ORF307 – Optimization

18. Interior-point methods II

Bartolomeo Stellato — Spring 2023

Ed Forum

- **2nd Midterm: April 18**
  Time: 11:00am — 12:20pm
  Students with extensions please reach out to me
  Location: Same room as lecture
  Topics: linear optimization
  Material allowed: Single sheet of paper. Double sided. Hand-written or typed.
  Exercises to prepare: past midterm + extra exercises on canvas

- **Questions**

  - How are tau, sigma, and mu related?

  - I was still a little confused by SY1. Why do we need to include it in the matrix?

SY1

Recap

(Sparse) Cholesky factorization

Every positive definite matrix A can be factored as

$$A = PLL^T P^T \longrightarrow P^T AP = LL^T$$

P permutation, L lower triangular

(Sparse) Cholesky factorization

Every positive definite matrix A can be factored as

$$A = PLL^T P^T \longrightarrow P^T AP = LL^T$$

P permutation, L lower triangular

Permutations

- Reorder rows/cols of A with P to (heuristically) get **sparser** L
- P depends only on sparsity pattern of A (unlike LU factorization)
- If A is dense, we can set $P = I$

(Sparse) Cholesky factorization

Every positive definite matrix A can be factored as

$$A = PLL^T P^T \longrightarrow P^T AP = LL^T$$

P permutation, L lower triangular

Permutations

- Reorder rows/cols of A with P to (heuristically) get **sparser** L
- P depends only on sparsity pattern of A (unlike LU factorization)
- If A is dense, we can set $P = I$

Cost

- If A dense, typically $O(n^3)$ but usually much less
- It depends on the number of nonzeros in A, sparsity pattern, etc.
- Typically 50% faster than LU (need to find only one matrix)

Linear optimization as a root finding problem

Optimality conditions

minimize $\quad c^T x$

subject to $\quad Ax \leq b$

Linear optimization as a root finding problem

Optimality conditions

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

Linear optimization as a root finding problem

Optimality conditions

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

KKT conditions

$$Ax + s - b = 0$$

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

$$s_i y_i = 0, \quad i = 1, \dots, m$$

$$s, y \geq 0$$

1. PRIMAL FEASIBILITY
2. DUAL FEASIBILITY
3. COMPLEMENT SLACKNESS

Linear optimization as a root finding problem

$$
\begin{aligned}
Ax + s - b &= 0 \\
A^T y + c &= 0 \\
s_i y_i &= 0, \quad i = 1, \ldots, m \\
s, y &\geq 0
\end{aligned}
$$

Linear optimization as a root finding problem

$$
\begin{aligned}
Ax + s - b &= 0 \\
A^T y + c &= 0 \\
s_i y_i &= 0, \quad i = 1, \ldots, m \\
s, y &\geq 0
\end{aligned}
$$

Diagonalize complementary slackness

$$
S = \mathbf{diag}(s) = \begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_m \end{bmatrix}
$$

$$
SY\mathbf{1} = \mathbf{diag}(s)\mathbf{diag}(y)\mathbf{1} = \begin{bmatrix} s_1 y_1 & & & \\ & s_2 y_2 & & \\ & & \ddots & \\ & & & s_m y_m \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} s_1 y_1 \\ s_2 y_2 \\ \vdots \\ s_m y_m \end{bmatrix}
$$

$$
Y = \mathbf{diag}(y) = \begin{bmatrix} y_1 & & & \\ & y_2 & & \\ & & \ddots & \\ & & & y_m \end{bmatrix}
$$

Linear optimization as a root finding problem

$$
\begin{aligned}
Ax + s - b &= 0 \\
A^T y + c &= 0 \\
s_i y_i &= 0, \quad i = 1, \dots, m \\
s, y &\geq 0
\end{aligned}
$$

Diagonalize complementary slackness

$$
S = \mathbf{diag}(s) = \begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_m \end{bmatrix}
$$

$$
Y = \mathbf{diag}(y) = \begin{bmatrix} y_1 & & & \\ & y_2 & & \\ & & \ddots & \\ & & & y_m \end{bmatrix}
$$

$$
SY\mathbf{1} = \mathbf{diag}(s)\mathbf{diag}(y)\mathbf{1} = \begin{bmatrix} s_1 y_1 & & & \\ & s_2 y_2 & & \\ & & \ddots & \\ & & & s_m y_m \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} s_1 y_1 \\ s_2 y_2 \\ \vdots \\ s_m y_m \end{bmatrix}
$$

$$
s_i y_i = 0, \quad i = 1, \dots, m \quad \Longleftrightarrow \quad SY\mathbf{1} = 0
$$

Main idea

Optimality conditions

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

$$s, y \geq 0$$

$$S = \mathbf{diag}(s)$$
$$Y = \mathbf{diag}(y)$$

- 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"

Smoothed optimality conditions

Optimality conditions

$$Ax + s - b = 0$$

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

$$s_i y_i = \tau$$ ← Same τ for every pair

$$s, y \geq 0$$

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

Smoothed optimality conditions

Optimality conditions

$$Ax + s - b = 0$$

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

$$s_i y_i = \tau$$ ← Same τ for every pair

$$s, y \geq 0$$

Same optimality conditions for a "smoothed" version of our problem

Duality gap

$$m\tau = s^T y = (b - Ax)^T y = b^T x - x^T A^T y = b^T y + c^T x$$

Central path

minimize $\quad c^T x - \tau \sum_{i=1}^m \log(s_i)$
subject to $\quad Ax + s = b$

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

$$
\begin{aligned}
Ax + s - b &= 0 \\
A^T y + c &= 0 \\
s_i y_i &= \tau \\
s, y &\geq 0
\end{aligned}
$$

Central path

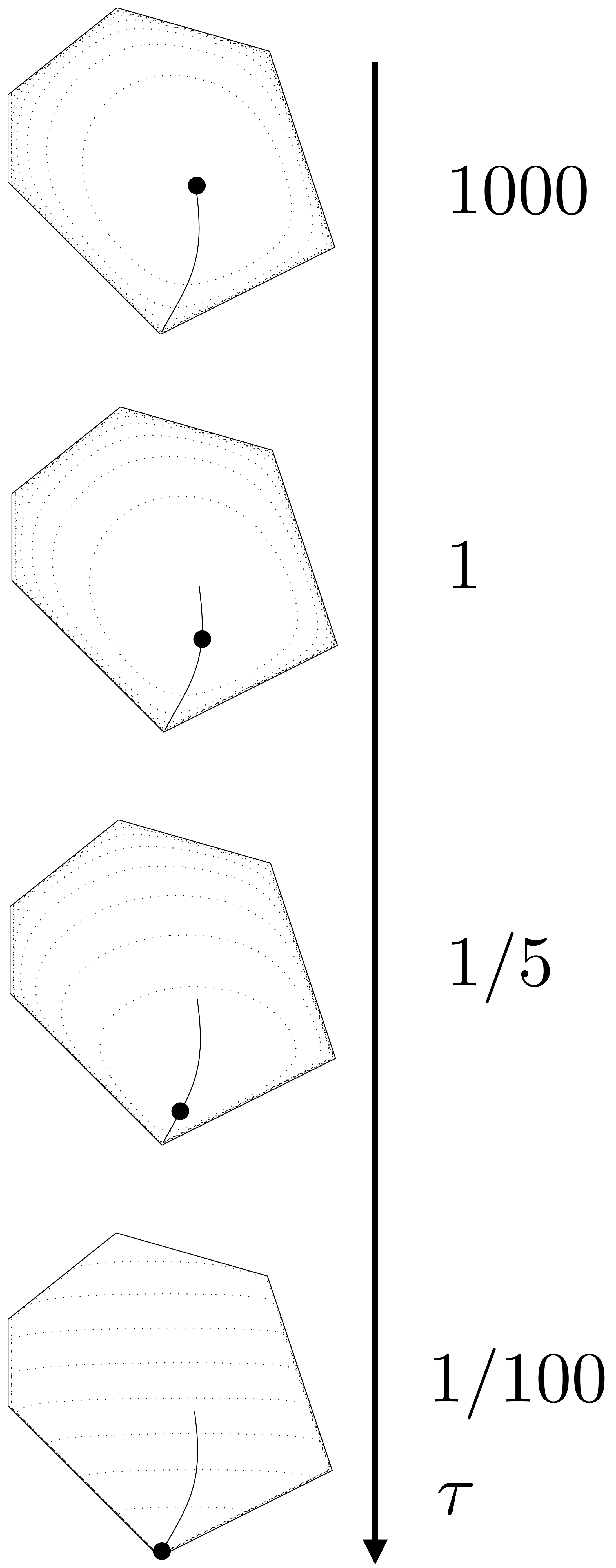
minimize $\quad c^T x - \tau \sum_{i=1}^{m} \log(s_i)$

subject to $\quad Ax + s = b$

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

$$\begin{aligned} Ax + s - b &= 0 \\ A^T y + c &= 0 \\ s_i y_i &= \tau \\ s, y &\geq 0 \end{aligned}$$

Analytic Center
$\tau \to \infty$



Main idea

Follow central path as $\tau \to 0$

The path parameter

Duality measure

$$\mu = \frac{s^T y}{m}$$ (average value of the pairs $s_i y_i$)

$\tau = \sigma\mu$

The path parameter

Duality measure

$$\mu = \frac{s^T y}{m}$$ (average value of the pairs $s_i y_i$)

Linear system

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

The path parameter

Duality measure

$$\mu = \frac{s^T y}{m}$$ (average value of the pairs $s_i y_i$)

Centering parameter

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

Linear system

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

The path parameter

$\tau = \sigma\mu$

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

$$\mu = \frac{s^T y}{m}$$ (average value of the pairs $s_i y_i$)

Centering parameter

$\sigma \in [0, 1]$

$\sigma = 0 \quad \Rightarrow \quad$ Newton step

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

$\tau = \mu$

The path parameter

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

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

(average value of the pairs $s_i y_i$)

Centering parameter

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

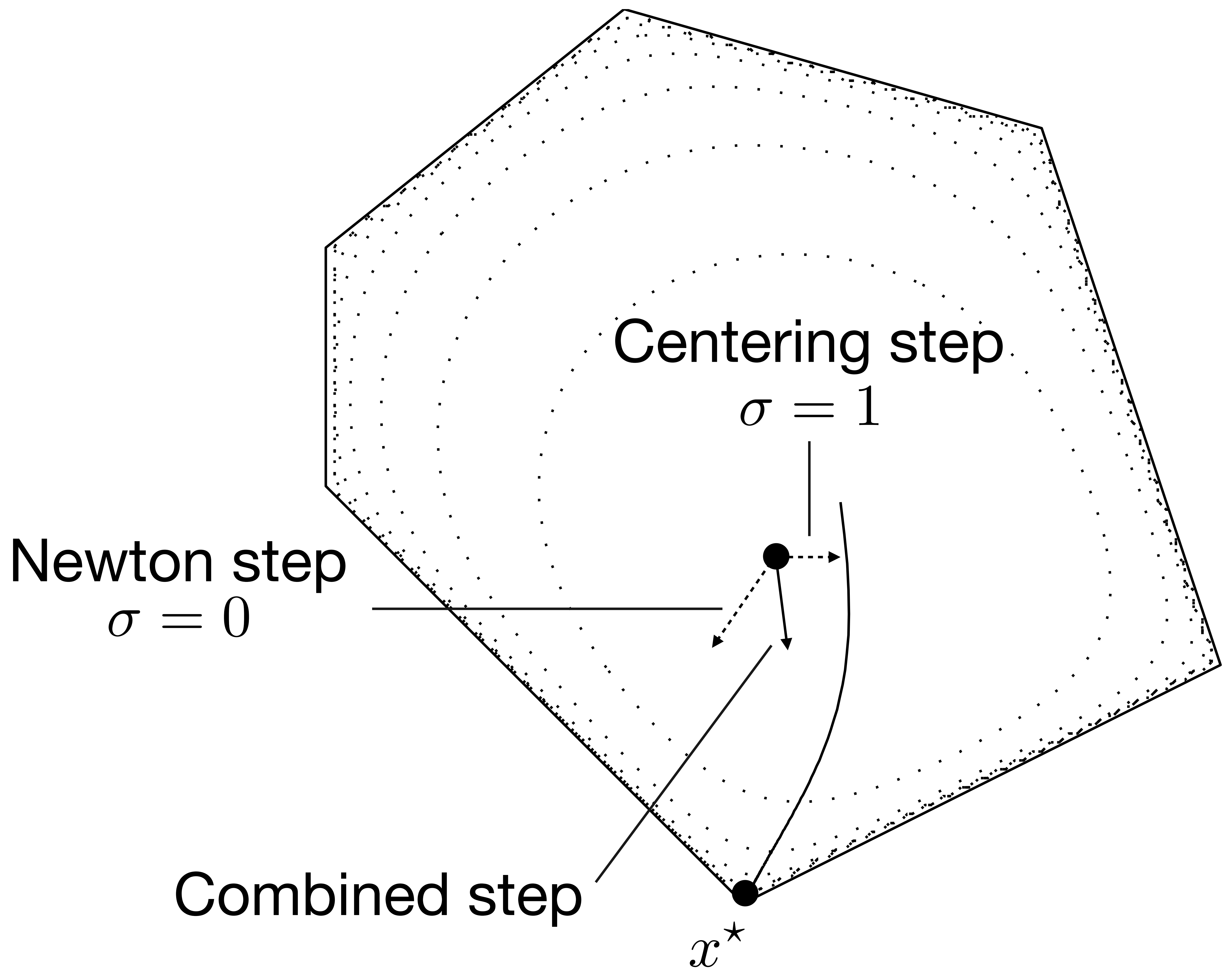
$\sigma = 0 \quad \Rightarrow \quad$ Newton step

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

Line search to enforce $s, y > 0$

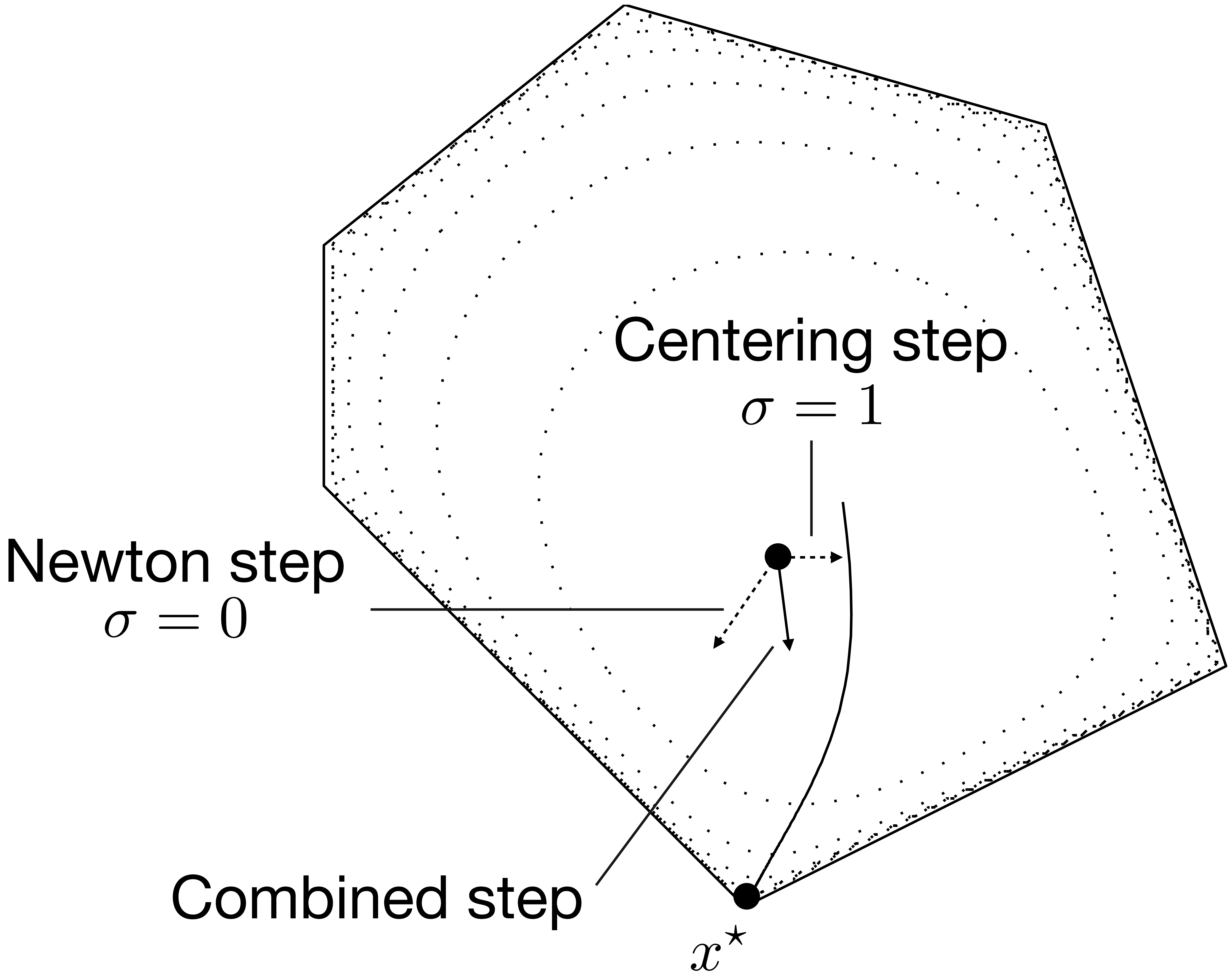
$$(y, x, s) \leftarrow (y, x, s) + \alpha(\Delta y, \Delta x, \Delta s)$$

Path-following algorithm idea



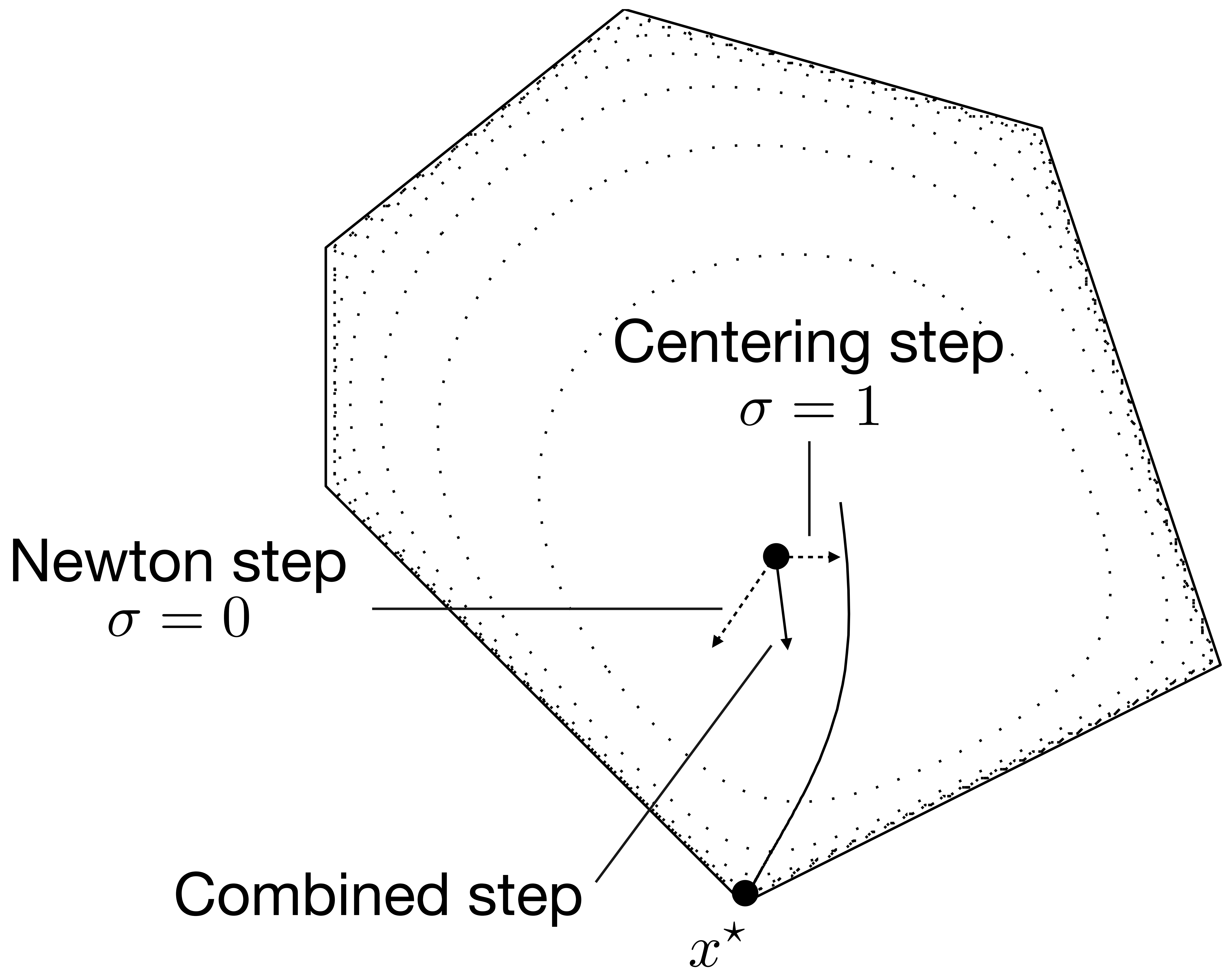
Path-following algorithm idea

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



Centering step
It brings towards the **central path** and is usually biased towards $s, y > 0$.
No progress on duality measure μ

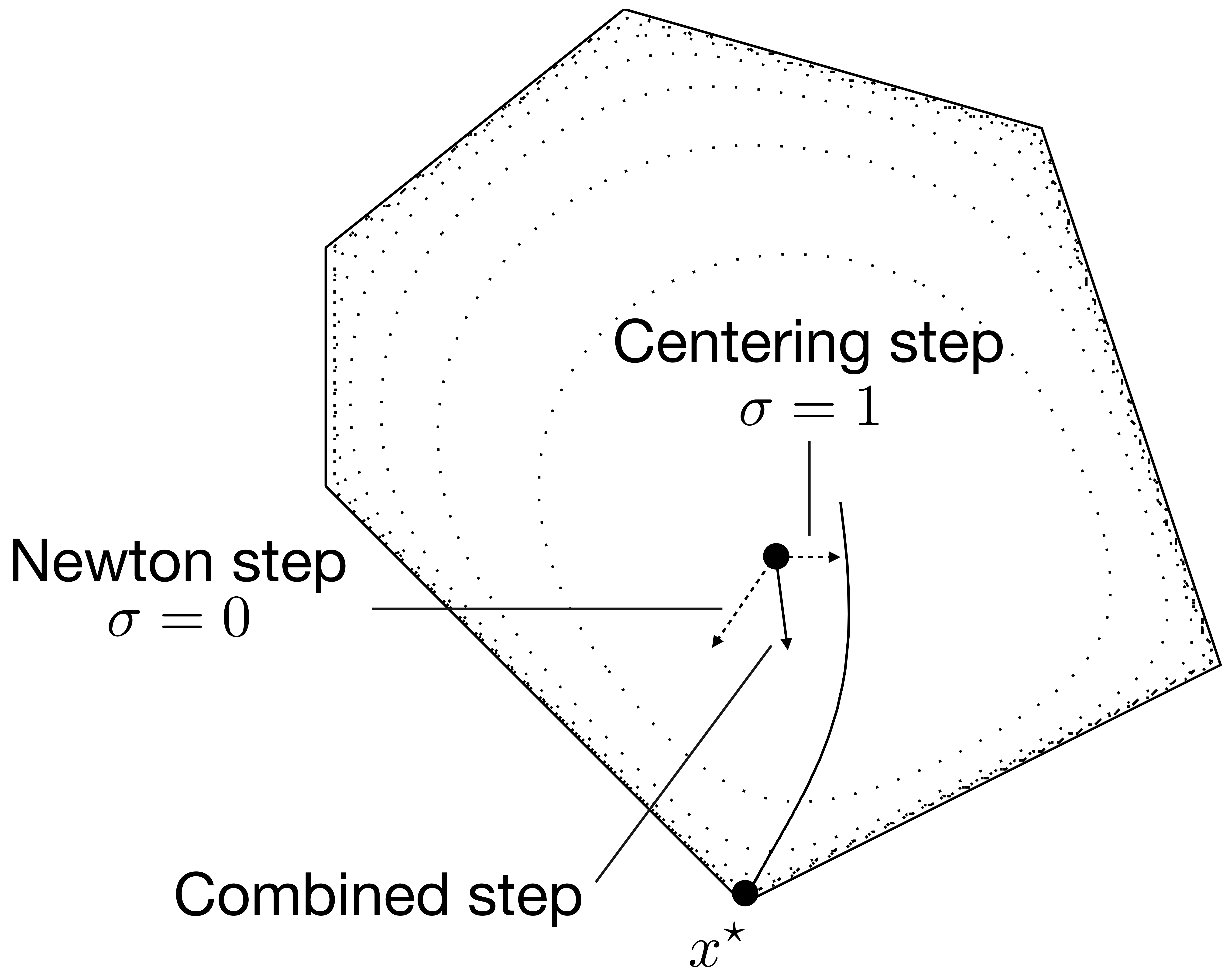
Path-following algorithm idea



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 $s, y > 0$.

Path-following algorithm idea

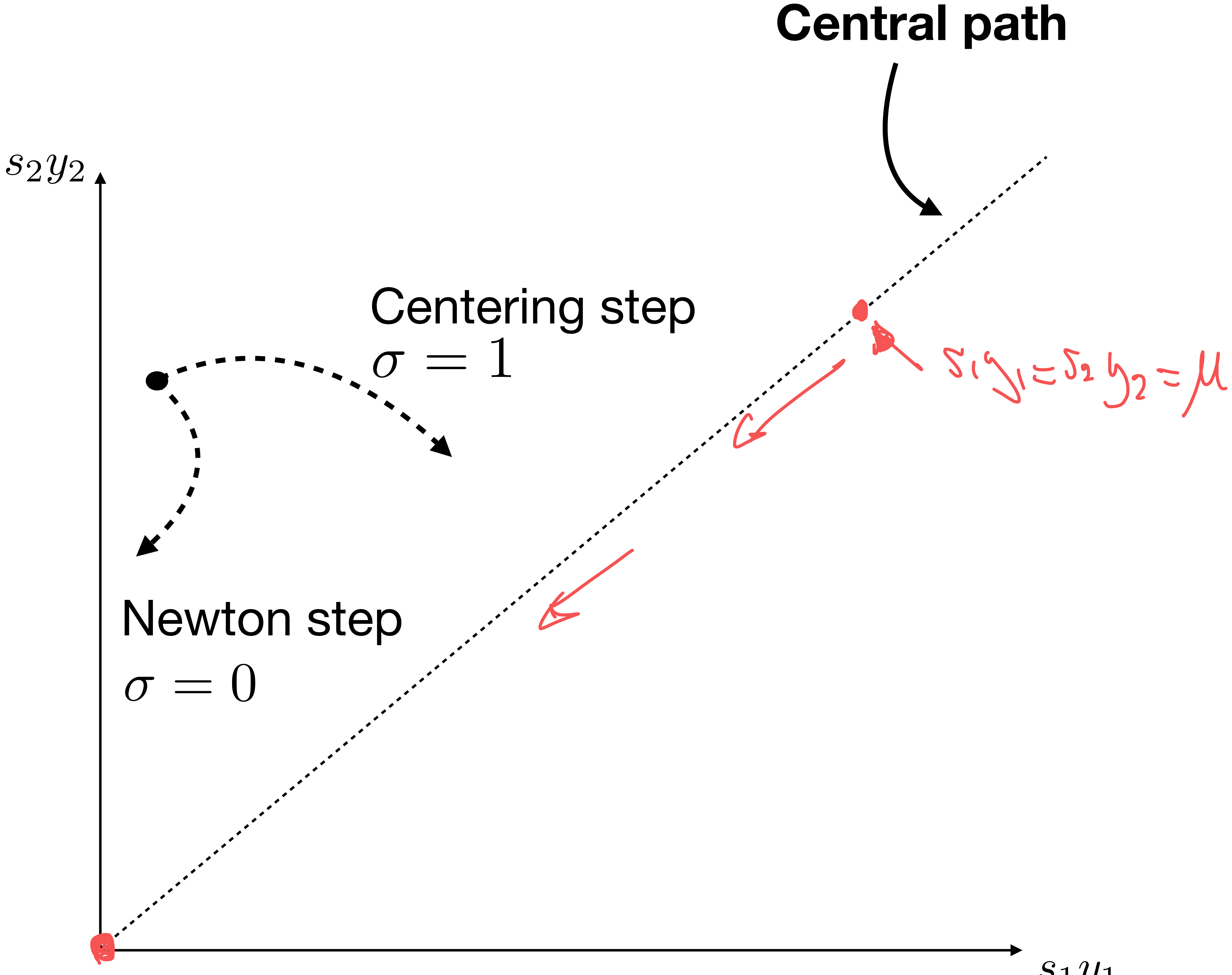


Centering step
It brings towards the **central path** and is usually biased towards $s, y > 0$.
No progress on duality measure μ

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

Combined step
Best of both worlds with longer steps

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

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

Iterations

1. 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 $\mu = s^T y / m$
3. Find maximum α such that $y + \alpha\Delta y > 0$ and $s + \alpha\Delta s > 0$
4. Update $(y, x, s) \leftarrow (y, x, s) + \alpha(\Delta y, \Delta x, \Delta s)$

Today's lecture

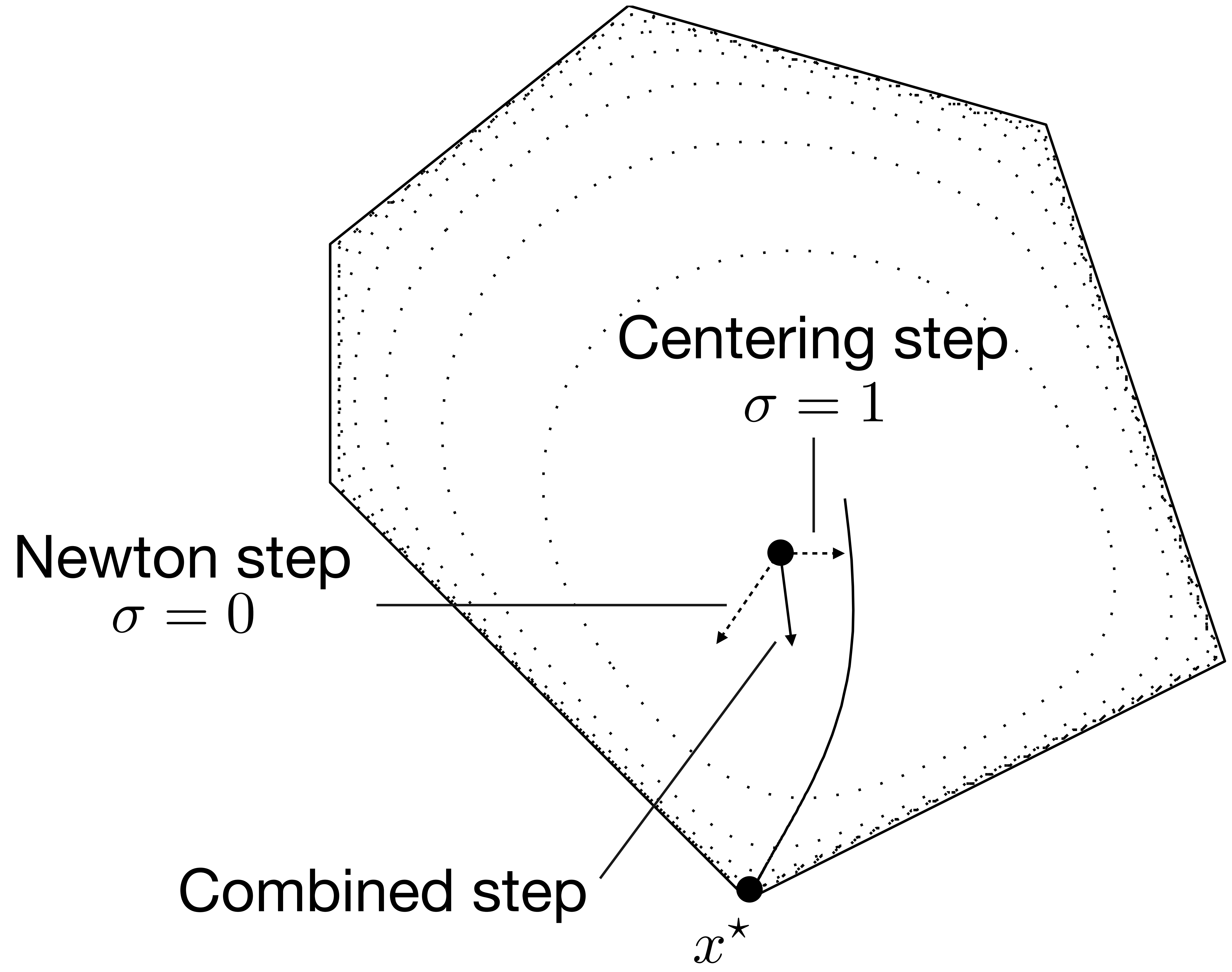
Interior-point methods II

- Mehrotra predictor-corrector algorithm
- Implementation and linear algebra
- Interior-point vs simplex

Predictor-corrector algorithm

Main idea

Predict and select centering parameter

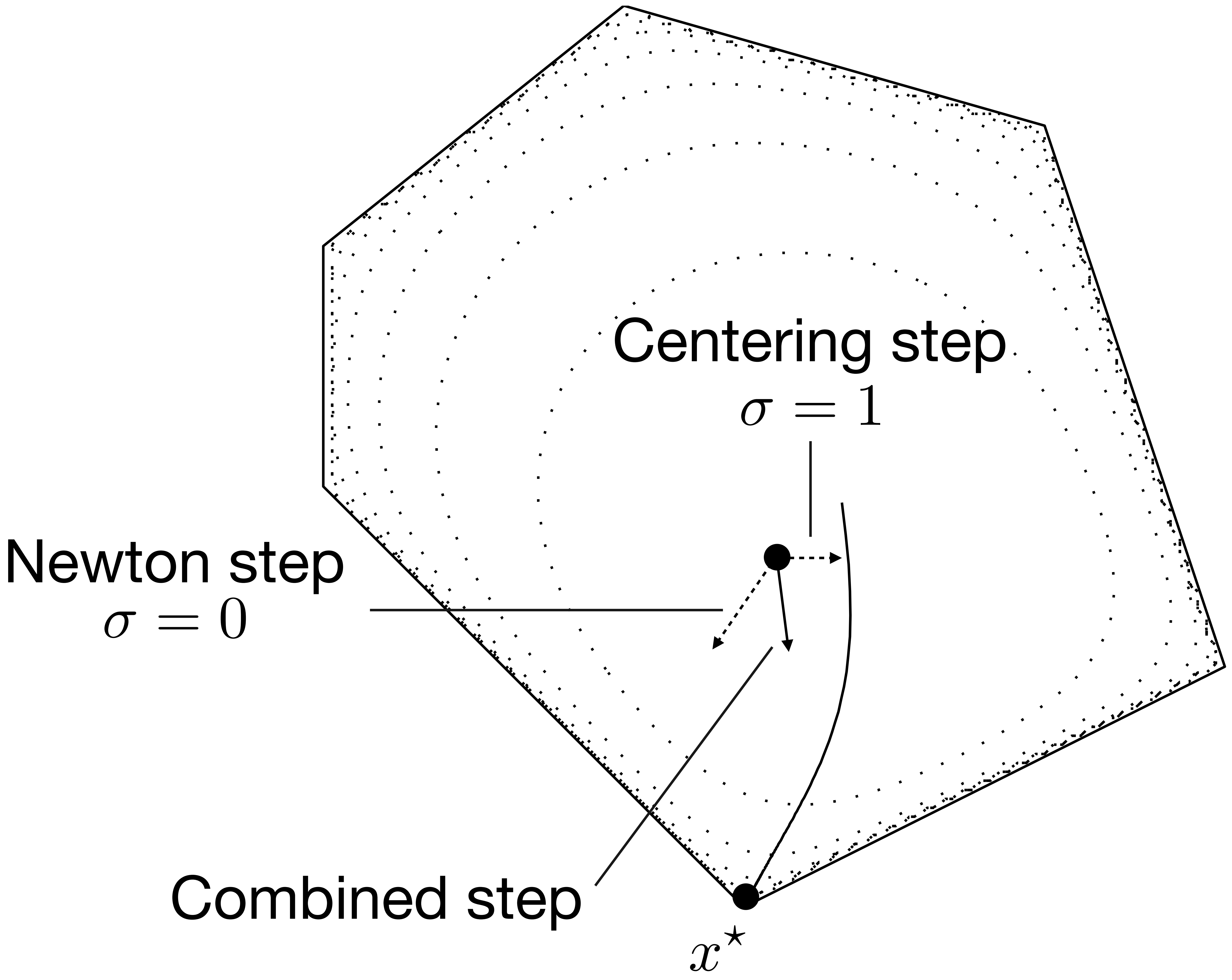


Predict

Compute Newton direction

Main idea

Predict and select centering parameter



Predict

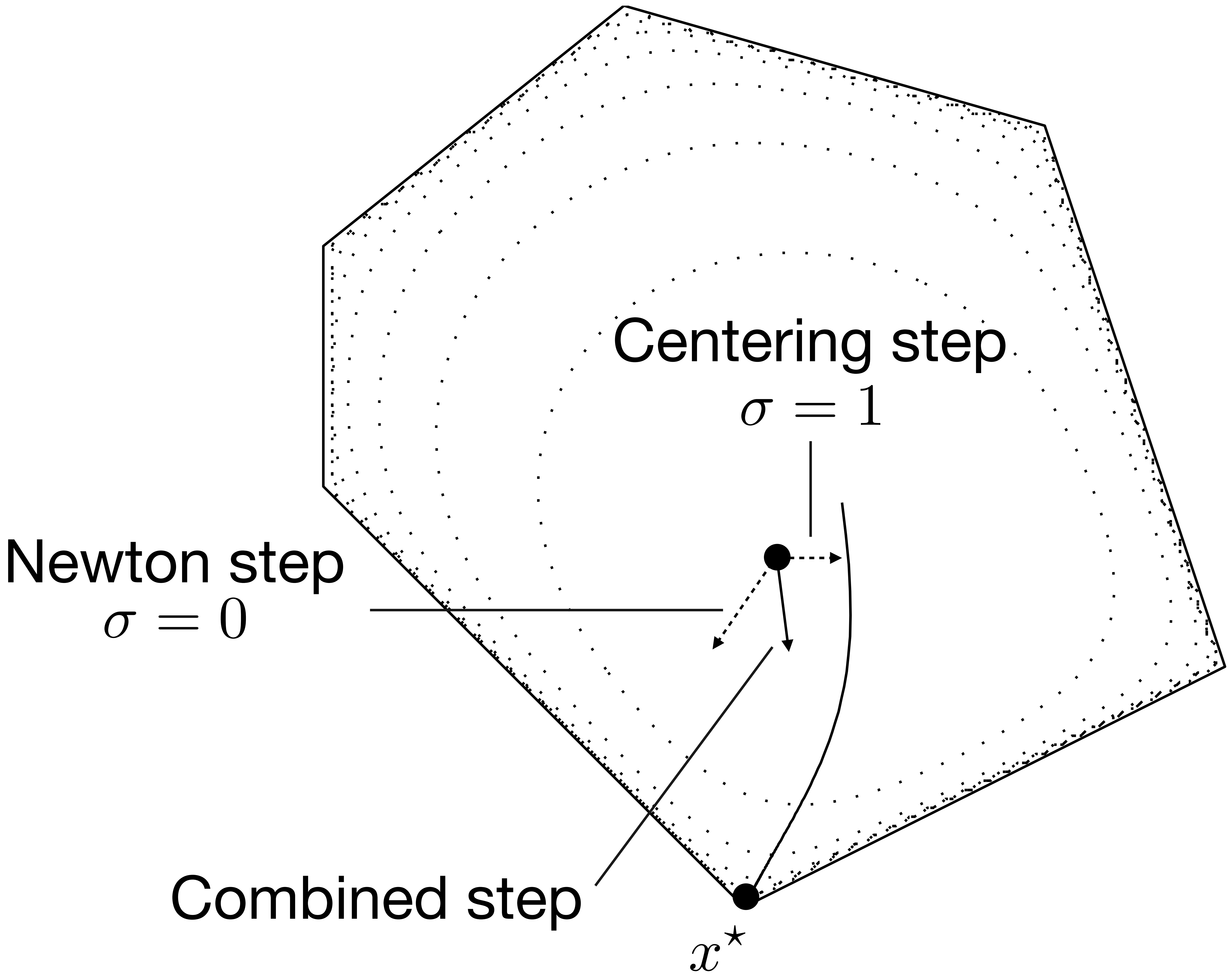
Compute Newton direction

Estimate

How good is the Newton step?
(how much can μ decrease?)

Main idea

Predict and select centering parameter



Predict

Compute Newton direction

Estimate

How good is the Newton step?
(how much can μ decrease?)

Select centering parameter

Very roughly:
Pick $\sigma \approx 0$ if Newton step is good
Pick $\sigma \approx 1$ if Newton step is bad

How good is the Newton step?

Newton step

$$(\Delta x_a, \Delta s_a, \Delta y_a)$$

Maximum step-size

$$\alpha_p = \max\{\alpha \in [0,1] \mid s + \alpha \Delta s_a \geq 0\}$$
$$\alpha_d = \max\{\alpha \in [0,1] \mid y + \alpha \Delta y_a \geq 0\}$$

How good is the Newton step?

Newton step

$$(\Delta x_a, \Delta s_a, \Delta y_a)$$

Maximum step-size

$$\alpha_p = \max\{\alpha \in [0,1] \mid s + \alpha \Delta s_a \geq 0\}$$
$$\alpha_d = \max\{\alpha \in [0,1] \mid y + \alpha \Delta y_a \geq 0\}$$

Two issues

- The new points will not produce much improvement: $(s + \alpha_p \Delta s_a)_i (y + \alpha_d \Delta y_a)_i$ much larger than 0
- The complementarity error depends on step lengths α_p and α_d

Choosing a centering parameter to make good improvement

Newton step

$$(\Delta x_a, \Delta s_a, \Delta y_a)$$

Maximum step-size

$$\alpha_p = \max\{\alpha \in [0,1] \mid s + \alpha \Delta s_a \geq 0\}$$
$$\alpha_d = \max\{\alpha \in [0,1] \mid y + \alpha \Delta y_a \geq 0\}$$

Choosing a centering parameter to make good improvement

Newton step

$$(\Delta x_a, \Delta s_a, \Delta y_a)$$

Maximum step-size

$$\alpha_p = \max\{\alpha \in [0,1] \mid s + \alpha \Delta s_a \geq 0\}$$
$$\alpha_d = \max\{\alpha \in [0,1] \mid y + \alpha \Delta y_a \geq 0\}$$

Duality measure candidate
(after Newton step)

$$\mu_a = \frac{(s + \alpha_p \Delta s_a)^T (y + \alpha_d \Delta y_a)}{m}$$

Choosing a centering parameter to make good improvement

Newton step

$$(\Delta x_a, \Delta s_a, \Delta y_a)$$

Maximum step-size

$$\alpha_p = \max\{\alpha \in [0,1] \mid s + \alpha \Delta s_a \geq 0\}$$
$$\alpha_d = \max\{\alpha \in [0,1] \mid y + \alpha \Delta y_a \geq 0\}$$

Duality measure candidate
(after Newton step)

$$\mu_a = \frac{(s + \alpha_p \Delta s_a)^T (y + \alpha_d \Delta y_a)}{m}$$

⟶

Centering parameter heuristic σ

$$\sigma = \left(\frac{\mu_a}{\mu}\right)^3$$

Correcting for complementary error

Newton step

$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y_a \\ \Delta x_a \\ \Delta s_a \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} \end{bmatrix}$$

Correcting for complementary error

Newton step

$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y_a \\ \Delta x_a \\ \Delta s_a \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} \end{bmatrix} \longrightarrow \quad s_i(\Delta y_a)_i + y_i(\Delta s_a)_i + s_i y_i = 0$$

Correcting for complementary error

Newton step

$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y_a \\ \Delta x_a \\ \Delta s_a \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} \end{bmatrix} \longrightarrow$$

$$s_i(\Delta y_a)_i + y_i(\Delta s_a)_i + s_i y_i = 0$$

Complementarity error

$$(s_i+(\Delta s_a)_i)(y_i + (\Delta y_a)_i) = (\Delta s_a)_i(\Delta y_a)_i \neq 0$$

$$s_i y_i + s_i(\Delta y_a)_i + (\Delta s_a)_i y_i + (\Delta s_a)_i(\Delta y_a)_i$$

Complementarity violation
depends on step length

Correcting for complementary error

Newton step

$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y_a \\ \Delta x_a \\ \Delta s_a \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} \end{bmatrix} \longrightarrow \quad s_i(\Delta y_a)_i + y_i(\Delta s_a)_i + s_i y_i = 0$$

Complementarity error

$$(s_i + (\Delta s_a)_i)(y_i + (\Delta y_a)_i) = (\Delta s_a)_i(\Delta y_a)_i \neq 0$$

Complementarity violation
depends on step length

Corrected direction

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

$$\Delta S_a = \mathbf{diag}(\Delta s_a)$$
$$\Delta Y_a = \mathbf{diag}(\Delta y_a)$$

Mehrotra predictor-corrector algorithm

Initialization

Given (x, s, y) such that $s, y > 0$

1. **Termination conditions**

   $$r_p = Ax + s - b, \quad r_d = A^T y + c, \quad \mu = (s^T y)/m$$

   If $\|r_p\|, \|r_d\|, \mu$ are small, **break** Optimal solution $(x^\star, s^\star, y^\star)$

2. **Newton step** (affine scaling) $\sigma = 0$

   $$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y_a \\ \Delta x_a \\ \Delta s_a \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} \end{bmatrix}$$

Mehrotra predictor-corrector algorithm

3. **Barrier parameter**

$$\alpha_p = \max\{\alpha \in [0,1] \mid s + \alpha \Delta s_a \geq 0\}$$

$$\alpha_d = \max\{\alpha \in [0,1] \mid y + \alpha \Delta y_a \geq 0\}$$

$$\mu_a = \frac{(s + \alpha_p \Delta s_a)^T (y + \alpha_d \Delta y_a)}{m}$$

// NEW CANDIDATE DUALITY MEASURE

$$\sigma = \left(\frac{\mu_a}{\mu}\right)^3$$

4. **Corrected direction**

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

Mehrotra predictor-corrector algorithm

5. **Update iterates**

$$\alpha_p = \max\{\alpha \geq 0 \mid s + \alpha\Delta s \geq 0\}$$
$$\alpha_d = \max\{\alpha \geq 0 \mid y + \alpha\Delta y \geq 0\}$$

$$(x, s) = (x, s) + \min\{1, \eta\alpha_p\}(\Delta x, \Delta s)$$
$$y = y + \min\{1, \eta\alpha_d\}\Delta y$$

Avoid corners

$$\eta = 1 - \epsilon \approx 0.99$$

Implementation and linear algebra

Search equations

Step 2 (**Newton**) and 4 (**Corrected direction**) solve equations of the form

$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} b_y \\ b_x \\ b_s \end{bmatrix}$$

The **Newton** step right hand side: $$\begin{bmatrix} b_y \\ b_x \\ b_s \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} \end{bmatrix}$$

The **corrector** step right hand side: $$\begin{bmatrix} b_y \\ b_x \\ b_s \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} - \Delta S_a \Delta Y_a \mathbf{1} + \sigma\mu\mathbf{1} \end{bmatrix}$$

Solving the search equations

Our linear system is not symmetric

$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} b_y \\ b_x \\ b_s \end{bmatrix}$$

Solving the search equations

Our linear system is not symmetric

$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} b_y \\ b_x \\ b_s \end{bmatrix}$$

$A\Delta x + \Delta s = b_y$

$S\Delta y + Y\Delta s = b_s$

Substitute last equation, $\Delta s = Y^{-1}(b_s - S\Delta y)$, into first

$$\begin{bmatrix} -Y^{-1}S & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \end{bmatrix} = \begin{bmatrix} b_y - Y^{-1}b_s \\ b_x \end{bmatrix}$$

Solving the search equations

Our reduced system is symmetric but not positive definite

Solving the search equations

Our reduced system is symmetric but not positive definite

$$\begin{bmatrix} -Y^{-1}S & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \end{bmatrix} = \begin{bmatrix} b_y - Y^{-1}b_s \\ b_x \end{bmatrix}$$

$-Y^{-1}S\Delta y + A\Delta x = b_y - Y^{-1}b_s$

Solving the search equations

Our reduced system is symmetric but not positive definite

$$\begin{bmatrix} -Y^{-1}S & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \end{bmatrix} = \begin{bmatrix} b_y - Y^{-1}b_s \\ b_x \end{bmatrix}$$

$-Y^{-1}S\Delta y + A\Delta x = b_y - Y^{-1}b_s$

Substitute first equation, $\Delta y = S^{-1}Y(A\Delta x - b_y + Y^{-1}b_s)$, into second

$$A^T S^{-1} Y A \Delta x = b_x + A^T S^{-1} Y b_y - A^T S^{-1} b_s$$

easy to compute

Reduced linear system

Coefficient matrix

$$B = A^T S^{-1} Y A$$

Characteristics

- A is **large** and **sparse**
- $S^{-1}Y$ is **positive** and **diagonal**, different at each iteration
- B is **positive definite** if $\text{rank}(A) = n$
- Sparsity pattern of B is the **pattern** of $A^T A$ (independent of $S^{-1}Y$)

Reduced linear system

Coefficient matrix

$$B = A^T S^{-1} Y A$$

Cholesky factorizations

$$B = PLL^T P^T$$

- Reordering only once to get P
- One numerical factorizaton per interior-point iteration $O(n^3)$
- Forward/backward substitution twice per iteration $O(n^2)$

Reduced linear system

Coefficient matrix

$$B = A^T S^{-1} Y A$$

Cholesky factorizations

$$B = PLL^T P^T$$

- Reordering only once to get P
- One numerical factorizaton per interior-point iteration $O(n^3)$ ⟶ **Per-iteration complexity** $O(n^3)$
- Forward/backward substitution twice per iteration $O(n^2)$

Convergence

Mehrotra's algorithm

No convergence theory → Examples where it **diverges** (rare!)

Convergence

Mehrotra's algorithm

No convergence theory → Examples where it **diverges** (rare!)

Fantastic convergence **in practice** → Less than 30 iterations

Convergence

Mehrotra's algorithm

No convergence theory → Examples where it **diverges** (rare!)

Fantastic convergence **in practice** → Less than 30 iterations

Theoretical iteration complexity

Alternative versions (slower than Mehrotra) converge in $O(\sqrt{n})$ iterations

Convergence

Mehrotra's algorithm

No convergence theory → Examples where it **diverges** (rare!)

Fantastic convergence **in practice** → Less than 30 iterations

Theoretical iteration complexity

Alternative versions (slower than Mehrotra) converge in $O(\sqrt{n})$ iterations

Average iteration complexity

Average iterations complexity is $O(\log n)$

Convergence

Mehrotra's algorithm

No convergence theory → Examples where it **diverges** (rare!)

Fantastic convergence **in practice** → Less than 30 iterations

Floating point operations

Theoretical iteration complexity

Alternative versions (slower than Mehrotra) converge in $O(\sqrt{n})$ iterations → $O(n^{3.5})$

Average iteration complexity

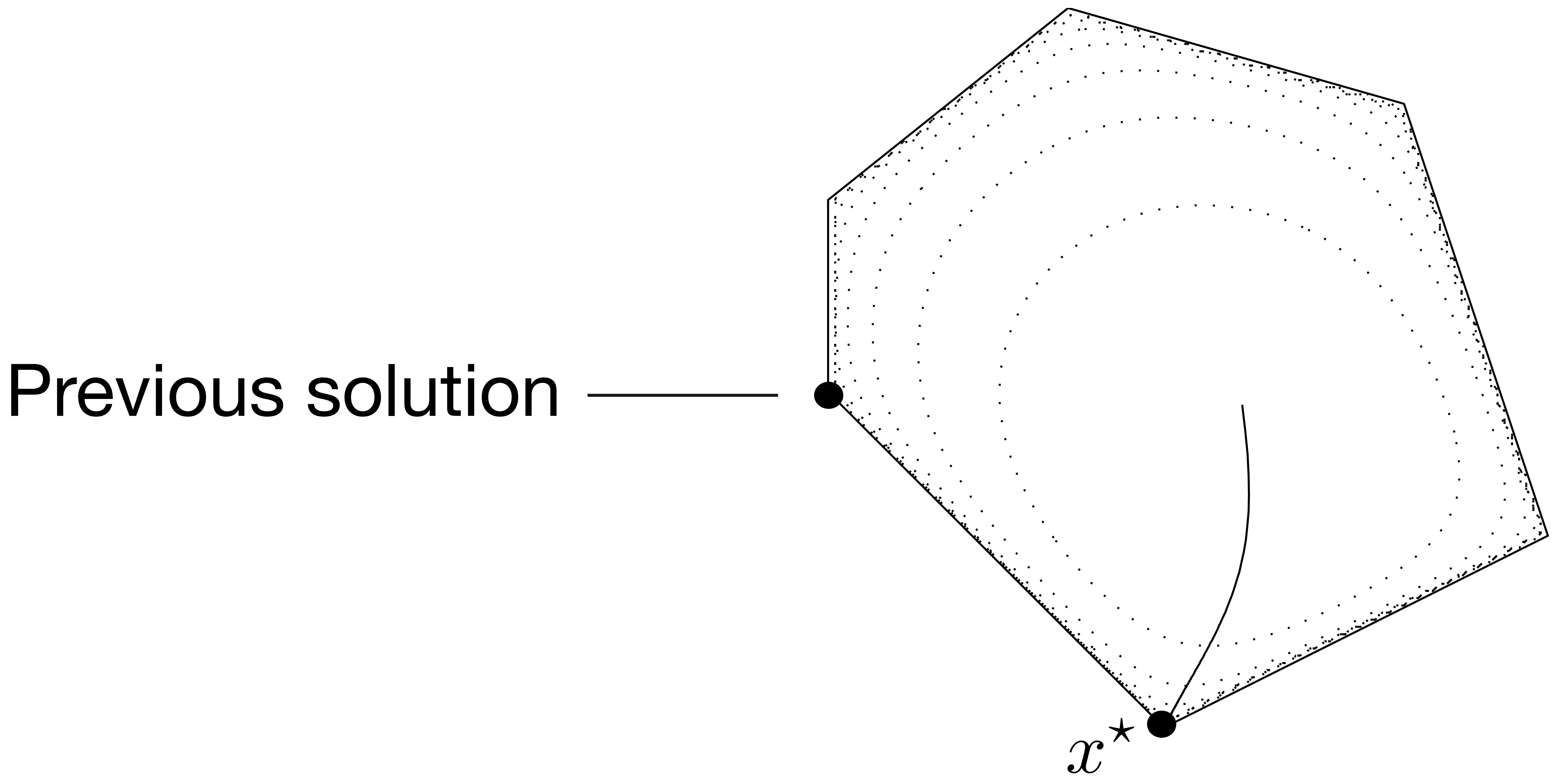
Average iterations complexity is $O(\log n)$ → $O(n^3 \log n)$

Warm-starting

Interior-point methods are **difficult to warm-start**

Warm-starting

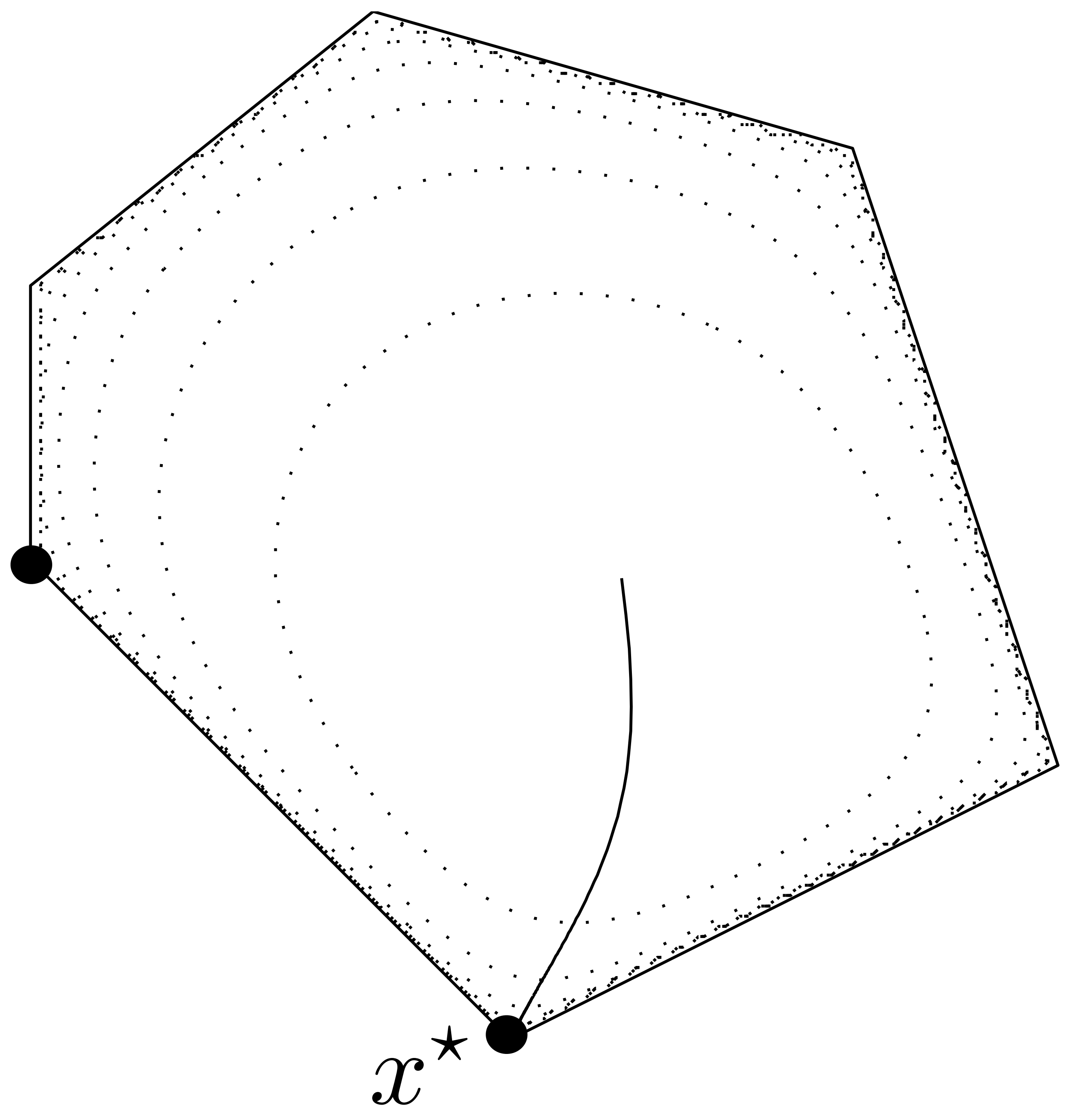
Interior-point methods are **difficult to warm-start**



Warm-starting

Interior-point methods are **difficult to warm-start**

Previous solution



Badly centered
initial point

↓

Hard to make progress
with long steps

Interior-point vs simplex

Example

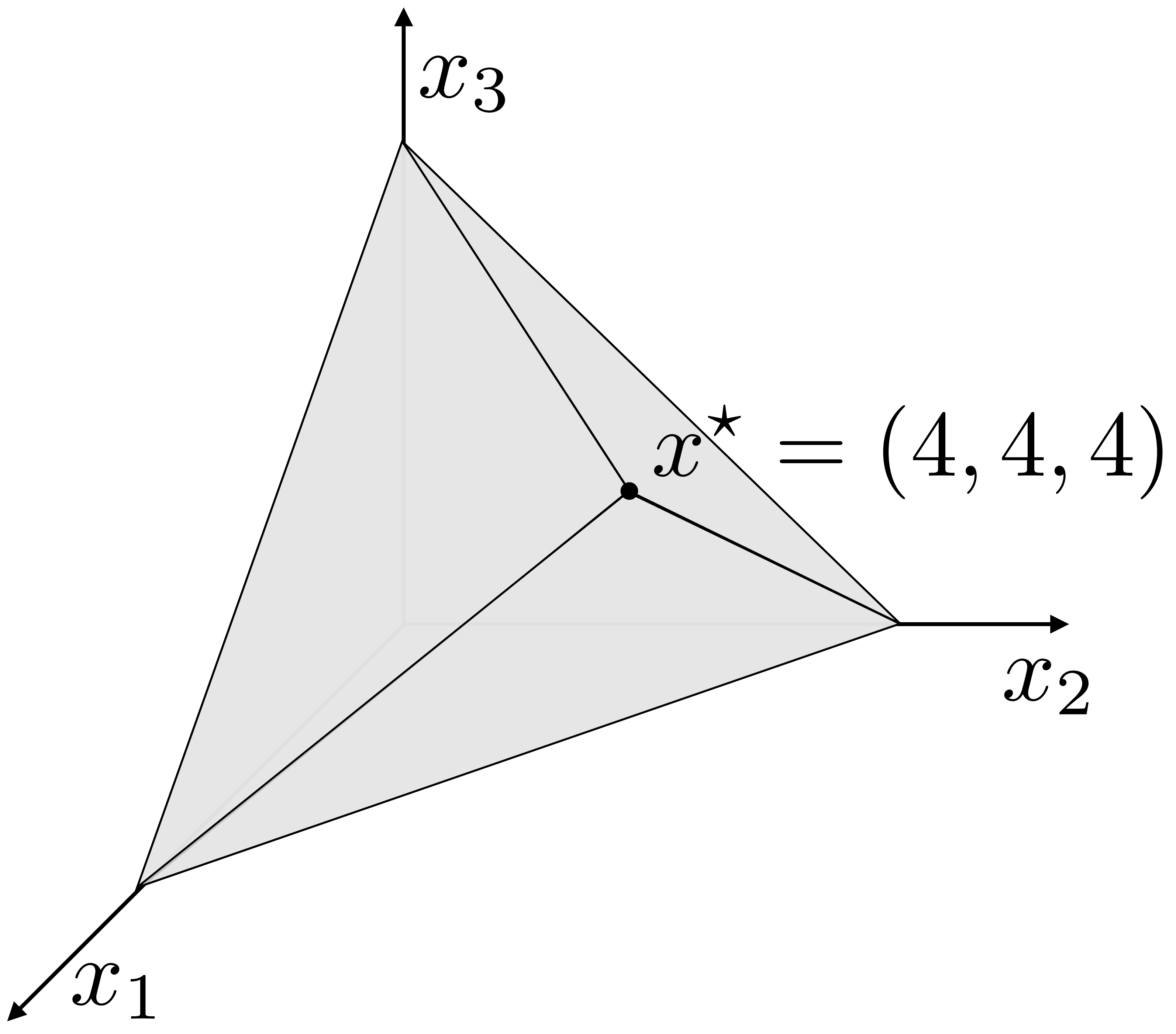
minimize $\quad -10x_1 - 12x_2 - 12x_3$

subject to $\quad x_1 + 2x_2 + 2x_3 \leq 20$

$$2x_1 + x_2 + x_3 \leq 20$$

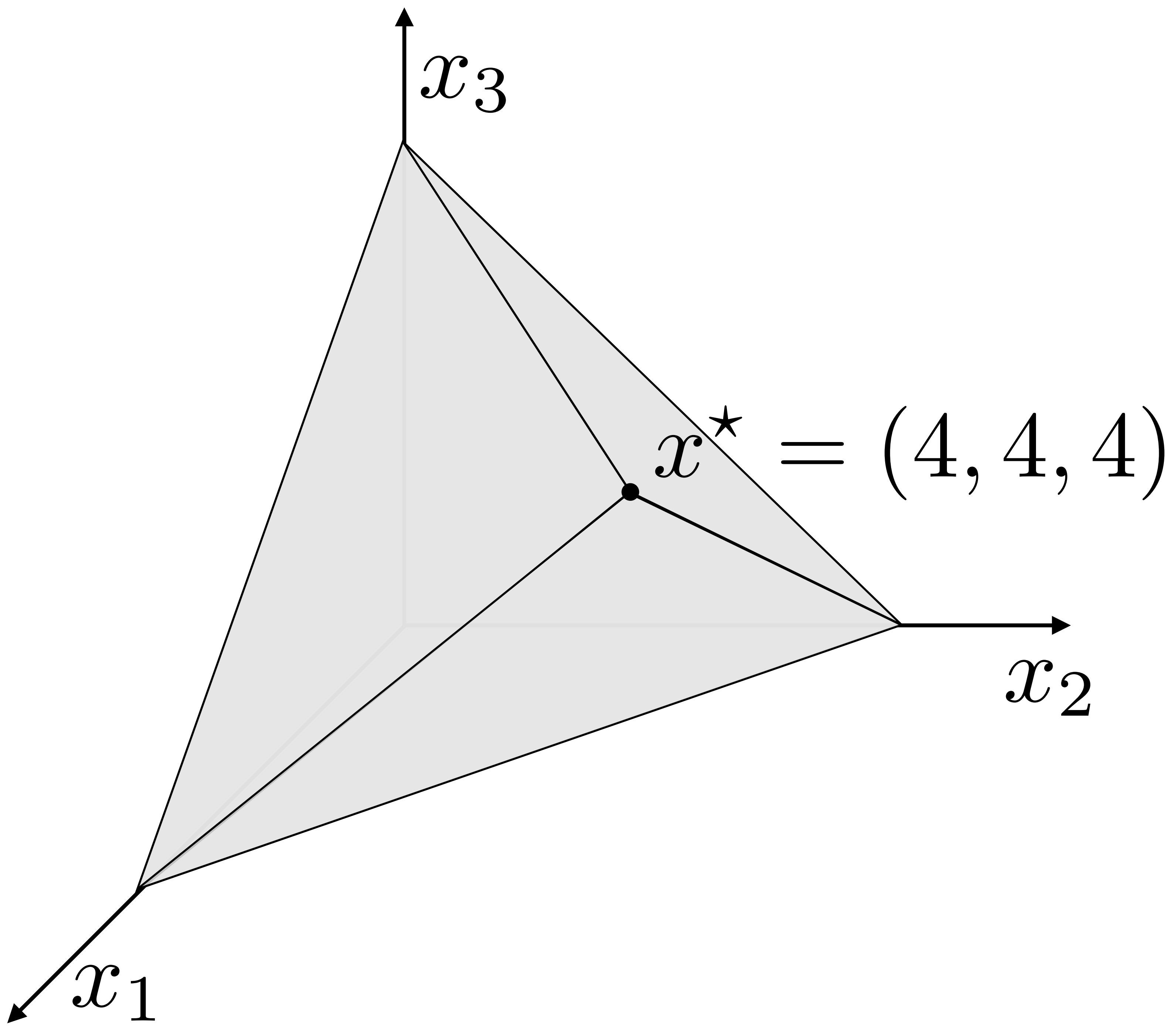
$$2x_1 + 2x_2 + x_3 \leq 20$$

$$x_1, x_2, x_3 \geq 0$$



Example

$$\begin{array}{ll} \text{minimize} & -10x_1 - 12x_2 - 12x_3 \\ \text{subject to} & x_1 + 2x_2 + 2x_3 \leq 20 \\ & 2x_1 + x_2 + x_3 \leq 20 \\ & 2x_1 + 2x_2 + x_3 \leq 20 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$



$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \\ & x \geq 0 \end{array}$$

$$c = (-10, -12, -12)$$

$$A = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}$$

$$b = (20, 20, 20)$$

- GLPK SIMPLEX
- CVXOPT INTERIOR-POINT

Example with real solver

CVXOPT (open-source)

Code

```python
import numpy as np
import cvxpy as cp

c = np.array([-10, -12, -12])
A = np.array([[1, 2, 2],
              [2, 1, 2],
              [2, 2, 1]])
b = np.array([20, 20, 20])
n = len(c)

x = cp.Variable(n)
problem = cp.Problem(cp.Minimize(c @ x),
                     [A @ x <= b, x >= 0])
problem.solve(solver=cp.CVXOPT, verbose=True)
```

Output

```
     pcost       dcost       gap    pres   dres   k/t
 0: -1.3077e+02 -2.3692e+02  2e+01  1e-16  6e-01  1e+00
 1: -1.3522e+02 -1.4089e+02  1e+00  2e-16  3e-02  4e-02
 2: -1.3599e+02 -1.3605e+02  1e-02  2e-16  3e-04  4e-04
 3: -1.3600e+02 -1.3600e+02  1e-04  1e-16  3e-06  4e-06
 4: -1.3600e+02 -1.3600e+02  1e-06  1e-16  3e-08  4e-08
Optimal solution found.
```

Solution

```
In [3]: x.value
Out[3]: array([3.99999999, 4.        , 4.        ])
```

[The CVXOPT linear and quadratic cone program solvers, L. Vandenberghe 2010]

Average interior-point complexity

Random LPs

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

n variables
$3n$ constraints

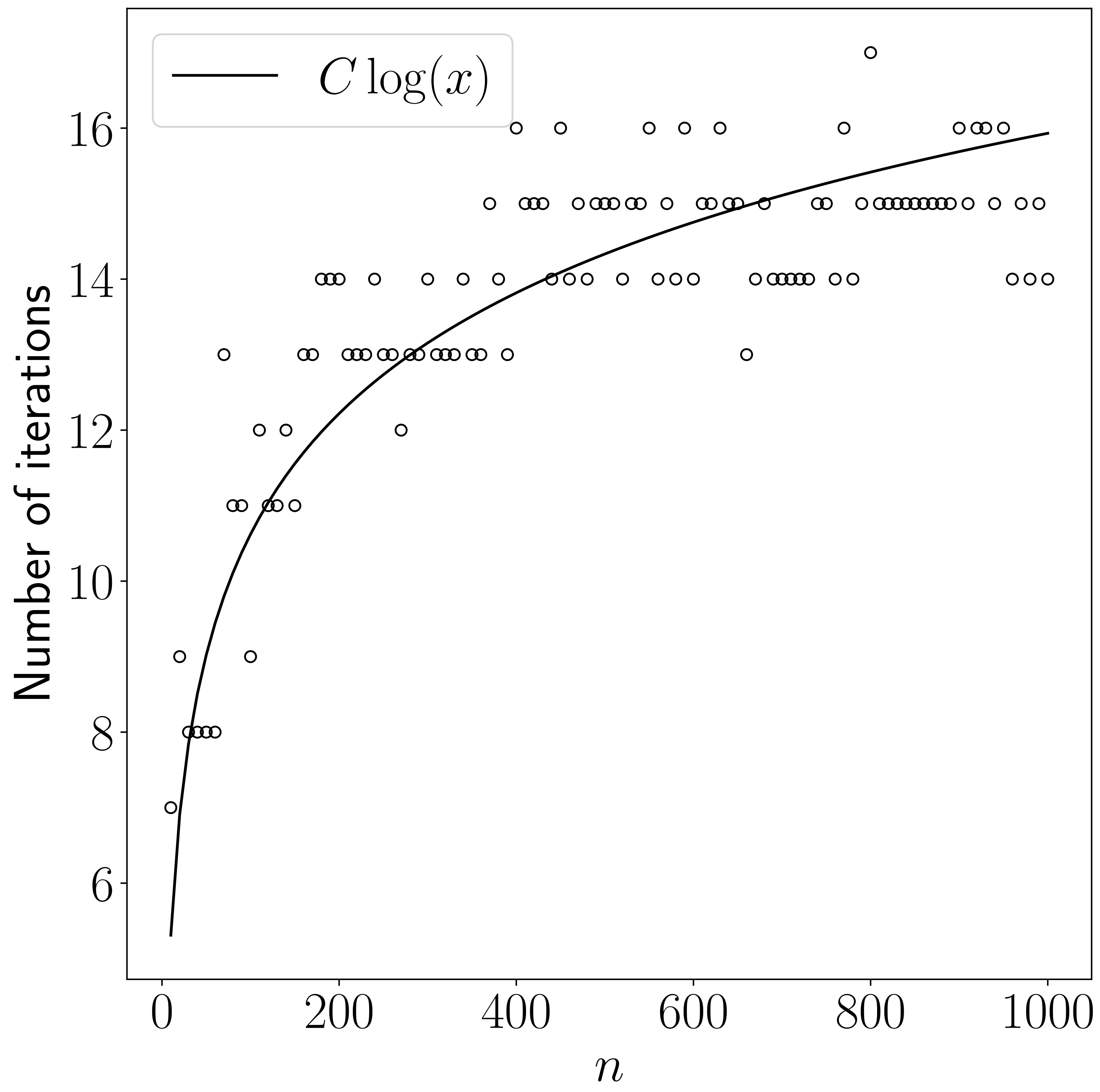
Average interior-point complexity

Random LPs

minimize $c^T x$
subject to $Ax \leq b$

n variables
$3n$ constraints

Iterations: $O(\log n)$



Average interior-point complexity

Random LPs

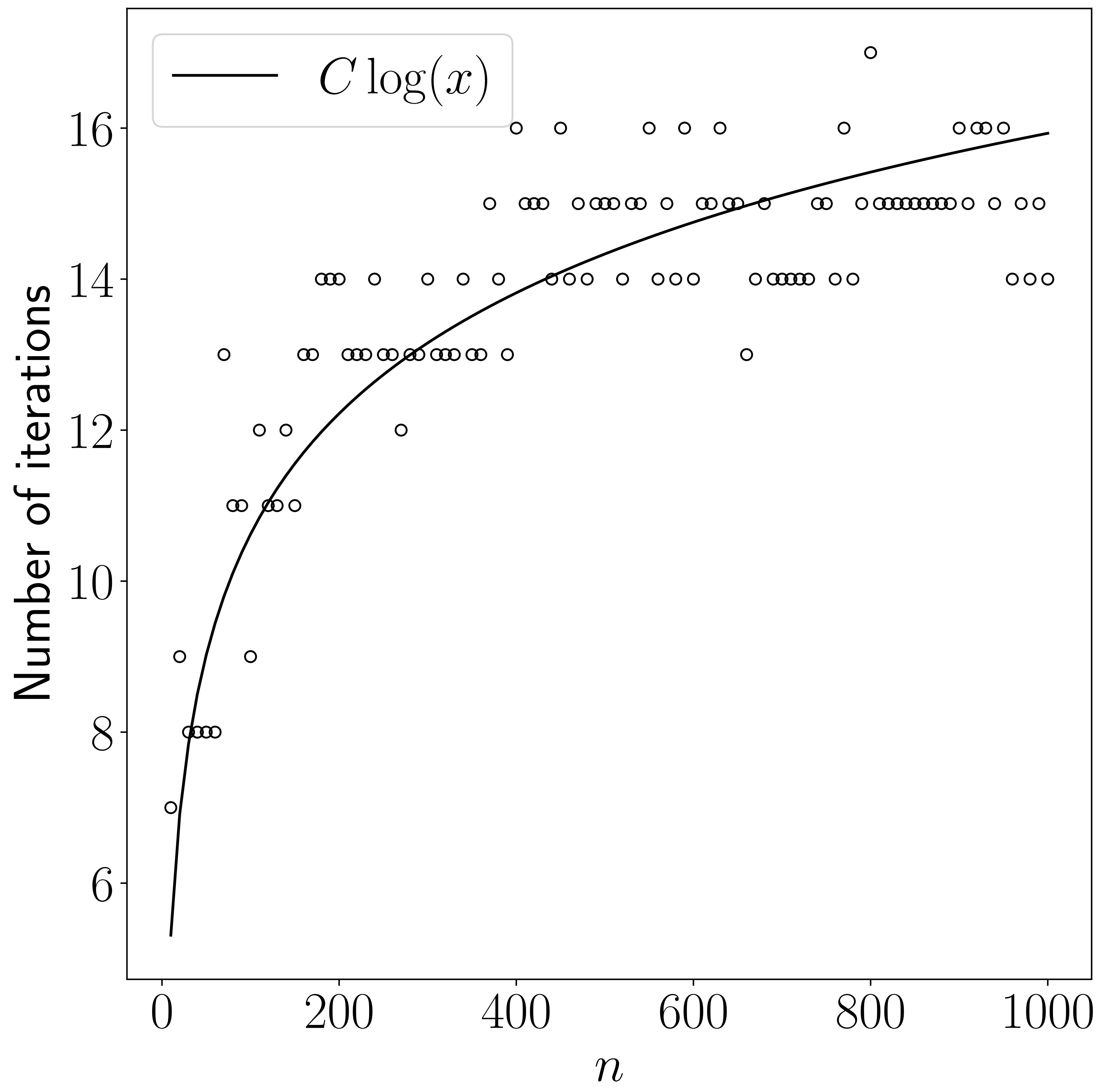
minimize $\quad c^T x$

subject to $\quad Ax \leq b$

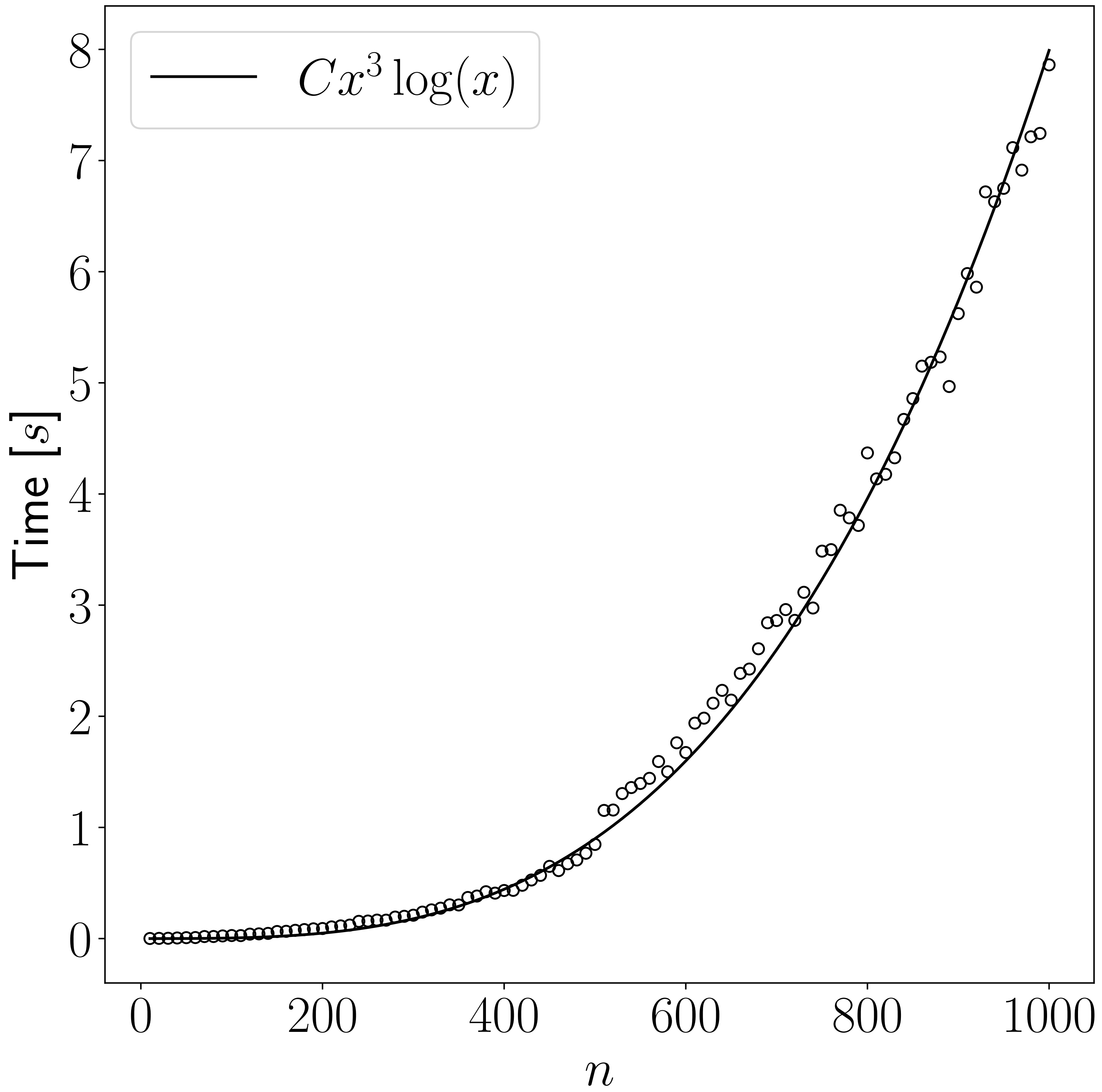
n variables

$3n$ constraints

Iterations: $O(\log n)$



Time: $O(n^3 \log n)$



Comparison between interior-point method and simplex

Primal simplex

- Primal feasibility
- Zero duality gap

↓

Dual feasibility

Dual simplex

- Dual feasibility
- Zero duality gap

↓

Primal feasibility

Primal-dual interior-point

- Interior condition

↓

- Primal feasibility
- Dual feasibility
- Zero duality gap

Comparison between interior-point method and simplex

Primal simplex

- Primal feasibility
- Zero duality gap

↓

Dual feasibility

Dual simplex

- Dual feasibility
- Zero duality gap

↓

Primal feasibility

Exponential worst-case complexity

Primal-dual interior-point

- Interior condition

↓

- Primal feasibility
- Dual feasibility
- Zero duality gap

Polynomial worst-case complexity

Comparison between interior-point method and simplex

Primal simplex

- Primal feasibility
- Zero duality gap

↓

Dual feasibility

Dual simplex

- Dual feasibility
- Zero duality gap

↓

Primal feasibility

Primal-dual interior-point

- Interior condition

↓

- Primal feasibility
- Dual feasibility
- Zero duality gap

Exponential worst-case complexity

Requires feasible point

Polynomial worst-case complexity

Allows infeasible start

Comparison between interior-point method and simplex

Primal simplex

- Primal feasibility
- Zero duality gap

↓

Dual feasibility

Dual simplex

- Dual feasibility
- Zero duality gap

↓

Primal feasibility

Primal-dual interior-point

- Interior condition

↓

- Primal feasibility
- Dual feasibility
- Zero duality gap

Simplex	Primal-dual interior-point
Exponential worst-case complexity	**Polynomial worst-case complexity**
Requires feasible point	**Allows infeasible start**
Can be warm-started	**Cannot be warm-started**

Which algorithm should I use?

Dual simplex

- Small-to-medium problems
- Repeated solves with varying data

Interior-point (barrier)

- Medium-to-large problems
- Sparse structured problems

Which algorithm should I use?

Dual simplex

- Small-to-medium problems
- Repeated solves with varying data

Interior-point (barrier)

- Medium-to-large problems
- Sparse structured problems

How do solvers with multiple options decide?

Concurrent Optimization

Which algorithm should I use?

Dual simplex

- Small-to-medium problems
- Repeated solves with varying data

Interior-point (barrier)

- Medium-to-large problems
- Sparse structured problems

How do solvers with multiple options decide?

<u>Concurrent Optimization</u>

Why not both? (crossover)

Interior-point ⟶ Few simplex steps

Interior-point methods implementation

Today, we learned to:

- **Apply** Mehrotra predictor-corrector algorithm
- **Exploit** linear algebra to speedup computations
- **Analyze** empirical complexity
- **Compare** interior-point and simplex methods

References

- D. Bertsimas and J. Tsitsiklis: Introduction to Linear Optimization
  - Chapter 9.4 — 9.6: Interior point methods

- R. Vanderbei: Linear Programming
  - Chapter 17: The Central Path
  - Chapter 15: A Path-Following Method

Next lecture

- Overview for linear optimization