\partial z} \frac{\partial z}{\partial \theta} \right)^T = \left(\frac{\partial z}{\partial \theta} \right)^T \nabla_z \mathcal{L}$$

Can f be an **optimization layer**?

Optimality conditions

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & Ax \leq b\end{array}$$

Parameters: $\theta = \{c, A, b\}$
Solution $x^*(\theta)$

Solve and obtain primal-dual pair x^*, y^* (forward-pass)

Optimality conditions

$$\begin{aligned}A^T y + c &= 0 \\ \text{diag}(y)(Ax - b) &= 0\end{aligned}$$

$$y \geq 0, \quad b - Ax \geq 0$$

Mapping $r(\theta, x(\theta)) = 0$

Computing derivatives

Take differentials

$$\begin{array}{ccc} A^T y^* + c = 0 & \xrightarrow{\hspace{1cm}} & dA^T y^* + A^T dy = 0 \\ \text{diag}(y^*)(Ax - b) = 0 & & \text{diag}(Ax - b)dy + \text{diag}(y^*)(dAx^* + Adx - db) = 0 \end{array}$$

Linear system

$$\begin{bmatrix} 0 & A^T \\ \text{diag}(y^*)A & \text{diag}(Ax^* - b) \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} = - \begin{bmatrix} dA^T y^* + dc \\ \text{diag}(y^*)(dAx^* - db) \end{bmatrix}$$

Example: How does x change with b_1 ?

Set $db = e_1$, $dA = 0$, $dc = 0$ and solve the linear system.

The solution dx will correspond to $\frac{\partial x}{\partial b}$

Example

Learning to play Sudoku

			3
1			
		4	
4			1

2	4	1	3
1	3	2	4
3	1	4	2
4	2	3	1

Sudoku constraint satisfaction problem

minimize 0

subject to $Ax = b$

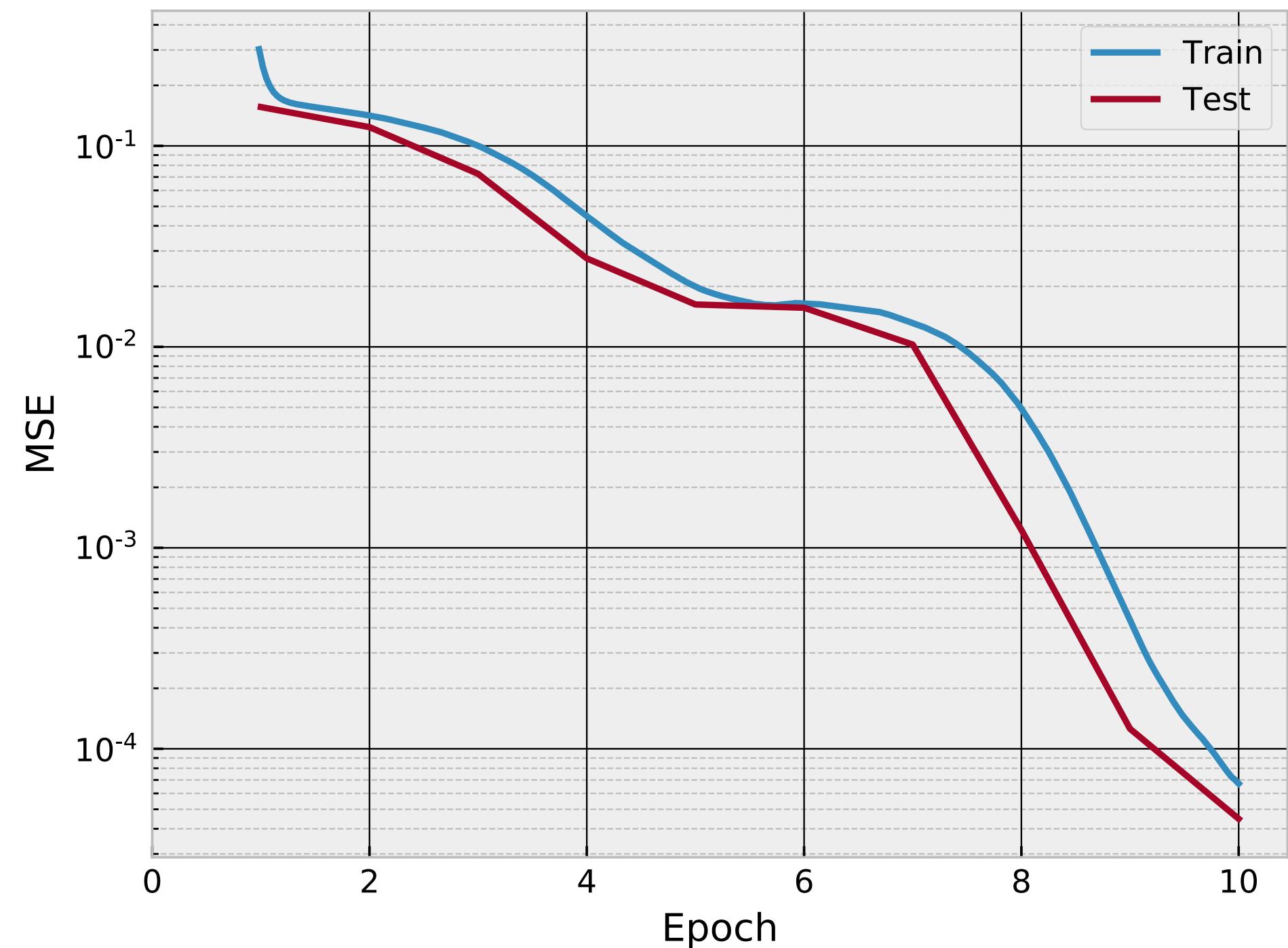
$x \in \mathbf{Z}^d$

Linear optimization layer (parameters $\theta = \{A, b\}$)

$z = x^* = \operatorname{argmin}_x 0$

subject to $Ax = b$

$x \geq 0$



Reference readings for differentiable optimization

[OptNet: Differentiable Optimization as a Layer in Neural Networks, B. Amos and J. Z. Kolter, ICML 2017]

[Differentiating Through a Conic Program, A. Agrawal, Barratt S., Boyd S., Busseti E., and Moursi W. M., Journal of Applied and Numerical Optimization, 2019]

[Differentiable Convex Optimization Layers, A. Agrawal, Amos B., Barratt S., Boyd S., Diamond S., and Kolter Z. NeurIPS 2019]

[Learning Convex Optimization Control Policies, A. Agrawal, Barratt S., Boyd S., Stellato, B., Proceedings MLR 2020]

[Learning Convex Optimization Models, A. Agrawal, Amos B., Barratt S., Boyd S., arXiv:2006.04248, 2020]

And many more...

Today's lecture

[Chapter 14, Nocedal Wright][Chapters 17/18, Vanderbei]

- History
- Newton's method
- Central path
- Primal-dual path-following algorithm

History

Ellipsoid method

Khachian (1979)

Answer to major question
Is worst-case LP complexity polynomial? Yes!

Drawbacks

Very inefficient. Much slower than simplex!

Benefits

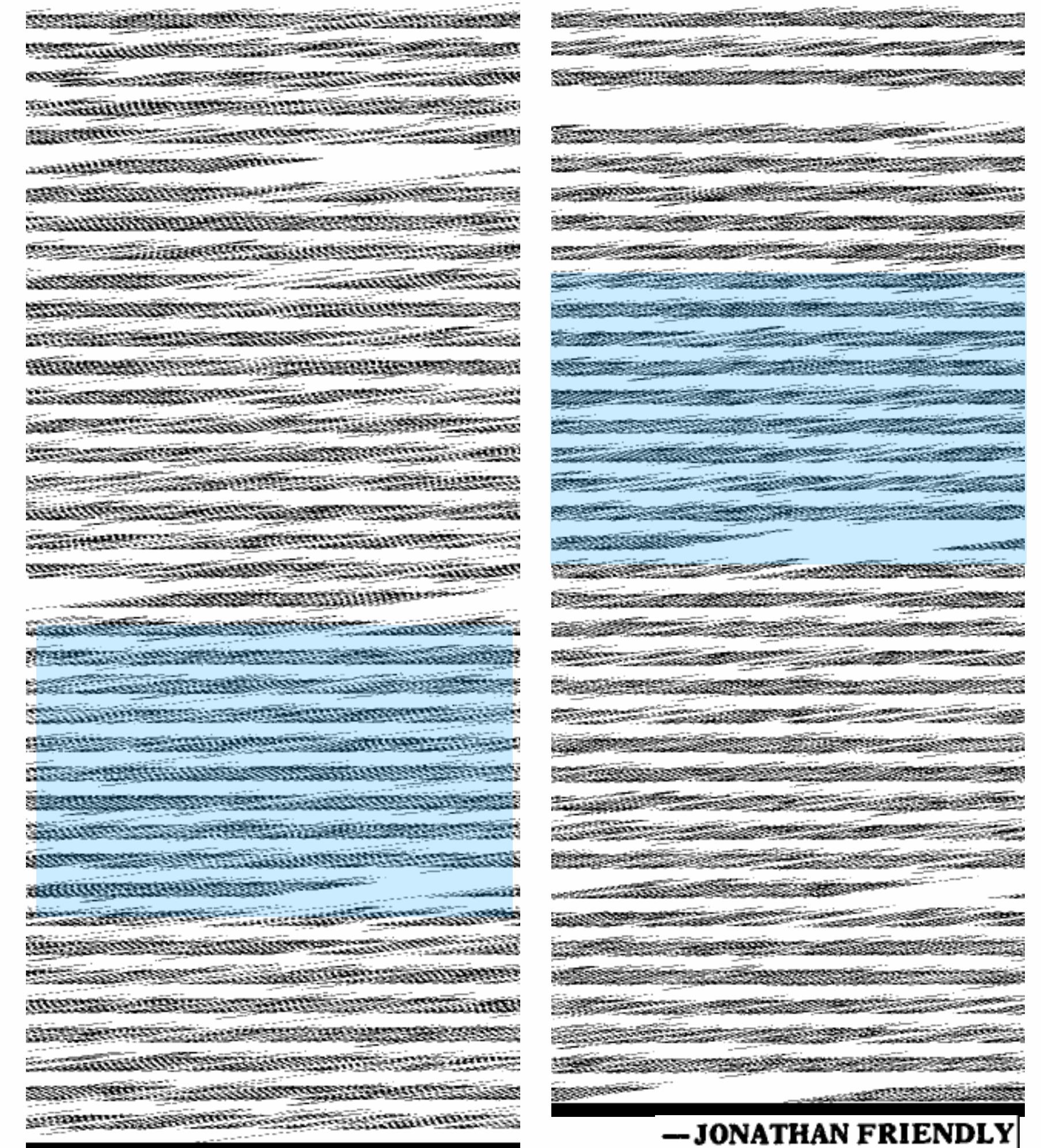
Motivated new research directions

Shazam! A Shortcut for Computers

A garment manufacturer has three kinds of dresses — A, B and C. On kind he has 17 bolts of one cloth and another, as well as 200 buttons. 25 of each. He has three cutters, 10 and 75 bakers and one finisher. Dress A, on which he makes a profit of \$1.25 a unit, requires one combination of unit, accessories and work; the material with a \$1.50 profit, takes a B dress, combination, and the \$2.25 different cost. A third set of requirements for dress C has yet to be scheduled. How should he schedule his production to make the most money? That is an easy example of a kind of problem that is eminently practical, but difficult becomes computationally variable because of the number of factors and constraints that must be handled to get a best solution. As the number of variables and constraints grows — as, for instance, in model of the national economy or in scheduling of production at any refinery — the difficulty must increase.

most powerful computers might have to run for hours to tell a plant manager how to handle a small change in, say, the amount of crude oil being delivered to his tanks. And adding one new restriction can substantially increase the number of possible answers and thus the time required to check them for an optimum solution.

Last week, intrigued mathematicians were trying to sort out the meaning of what looked like a stun-



— JONATHAN FRIENDLY

The New York Times

Published: November 11, 1979
Copyright © The New York Times

Interior-point methods

1950s-1960s: nonlinear convex optimization

- Sequential unconstrained optimization (Fiacco & McCormick), Logarithmic barrier method (Frish), affine scaling method (Dikin), etc.
 - No worst-case complexity theory but often good practical performance

1980s-1990s: interior point methods

- Karmarkar's algorithm (1984)
 - Competitive with simplex, often faster for larger problems

plicated systems, often with thousands of variables. They arise in a variety of commercial and government applications, ranging from allocating time on a communications satellite to routing millions of telephone calls over long distances. Linear programming is particularly useful whenever a limited, expensive resource must be spread most efficiently among competing users. And investment companies use the approach in creating portfolios with the best mix of stocks and bonds.

Power Solutions Seen

Labs mathematician, Dr. Karmarkar, has devised a new procedure that may routine handling of such problems that are now far

a path-breaking result," Donald L. Graham, director of optical sciences for Bell Labs in Holmdel, N.J. "Science has its mo-



Newton's method

Newton's method for nonlinear equations

Goal: solve
$$h(x) = 0$$

Derivative

$$Dh = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \dots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m}{\partial x_1} & \dots & \frac{\partial h_m}{\partial x_n} \end{bmatrix}$$

First-order approximation

$$h(x) \approx h(\bar{x}) + Dh(\bar{x})(x - \bar{x}) \longrightarrow$$

Iteratively set to zero

$$h(x^k) + Dh(x^k)(x^{k+1} - x^k) = 0$$

Iterations

- Solve $Dh(x^k)\Delta x = -h(x^k)$
- $x^{k+1} \leftarrow x^k + \Delta x$

Newton method

Convergence

Iterations

- Solve $Dh(x^k)\Delta x = -h(x^k)$
- $x^{k+1} \leftarrow x^k + \Delta x$

Remarks

- Iterations can be **expensive** (linear system solution)
- **Fast (quadratic) convergence** close to the solution x^*

Symmetric primal-dual problems

	→	Primal	
minimize	$c^T x$	maximize	$-b^T y$
subject to	$Ax \leq b$	subject to	$A^T y + c = 0$
	$Ax + s = b$		$y \geq 0$
	$s \geq 0$		

Optimality conditions

$$Ax + s - b = 0$$

$$A^T y + c = 0$$

$$s_i y_i = 0$$

$$s, y \geq 0$$

Main idea

Optimality conditions

$$h(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY\mathbf{1} \end{bmatrix} = 0$$
$$s, y \geq 0$$
$$S = \text{diag}(s)$$
$$Y = \text{diag}(y)$$

- Apply variants of Newton's method to solve $h(x, s, y) = 0$
- Enforce $s, y > 0$ (strictly) at every iteration
- **Motivation** avoid getting stuck in “corners”

Duality measure

Definition

$$\mu = \frac{s^T y}{m}$$

Average value of the pairs $s_i y_i$

It describes the “desirability” of each point in the search space

Newton's method for optimality conditions

Linear system

$$Dh \begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} -h \\ -r_p \\ -r_d \\ -SY1 \end{bmatrix}$$

Residuals

$$r_p = Ax + s - b$$

$$r_d = A^T y + c$$

Line search to enforce $x, s > 0$

$$(x, s, y) \leftarrow (x, s, y) + \alpha(\Delta x, \Delta s, \Delta y)$$

Issue

Pure **Newton's step** does not allow significant progress towards

$$h(x, s, y) = 0 \text{ and } x, y \geq 0.$$

Central path

Smoothed optimality conditions

Optimality conditions

$$Ax + s - b = 0$$

$$A^T y + c = 0$$

$$s_i y_i = \tau \quad \longleftarrow \quad \text{Same } \tau \text{ for every pair}$$

$$s, y \geq 0$$

Same optimality conditions for a “smoothed” version of our problem

Newton's method for smoothed optimality conditions

Smoothed optimality conditions

$$h_\tau(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY\mathbf{1} - \tau\mathbf{1} \end{bmatrix} = 0$$
$$s, y \geq 0$$

Linear system

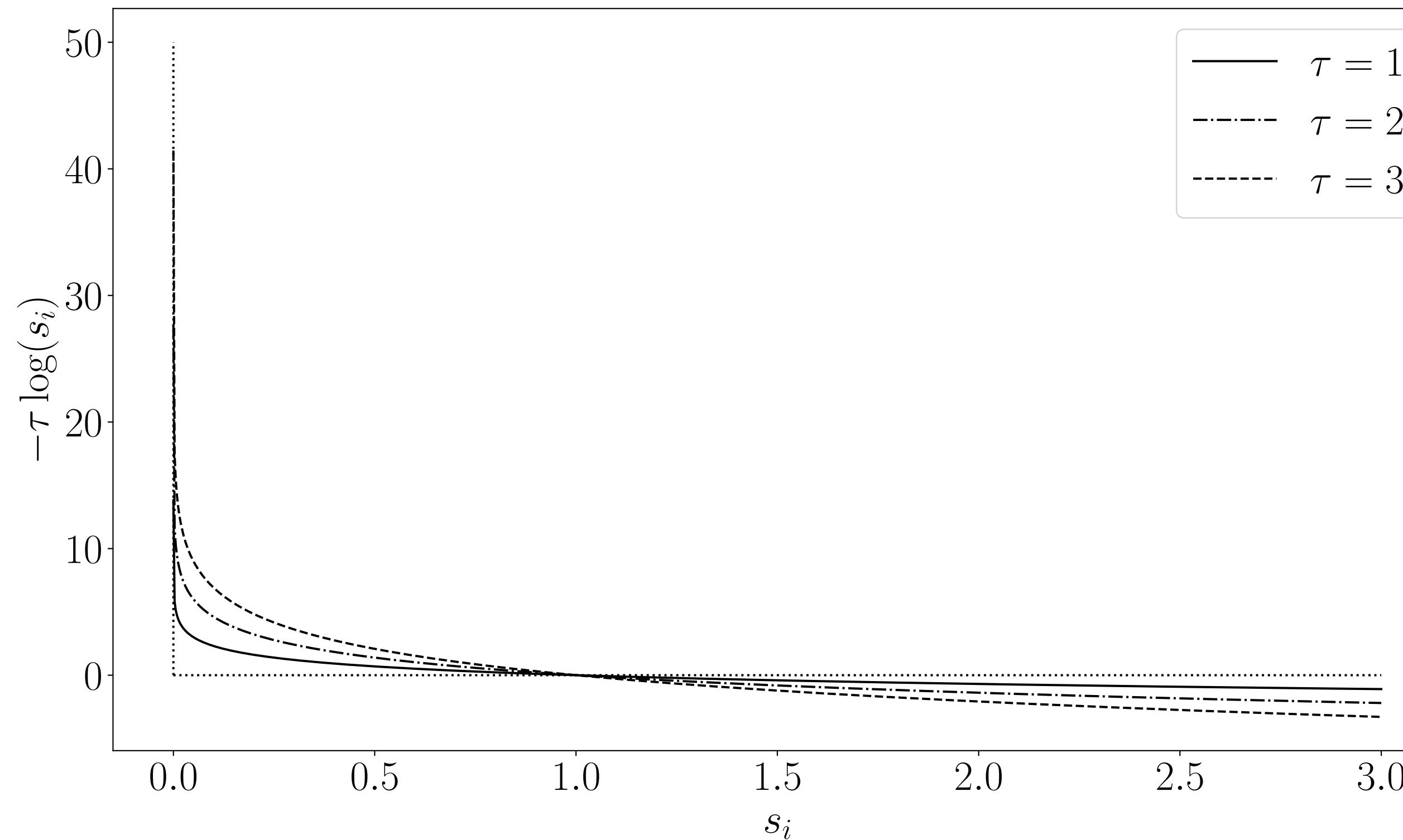
$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY + \tau\mathbf{1} \end{bmatrix}$$

Line search to enforce $x, s > 0$

$$(x, s, y) \leftarrow (x, s, y) + \alpha(\Delta x, \Delta s, \Delta y)$$

Logarithmic barrier

$$\phi(s) = -\tau \sum_{i=1}^m \log(s_i) \quad \text{on domain} \quad s_i > 0$$



As $\tau \rightarrow 0$ it approximates

$$\mathcal{I}_{s_i \geq 0} = \begin{cases} 0 & \text{if } s_i \geq 0 \\ \infty & \text{otherwise} \end{cases}$$

Smoothed problem

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax + s = b \\ & s \geq 0 \end{array} \longrightarrow \begin{array}{ll} \text{minimize} & c^T x + \phi(x) = c^T x - \tau \sum_{i=1}^m \log(s_i) \\ \text{subject to} & Ax + s = b \end{array}$$

Dual cost

$$g(y) = \underset{x, s}{\text{minimize}} \mathcal{L}(x, s, y) = c^T x + \phi(s) + y^T (Ax + s - b)$$

$$\frac{\partial \mathcal{L}}{\partial x} = A^T y + c = 0$$

$$\frac{\partial \mathcal{L}}{\partial s_i} = -\tau \frac{1}{s_i} + y_i = 0 \implies s_i y_i = \tau$$

Central path

$$\begin{array}{ll}\text{minimize} & c^T x - \tau \sum_{i=1}^m \log(s_i) \\ \text{subject to} & Ax + s = b\end{array}$$

Set of points $(x^*(\tau), s^*(\tau), y^*(\tau))$
with $\tau > 0$ such that

$$Ax + s - b = 0$$

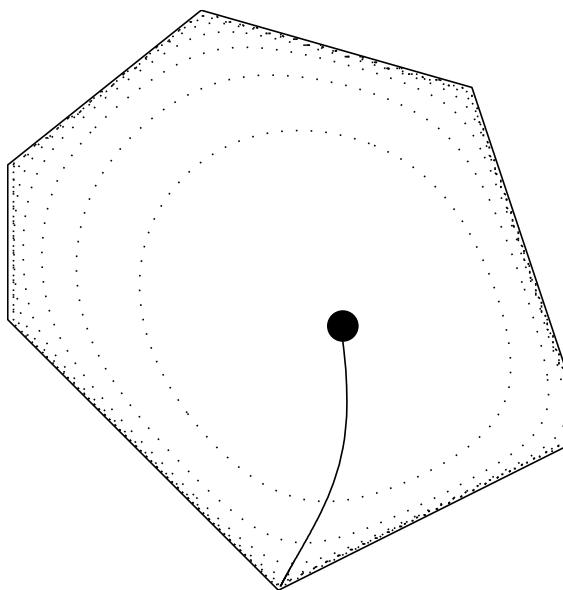
$$A^T y + c = 0$$

$$s_i y_i = \tau$$

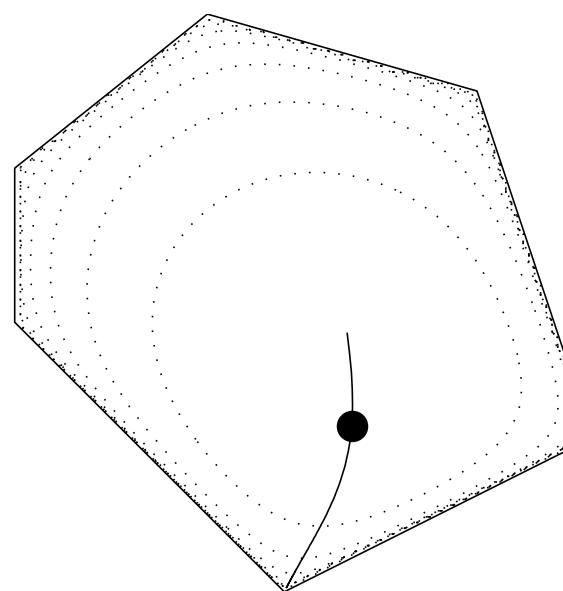
$$s, y \geq 0$$

Analytic
Center

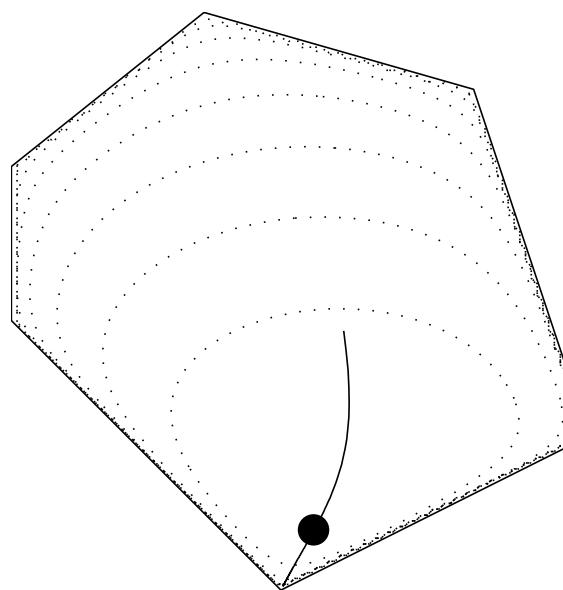
$$\tau \rightarrow \infty$$



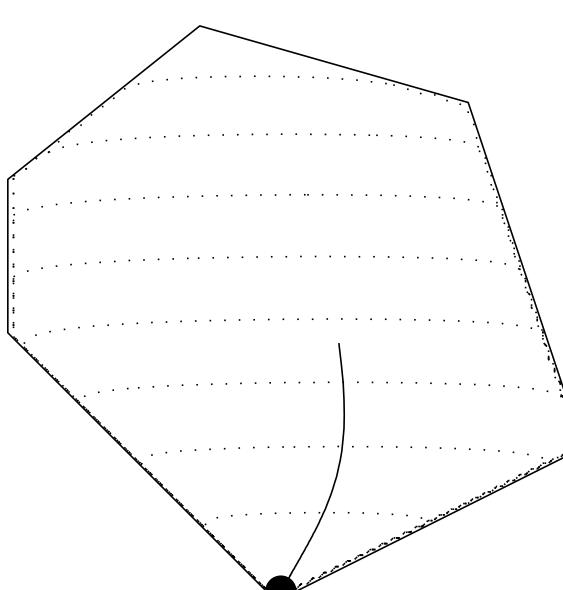
1000



1



1/5



1/100

τ

Main idea

Follow central path as $\tau \rightarrow 0$

25

Primal-dual path-following method

Algorithm step

Linear system

$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} + \sigma\mu\mathbf{1} \end{bmatrix}$$

Duality measure
 $\mu = \frac{s^T y}{m}$

Centering parameter

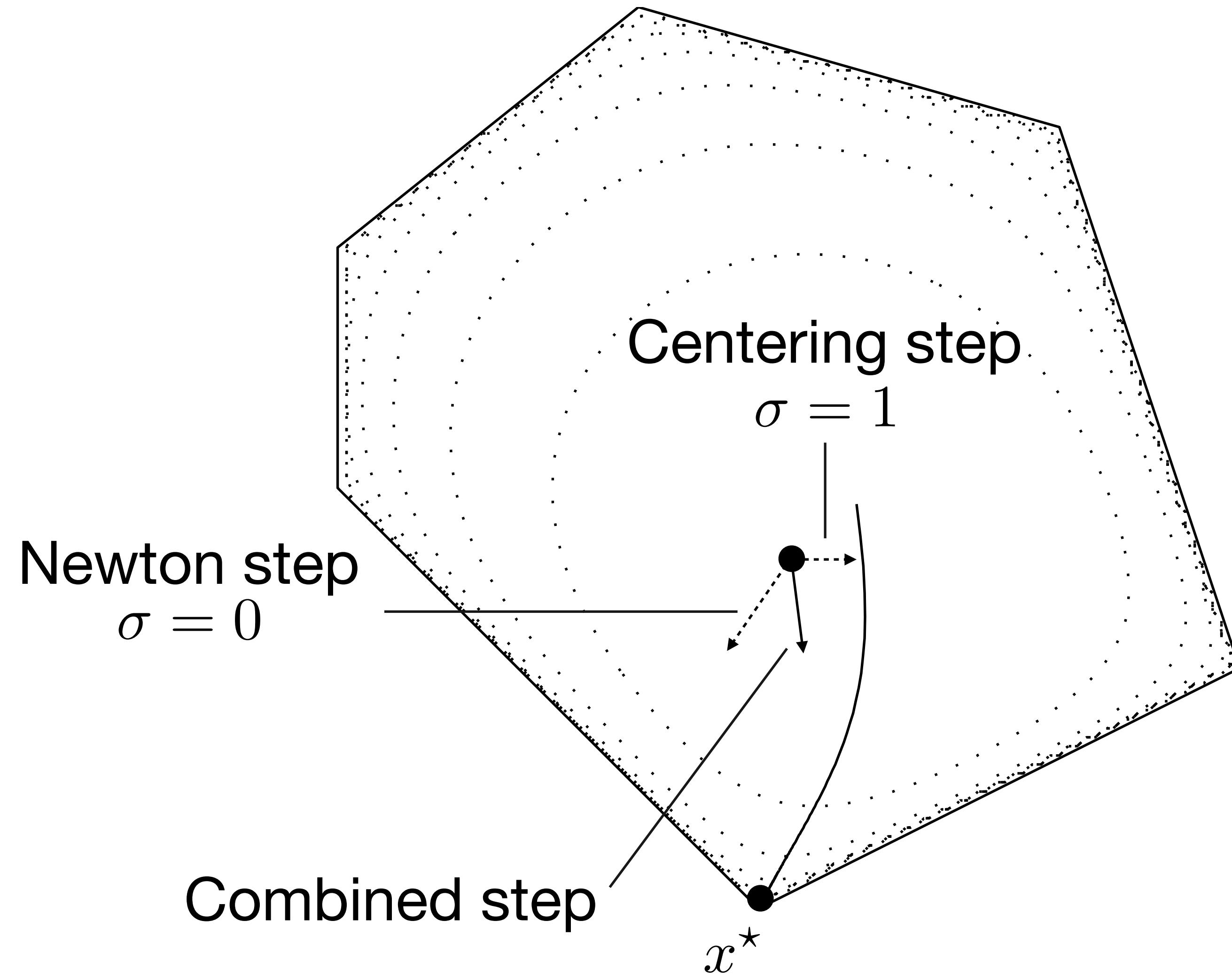
$$\sigma \in [0, 1]$$

$\sigma = 0 \Rightarrow$ Newton step

$\sigma = 1 \Rightarrow$ Centering step towards $(x^\star(\mu), s^\star(\mu), y^\star(\mu))$

Line search to enforce $x, s > 0$
 $(x, s, y) \leftarrow (x, s, y) + \alpha(\Delta x, \Delta s, \Delta y)$

Path-following algorithm idea



Centering step

It brings towards the **central path** and is usually biased towards $s, y > 0$.

No progress on duality measure μ

Newton step

It brings towards the **zero duality measure** μ . Quickly violates $s, y > 0$.

Combined step

Best of both worlds with longer steps

Primal-dual path-following algorithm

Initialization

- Given (x_0, s_0, y_0) such that $s_0, y_0 > 0$

Iterations

- Choose $\sigma \in [0, 1]$

2. Solve
$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} + \sigma\mu\mathbf{1} \end{bmatrix}$$
 where $\mu = s^T y/m$

- Find maximum α such that $y + \alpha\Delta y > 0$ and $s + \alpha\Delta s > 0$
- Update $(x, s, y) \leftarrow (x, s, y) + \alpha(\Delta x, \Delta s, \Delta y)$

Working towards optimality conditions

Optimality conditions satisfied **only at convergence**

Primal residual

$$r_p = Ax + s - b \rightarrow 0$$

Stopping criteria

$$\|r_p\| \leq \epsilon_{\text{pri}}$$

Dual residual

$$r_d = A^T y + c \rightarrow 0$$

$$\|r_d\| \leq \epsilon_{\text{dua}}$$

Complementary slackness

$$s^T y \rightarrow 0$$

$$s^T y \leq \epsilon_{\text{gap}}$$

Convergence

Definitions

Primal-dual strictly feasible set

$$\mathcal{F}^\circ = \{(x, s, y) \mid Ax + s = b, A^T y + c = 0, s, y > 0\}$$

Central path neighborhood

$$\mathcal{N}(\gamma) = \{(x, s, y) \in \mathcal{F}^\circ \mid s_i y_i \geq \gamma \mu\} \quad \text{with } \gamma \in (0, 1] \quad (\text{almost all the feasible region})$$

Theorem

[Page 402-406, Nocedal Wright]

Smallest decrement

$$\mu_{k+1} \leq \left(1 - \frac{\delta}{m}\right) \mu_k \quad \text{with } \delta > 0$$

Iteration complexity

Given $(x_0, s_0, y_0) \in \mathcal{N}(\gamma)$, there exists $K = O(n \log(1/\epsilon))$ such that

$$\mu_k \leq \epsilon \mu_0 \quad \text{for all } k \geq K$$

Modified versions achieve $O(\sqrt{n} \log(1/\epsilon))$

Interior-point methods for linear optimization

Today, we learned to:

- **Apply** Newton's method to solve optimality conditions
- **Analyze** the central path and the smoothed optimality conditions
- **Develop** a prototype primal-dual path-following algorithm

Next lecture

- Practical interior-point method (Mehrotra predictor-corrector algorithm)
- Linear algebra implementation details
- Linear optimization recap