Linear Programming: Interior-Point Methods

Note Title 2/26/2022

One big question remains : How many simplex steps do we have to take?

In practice : $O(\max(\#\text{ of vars}, \#\text{ of constraints}))$

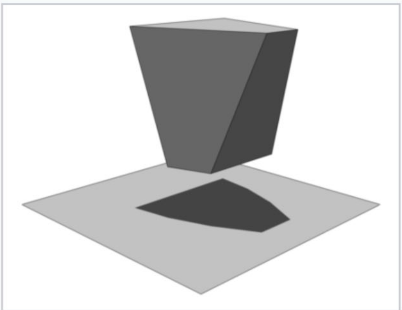
worst case : exponential time

Klee-Minty (1973) : $\exists$ an n-d LP whose polytope has 2^n vertices, for which the simplex method visits every vertex before reaching the optimal point!

Klee–Minty cube

From Wikipedia, the free encyclopedia

The **Klee–Minty cube** or **Klee–Minty polytope** (named after Victor Klee and George J. Minty) is a unit hypercube of variable dimension whose corners have been perturbed. Klee and Minty demonstrated that George Dantzig's simplex algorithm has poor worst-case performance when initialized at one corner of their "squashed cube". On the three-dimensional version, the simplex algorithm and the criss-cross algorithm visit all 8 corners in the worst case.



Klee Minty cube for shadow vertex simplex method.

In particular, many optimization algorithms for linear optimization exhibit poor performance when applied to the Klee–Minty cube. In 1973 Klee and Minty showed that Dantzig's simplex algorithm was not a polynomial-time algorithm when applied to their cube.[1] Later, modifications of the Klee–Minty cube have shown poor behavior both for other basis-exchange pivoting algorithms and also for interior-point algorithms.[2]

Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time

Spielman - Teng
2004

Authors: Daniel A. Spielman, Shang-Hua Teng Authors Info & Claims

Journal of the ACM, Volume 51, Issue 3 • May 2004 • pp 385–463 • https://doi.org/10.1145/990308.990310

Online: 01 May 2004 Publication History

Abstract

We introduce the *smoothed analysis of algorithms*, which continuously interpolates between the worst-case and average-case analyses of algorithms. In smoothed analysis, we measure the maximum over inputs of the expected performance of an algorithm under small random perturbations of that input. We measure this performance in terms of both the input size and the magnitude of the perturbations. We show that the simplex algorithm has *smoothed complexity* polynomial in the input size and the standard deviation of Gaussian perturbations.

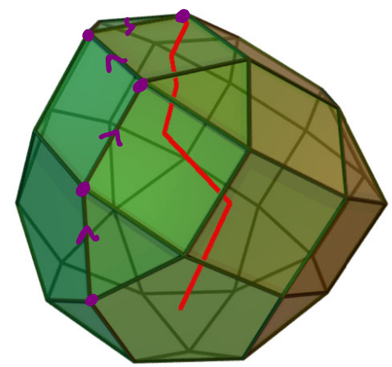
Late 70's : Khachiyan developed a provably polynomial time algorithm for LP called the ellipsoid method.

But it performs poorly in practice.

Mid-80's : Karmarkar described a polynomial time algorithm that approaches the solution through the interior of the feasible polytope.

COMBINATORICA 4 (4) (1984) 373—395

— simplex
— interior point



A NEW POLYNOMIAL-TIME ALGORITHM FOR LINEAR PROGRAMMING

N. KARMARKAR

Received 20 August 1984
Revised 9 November 1984

We present a new polynomial-time algorithm for linear programming. In the worst case, the algorithm requires $O(n^{3.5}L)$ arithmetic operations on $O(L)$ bit numbers, where n is the number of variables and L is the number of bits in the input. The running-time of this algorithm is better than the ellipsoid algorithm by a factor of $O(n^{2.5})$. We prove that given a polytope P and a strictly interior point $\mathbf{a} \in P$, there is a projective transformation of the space that maps P, $\mathbf{a}$ to P', $\mathbf{a}'$ having the following property. The ratio of the radius of the smallest sphere with center $\mathbf{a}'$, containing P' to the radius of the largest sphere with center $\mathbf{a}'$ contained in P' is $O(n)$. The algorithm consists of repeated application of such projective transformations each followed by optimization over an inscribed sphere to create a sequence of points which converges to the optimal solution in polynomial time.

Karmarkar's work sparked intense research in the field of interior point methods. We study a specific interior point method, different from Karmarkar's original method and is used in the current generation of software, called

primal-dual interior point method.

Given (P) $\min c^T x$ st $Ax = b$, $x \geq 0$, its dual is (D) $\max b^T \lambda$ st $A^T\lambda + s = c$, $s \geq 0$.

KKT conditions (necc. and suff. for optimality):

① $A^T\lambda + s = c$

② $Ax = b$

③ $x_i s_i = 0$, $i = 1, \cdots, n$,

④ $x \geq 0$, $s \geq 0$.

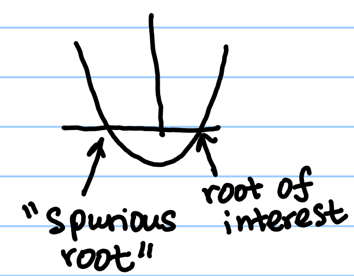
$n + m + n = 2n+m$ variables (x, λ, s)

in

$n + m + n = 2n+m$ equations (①, ②, ③)

① linear, ② linear, ③ quadratic!

The situation is like we have a quadratic equation to solve, but we are only interested in the non-negative roots.



The primal-dual interior point method finds a solution (x^*, λ^*, s^*) by

- applying a variant of Newton's method to the (mildly) nonlinear $(2n+m)$ by $(2n+m)$ system of equations,
- while maintaining the iterates (x^k, λ^k, s^k) to satisfy $\underbrace{x^k > 0,\ s^k > 0}_{\text{"interior point"}}$ (in order to avoid spurious solutions.)

If you don't know what Newton's method is, it's only because you don't realize that you know:

Given $F: \mathbb{R}^N \to \mathbb{R}^N$, x^0 an initial guess of the solution of $F(x)=0$.

$$\underset{\substack{\uparrow \\ \text{nonlinear}}}{F(x)} \approx F(x^0) + \underset{\substack{\uparrow \\ \text{linear}}}{[DF(x^0)](x-x^0)}, \quad x \approx x^0, \qquad DF(x^0) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_N} \\ \vdots & & \vdots \\ \frac{\partial F_N}{\partial x_1} & \cdots & \frac{\partial F_N}{\partial x_N} \end{bmatrix}\Bigg|_{x^0}$$

We don't quite know how to solve the nonlinear system $F(x)=0$,

but we know how to solve the linear system $F(x^0) + [DF(x^0)](x-x^0) = 0$

$$\Downarrow$$

$$x = x^0 - [DF(x^0)]^{-1} F(x^0)$$

call the solution x^1, and iterate.

Newton's method for solving nonlinear equations:	For $k = 0, 1, \cdots,$ compute $DF(x^k)$ Solve $[DF(x^k)]\Delta x = -F(x^k)$ $x^{k+1} = x^k + \Delta x$

damped Newton's method
$x^{k+1} = x^k + \alpha \Delta x \quad (\alpha \in [0,1])$

It doesn't always work, but when it works it works like charm (quadratic convergence)

The nonlinear system we need to solve here is very specific:

$$F(x,\lambda,s) = \begin{bmatrix} A^T\lambda + s - c \\ Ax - b \\ XSe \end{bmatrix} = 0 \qquad (\text{with } x \geq 0,\ s \geq 0)$$

$X = \operatorname{diag}(x_1, \cdots, x_n)$,
$S = \operatorname{diag}(s_1, \cdots, s_n)$, $\quad e = [1, \cdots, 1]^T$.

$$DF(x,\lambda,s) = \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix}$$

the derivatives are constant

the derivatives vary with x and s

A Newton iteration solves the following linear system:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = - \begin{bmatrix} A^T\lambda^k + s^k - c \\ A x^k - b \\ X^k S^k e \end{bmatrix} \begin{matrix} =: r_c^k \\ =: r_b^k \\ \end{matrix}$$

$$\begin{bmatrix} x^{k+1} \\ \lambda^{k+1} \\ s^{k+1} \end{bmatrix} = \begin{bmatrix} x^k \\ \lambda^k \\ s^k \end{bmatrix} + \alpha \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix}.$$

Experience shows that α typically has to be very small before violating $x \geq 0$, $s \geq 0$. This leads to very slow convergence.

It was discovered that the following modification works well.

At the kth iterate, think of applying one Newton step to solve

$A^T\lambda + s - c = 0$, $\cancel{x_i s_i = 0}$ $\quad x_i s_i = \sigma_k \left(\frac{1}{n}\sum_{i=1}^{n} x_i^k s_i^k\right)$

$Ax - b = 0$,

called the "duality measure"

a reduction factor called the "centering parameter", $\sigma_k \in [0,1]$

The Newton equation becomes:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c^k \\ -r_b^k \\ -X^k S^k e + \sigma_k \mu_k e \end{bmatrix}$$

At optimality, the duality measure $\mu := \frac{1}{n}\sum_{i=1}^{n} x_i s_i$ satisfies $\mu = 0$.

(Technically, this isn't a Newton's method anymore.)

Everything on this page seems to come out of the blue, or too ad hoc.

We are about to have a much better picture of what's going on.

Ex: Prove that the duality gap $c^Tx - b^T\lambda = x^Ts$, justifying the name "duality measure" above.

Framework 14.1 (Primal-Dual Path-Following).

Given (x^0, λ^0, s^0) with $(x^0, s^0) > 0$;

$Ax^0 = b$

$A^T\lambda^0 + s^0 = c$

for $k = 0, 1, 2, \ldots$

Choose $\sigma_k \in [0, 1]$ and solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta \lambda^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} -r_c^k = 0 \\ -r_b^k = 0 \\ -X^k S^k e + \sigma_k \mu_k e \end{bmatrix},$$

where $\mu_k = (x^k)^T s^k / n$;

Set

$$(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k, \lambda^k, s^k) + \alpha_k(\Delta x^k, \Delta \lambda^k, \Delta s^k),$$

choosing α_k so that $(x^{k+1}, s^{k+1}) > 0$.

end (for).

Note: If $Ax^0 \neq b$

$A^T\lambda^0 + s^0 \neq c$

then, in general,

all x^k are not primal feasible,

all λ^k, s^k are not dual feasible.

However, if $Ax^0 = b$

$A^T\lambda^0 + s^0 = c$

then

all x^k are primal feasible,

all λ^k, s^k are dual feasible,

$r_c^k = 0$

$r_b^k = 0 \quad \forall k.$

Moreover,

$A^T \Delta\lambda^k + \Delta s^k = 0$

$A \Delta x^k = 0,$

so

$\Delta s^k \in \text{Image}(A^T)$

$\Delta x^k \in \text{Null}(A)$

$\Rightarrow \quad \Delta x^k \perp \Delta s^k.$

The central path

Define $\mathcal{F} := \{(x,\lambda,s) \mid Ax=b,\ A^T\lambda+s=c,\ x,s\geq 0\}$ ← the primal-dual feasible set

$\mathcal{F}^0 := \{(x,\lambda,s) \mid Ax=b,\ A^T\lambda+s=c,\ x,s>0\}$ ← the primal-dual strictly feasible set

The idea of solving the primal dual pair is to solve:

$$(\star)\begin{cases} A^T\lambda+s=c \\ Ax=b \\ x_is_i=\tau, \quad i=1,\cdots,n \\ x,s\geq 0 \end{cases}$$

with a choice of $\tau>0$, and gradually shrink τ towards 0.

When τ is too small, the solution of $(\star)$ is too close to the boundary of $\mathcal{F}$, and it is too hard for the damped Newton's method to make progress. (The $x,s\geq 0$ constraints are the culprits here.)

Notice an elegant property:

$$\left.\begin{aligned} A^T\lambda+s&=c \\ Ax&=b \\ x_is_i&=0, \quad i=1,\cdots,n \\ x,s&\geq 0 \end{aligned}\right\} \text{KKT conds. of } \begin{aligned} &\min c^Tx \\ &\text{st } Ax=b \\ &\qquad x\geq 0 \end{aligned}, \qquad \left.\begin{aligned} A^T\lambda+s&=c \\ Ax&=b \\ x_is_i&=\tau, \quad i=1,\cdots,n \\ x,s&>0 \end{aligned}\right\} \text{KKT conds of } \boxed{\begin{aligned} &\min c^Tx-\tau\sum_1^n \ln(x_i) \\ &\text{St. } Ax=b \end{aligned}}\ x>0$$

called the log-barrier formulation of the LP

Check : $\mathcal{L}(x,\lambda) = c^T x - \tau \sum \ln(x_i) - \lambda^T(Ax-b)$. $\nabla_x \mathcal{L}(x,\lambda) = c - \tau \underbrace{\begin{bmatrix} x_1^{-1} \\ \vdots \\ x_n^{-1} \end{bmatrix}}_{s=} - A^T\lambda = 0$

Note :

- in the log-barrier formulation, we may dispense with the $x \geq 0$ constraint, because $\ln(x_i) \to -\infty$ if $x_i \to 0^+$.

- $\nabla_x \left(c^T x - \tau \sum \ln(x_i) \right) = c - \begin{bmatrix} \tau/x_1 \\ \vdots \\ \tau/x_n \end{bmatrix}$, $\nabla_x^2 \left(c^T x - \tau \sum \ln(x_i) \right) = \tau \begin{bmatrix} x_1^{-2} & & \\ & \ddots & \\ & & x_n^{-2} \end{bmatrix} \succ 0$.

  This means the objective is strictly convex.

  So if the log-barrier formulation is feasible, then its solution is unique.

This also means that the solution of $\begin{cases} A^T\lambda + s = c \\ Ax = b \\ x_i s_i = \tau, \; i=1,\cdots,n \\ x, s > 0 \end{cases}$, when exists, is unique.

defined earlier $\rightarrow F(x,\lambda,s) = \begin{bmatrix} 0 \\ 0 \\ \tau e \end{bmatrix} \Leftrightarrow$ (the system above)

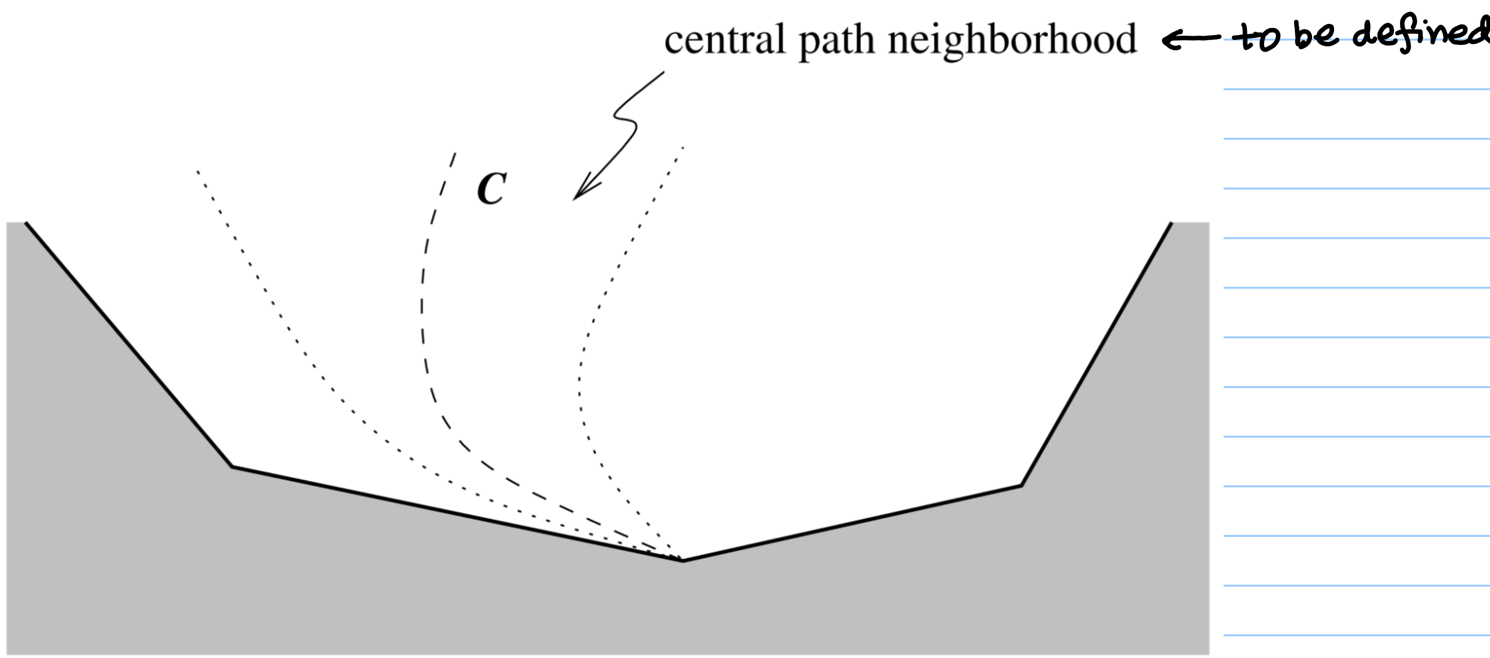
Proposition : the solution of $F(x,\lambda,s) = \begin{bmatrix} 0 \\ 0 \\ \tau e \end{bmatrix}$ exists for any $\tau > 0 \iff \mathcal{F}^0 \neq \emptyset$.

(Proof omitted.) ($\Rightarrow$ is obvious)

We define the central path (of $\mathcal{F}$, when $\mathcal{F}^o \neq \emptyset$) as:

$$\boxed{\mathcal{C} := \left\{ (x_\tau, \lambda_\tau, s_\tau) : \tau > 0,\ F(x_\tau, \lambda_\tau, s_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau e \end{bmatrix} \right\}.}$$

If $\mathcal{C}$ converges to anything as $\tau \downarrow 0$, it must converge to a primal-dual solution of the LP, and it does so in a way that stays away from the boundary of $\mathcal{F}$.



So let's rethink what the "Newton step" below tries to achieve:

$$(*) \quad \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c^k = 0 \\ -r_b^k = 0 \\ -X^k S^k e + \sigma_k \mu_k e \end{bmatrix}$$

Assume $(x^k, \lambda^k, s^k) \in \mathcal{F}$

$\mu_k = \frac{1}{n} \sum_{i=1}^{n} x_i^k s_i^k$

- If $\sigma_k = 1$, it is a bona-fide Newton step of the nonlinear system

$$A^T \lambda + s - c = 0, \quad Ax - b = 0, \quad x_i s_i = \sigma_k \mu_k, \quad i = 1, \dots, n.$$

meaning that it tries to move (x^k, λ^k, s^k) to $(x^{k+1}, \lambda^{k+1}, s^{k+1})$ with the latter point closer to $(x_{\mu_k}, \lambda_{\mu_k}, s_{\mu_k}) \in \mathcal{C}$.

Is it a good move or not?

the idea of 'centering' $\longrightarrow$

> Well, not from the point of view of reducing the duality measure μ.
> BUT, by moving closer to $\mathcal{C}$ — which typically means moving more towards the interior of the positive orthant — it sets the scene for a substantial reduction in μ on the next iteration.

- If $\sigma_k = 0$, $(*)$ is just the original Newton step. It directly drives μ towards 0,

but likely pushing x^k to the boundary of the (primal) feasible region, causing difficiulties in subsequent steps.
The idea is to use intermediate values of σ from $(0,1)$ to trade off the twin goals of reducing μ and improving centrality.

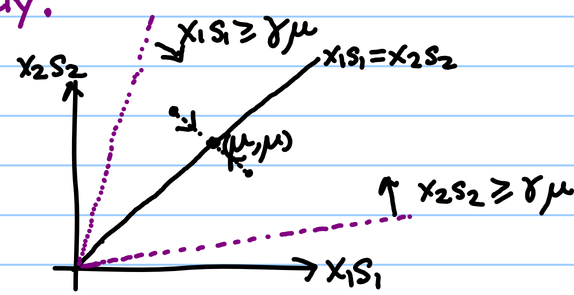
To develop an algorithm with a provably good convergence property (to be formulated), we impose the iterates to stay in

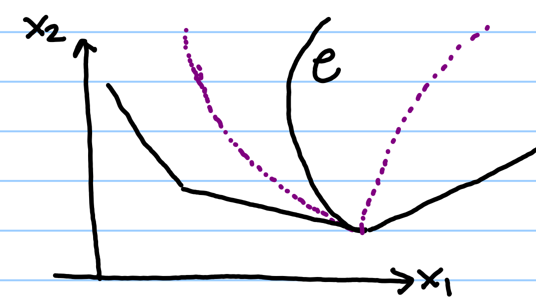
$$\mathcal{N}_{-\infty}(\gamma) := \{ (x,\lambda,s) \in \mathcal{F}^0 \mid x_i s_i \geq \gamma \underbrace{\frac{1}{n}\sum_{j=1}^{n} x_j s_j}_{\mu(x,s)} , \ i=1,\cdots,n \}$$

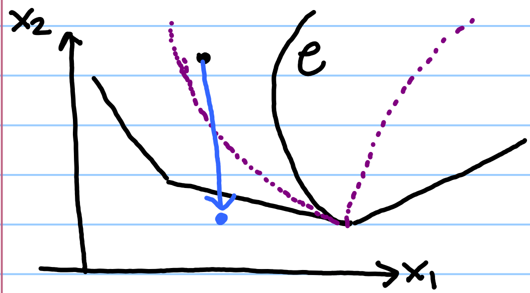
for some $\gamma \in [0,1]$. Typically $\gamma = 10^{-3}$.

Of course, $\mathcal{C} \subset \mathcal{N}_{-\infty}(\gamma)$ and $\mathcal{N}_{-\infty}(\gamma)$ can be thought of as a neighborhood of $\mathcal{C}$.

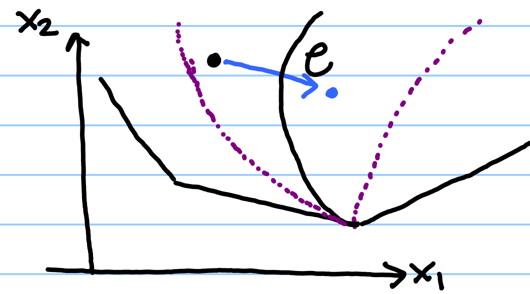
Imposing $(x^k, \lambda^k, s^k) \in \mathcal{N}_{-\infty}(\gamma)$ means the iterates, in some sense, stay away from the boundary of $\mathcal{F}$ uniformly.



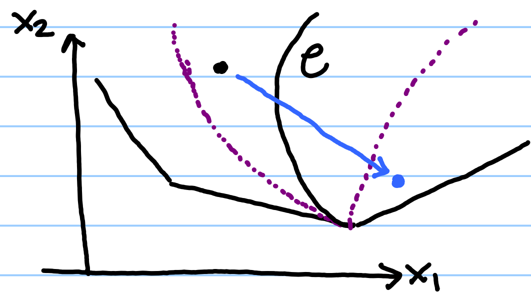




$\sigma_k = 0$
pure Newton



$\sigma_k = 1$



$0 < \sigma_k < 1$

The damping parameter α_k comes to rescue if the "Newton step" takes us too close to the boundary of $\mathcal{F}$.

Here and in later analysis, we use the notation

$$(x^k(\alpha), \lambda^k(\alpha), s^k(\alpha)) \stackrel{\text{def}}{=} (x^k, \lambda^k, s^k) + \alpha(\Delta x^k, \Delta\lambda^k, \Delta s^k),$$
$$\mu_k(\alpha) \stackrel{\text{def}}{=} x^k(\alpha)^T s^k(\alpha)/n.$$

Algorithm 14.2 (Long-Step Path-Following).

Given $\gamma, \sigma_{\min}, \sigma_{\max}$ with $\gamma \in (0, 1)$, $0 < \sigma_{\min} \leq \sigma_{\max} < 1$,
 and $(x^0, \lambda^0, s^0) \in \mathcal{N}_{-\infty}(\gamma)$;
for $k = 0, 1, 2, \ldots$
 Choose $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$;
 Solve (14.10) to obtain $(\Delta x^k, \Delta\lambda^k, \Delta s^k)$;
 Choose α_k as the largest value of α in $[0, 1]$ such that

$$(x^k(\alpha), \lambda^k(\alpha), s^k(\alpha)) \in \mathcal{N}_{-\infty}(\gamma);$$

 Set $(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k(\alpha_k), \lambda^k(\alpha_k), s^k(\alpha_k))$;
end (for).

(14.10)

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta\lambda^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} -r_c^k = 0 \\ -r_b^k = 0 \\ -X^k S^k e + \sigma_k \mu_k e \end{bmatrix}$$

where $\mu_k = (x^k)^T s^k / n$;

The algorithm may seem ad hoc, but we are about to see that its design guarantees that it takes

$$O(n\,|\log\epsilon|) \text{ iterations}$$

to reduce the duality measure by a factor of ϵ, i.e., to identify a point (x^k, λ^k, s^k) for which

$$\mu_k \le \epsilon\,\mu_0.$$

If you have seen the quadratic convergence analysis of Newton's method (to be presented in Math 671), you may realize that the $O(n|\log\epsilon|)$ estimate seems too slow. But note that the method here is not really a pure Newton's method.

Yet, I do not know how tight the $O(n|\log\epsilon|)$ bound is. And it is said that in practice the empirical rate of convergence seems to go faster than $O(n|\log\epsilon|)$.

Regarding "polynomial time algorithm": Assuming infinite precision $(+,-,*,\div)$ operations

- simplex method gives an exact solution in a finite number of operations, but may take unbearably long.

- Primal-dual interior point method basically never gives an exact solution in a finite number of operations. But will always produce an "ϵ-accurate" approximate solution in $O(n^4 \,|\log \epsilon|)$ operations.

$O(n^3)$ time for each Newton step

Now, let's indulge in the first (and only) rate of convergence proof in this course.

Theorem 14.3.

Given the parameters γ, $\sigma_{\min}$, and $\sigma_{\max}$ in Algorithm 14.2, there is a constant δ independent of n such that

← Tricky

$$\mu_{k+1} \leq \left(1 - \frac{\delta}{n}\right) \mu_k, \tag{14.25}$$

for all $k \geq 0$.

$$\Downarrow \text{ easy}$$

Theorem 14.4. (Main result)

Given $\epsilon \in (0,1)$ and $\gamma \in (0,1)$, suppose the starting point in Algorithm 14.2 satisfies $(x^0, \lambda^0, s^0) \in \mathcal{N}_{-\infty}(\gamma)$. Then there is an index K with $K = O(n \log 1/\epsilon)$ such that

$$\mu_k \le \epsilon \mu_0, \qquad \text{for all } k \ge K.$$

Idea of the proof of Theorem 14.3 :

$$\begin{aligned}
\mu_{k+1} &= (x^k + \alpha_k \Delta x^k)^T (s^k + \alpha_k \Delta s^k)/n \\
&= \mu_k + \alpha_k ((s^k)^T \Delta x^k + (x^k)^T \Delta s^k)/n + \alpha_k^2 \underbrace{(\Delta x^k)^T \Delta s^k}_{=0}/n \\
&= \mu_k + \alpha_k (-\mu_k + \sigma_k \mu_k) \\
&= \mu_k [1 - \alpha_k (1 - \sigma_k)]
\end{aligned}$$

if we can guarantee that the step size α_k stays away from 0 uniformly (in k), then we are in business.

Recall:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -X^k S^k e + \sigma_k \mu_k e \end{bmatrix}$$

First two block rows $\Rightarrow (\Delta x^k)^T \Delta s^k = 0$

Last block row

$$\Rightarrow S^k \Delta x^k + X^k \Delta s^k = -X^k S^k e + \sigma_k \mu_k e$$

e^T

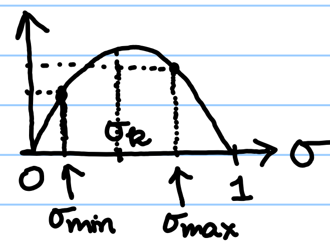
$$\Rightarrow (s^k)^T \Delta x^k + (x^k)^T \Delta s^k = -(x^k)^T s^k + \sigma_k \mu_k n$$

And the whole point of the "long-step path following" algorithm is to make it happen.

We shall prove $\alpha_k \geq 2^{3/2} \frac{\sigma_k}{n} \gamma \frac{1-\gamma}{1+\gamma}$, ——— (lower bound for step size)

from this we conclude the proof of Thm 14.3:

$$\mu_{k+1} \leq \mu_k \left[1 - \frac{2^{3/2}}{n} \gamma \frac{1-\gamma}{1+\gamma} \underbrace{\sigma_k (1-\sigma_k)}_{\min(\sigma_{min}(1-\sigma_{min}),\ \sigma_{max}(1-\sigma_{max}))} \right]$$



So we just have to choose

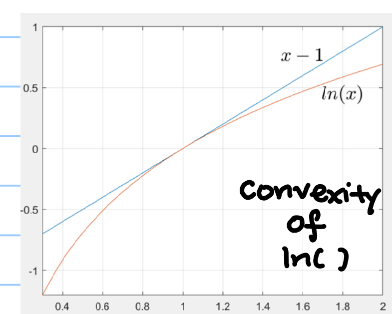
$$\delta = 2^{3/2} \gamma \frac{1-\gamma}{1+\gamma} \min\{\sigma_{min}(1-\sigma_{min}),\ \sigma_{max}(1-\sigma_{max})\}.$$ ■

Proof of Thm 14.3 $\Rightarrow$ Thm 14.4 :

$$\mu_{k+1} \leq (1-\delta/n)\mu_k \Rightarrow \ln \mu_k \leq \ln(1-\delta/n) + \ln \mu_k \quad \forall k$$

Iterate this inequality : $\ln \mu_k \leq k \underbrace{\ln(1-\delta/n)}_{\leq -\delta/n \ (\text{if } \delta/n < 1)} + \ln \mu_0$

$$\ln(\mu_k/\mu_0) \leq k(-\delta/n)$$



$\mu_k/\mu_0 \leq \epsilon \iff \ln(\mu_k/\mu_0) \leq \ln(\epsilon)$, and it happens when $k(-\delta/n) \leq \ln\epsilon$, or

$$k \geq \frac{n}{\delta}\ln\frac{1}{\epsilon} = \frac{n}{\delta}|\ln\epsilon| =: K. \qquad \blacksquare$$

It remains to prove the lower bound of step size.

Q: Given $(x^k, \lambda^k, s^k) \in \mathcal{N}_{-\infty}(\gamma) = \{\underbrace{(x,\lambda,s) \in \mathcal{F}}_{\text{feasibility}} : \underbrace{x_i s_i \geq \gamma \overset{=\frac{1}{n}\sum x_j s_j}{\mu},\ i=1,\cdots,n}_{\text{centrality}}\}$,

how big can α be in order for $(x^k, \lambda^k, s^k) + \alpha(\Delta x^k, \Delta\lambda^k, \Delta s^k) \in \mathcal{N}_{-\infty}(\gamma)$?

Feasibility is guaranteed by the method, so it boils down to analyzing the condition:

$$(x_i^k + \alpha\Delta x_i^k)(s_i^k + \alpha\Delta s_i^k) \underset{??}{\geq} \gamma \underbrace{\frac{1}{n}\sum_{j=1}^{n}(x_j^k + \alpha\Delta x_j^k)(s_j^k + \alpha\Delta s_j^k)}_{=:\mu_k(\alpha)}$$

Note: If $\alpha=0$, $LHS_\alpha \geq RHS_\alpha$. The question is how big we can set α and still guarantees $LHS_\alpha \geq RHS_\alpha$.

Recall: The 3rd block row of the Newton eqt:

$$S^k\Delta x^k + X^k\Delta s^k = -X^k S^k e + \sigma_k\mu_k e \quad \text{—}(\text{III})$$

Also: $\Delta x^k \perp \Delta s^k$.

$$LHS_\alpha = (\overset{=x_i^k(\alpha)}{x_i^k + \alpha\Delta x_i^k})(\overset{=s_i^k(\alpha)}{s_i^k + \alpha\Delta s_i^k}) = x_i^k s_i^k + \alpha\underbrace{(x_i^k\Delta s_i^k + s_i^k\Delta x_i^k)}_{= -x_i^k s_i^k + \sigma_k\mu_k \text{ (III used componentwise)}} + \alpha^2\Delta x_i^k\Delta s_i^k$$

$$= x_i^k s_i^k(1-\alpha) + \alpha\sigma_k\mu_k + \alpha^2\Delta x_i^k\Delta s_i^k$$

$$\geq \gamma(1-\alpha)\mu_k + \alpha\sigma_k\mu_k - \alpha^2|\Delta x_i^k\Delta s_i^k|$$

$$\gamma^{-1}RHS_\alpha = \mu_k(\alpha) = \mu_k + \alpha\underbrace{\frac{1}{n}\sum_{j=1}^n x_j^k\Delta s_j^k + s_j^k\Delta x_j^k}_{} + \frac{1}{n}\alpha^2\underbrace{\sum_{j=1}^n \Delta x_j^k\Delta s_j^k}_{(\Delta x^k)^T\Delta s^k = 0}$$

$$= \frac{1}{n}e^T(S^k\Delta x^k + X^k\Delta s^k)$$

$$= \frac{1}{n}e^T(-X^kS^ke + \sigma_k\mu_k e) \leftarrow \text{(III)}$$

$$= -\mu_k + \sigma_k\mu_k$$

so $\mu_k(\alpha) = \mu_k(1-\alpha(1-\sigma_k))$, $RHS_\alpha = \gamma\mu_k(1-\alpha(1-\sigma_k))$

If we can obtain an upper bound for $|\Delta x_i^k\Delta s_i^k|$ of the form $(*)\mu_k$, then "$LHS_\alpha \geq RHS_\alpha$" is implied by $(\text{—})\mu_k \geq \gamma(1-\alpha(1-\sigma_k))\mu_k$, and there is a chance to see how big we can set the step size and still guarantee $LHS_\alpha \geq RHS_\alpha$. Also, $|\Delta x_i^k\Delta s_i^k|$ is relatively small, it suffices to bound it crudely.

Claim: $|\Delta x_i^k\Delta s_i^k| \leq 2^{-3/2}(1+\gamma^{-1})n\mu_k$

With this upper bound, the lower bound for step size can be obtained:

$$LHS_\alpha \geq RHS_\alpha \Leftarrow \gamma(1-\alpha)\mu_k + \alpha\sigma_k\mu_k - \alpha^2|\Delta x_i^k \Delta s_i^k| \geq \mu_k\gamma(1-\alpha(1-\sigma_k))$$

$$\Leftarrow \gamma(1-\alpha)\mu_k + \alpha\sigma_k\mu_k - \alpha^2 2^{-3/2}(1+\gamma^{-1})n\mu_k \geq \mu_k\gamma(1-\alpha(1-\sigma_k))$$

$$\Leftrightarrow \gamma(1-\alpha) + \alpha\sigma_k - \alpha^2 2^{-3/2}(1+\gamma^{-1})n \geq \gamma(1-\alpha(1-\sigma_k))$$

$$= \gamma(1-\alpha) + \gamma\alpha\sigma_k$$

$$\Leftrightarrow (1-\gamma)\sigma_k\alpha \geq 2^{-3/2}(1+\gamma^{-1})n\alpha^2$$

$$\Leftrightarrow \alpha \leq 2^{3/2}\frac{\sigma_k}{n}\gamma\frac{1-\gamma}{1+\gamma}$$

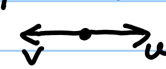
← this gives a lower bound, not upper bound, for α_k! Because we choose α_k to be the <u>largest α</u> that guarantees $LHS_\alpha \geq RHS_\alpha$.

Proof of $|\Delta x_i^k \Delta s_i^k| \leq 2^{-3/2}(1+\gamma^{-1})n\mu_k$:

Lemma : If $u, v \in \mathbb{R}^n$ with $u^T v \geq 0$, then $|u_i v_i| \leq 2^{-3/2}\|u+v\|_2^2$.

[See N&W, 2nd edition, pg 402.]

Note : it clearly would not work if $u = -v$



For the rest of this proof, let's drop the superscripts and subscripts 'k' for convenience.

We again make use of $\Delta x^T \Delta s = 0$ and $S\Delta x + X\Delta s = -XSe + \sigma\mu e$. — (III)

By the lemma, we can choose $u = D^{-1}\Delta x$, $v = D\Delta s$ ($u^T v = 0$ for any D)

$$|\Delta x_i \Delta s_i| \underset{\text{lemma}}{\le} 2^{-3/2}\|D^{-1}\Delta x + D\Delta s\|_2^2 \underset{\substack{\text{choose}\\ D = X^{1/2}S^{-1/2}}}{=} 2^{-3/2}\|X^{-1/2}S^{-1/2}\underbrace{(S\Delta x + X\Delta s)}_{-XSe+\sigma\mu e \text{ by (III)}}\|_2^2$$

$$= 2^{-3/2}\|-X^{\frac{1}{2}}S^{1/2}e + \sigma\mu X^{-\frac{1}{2}}S^{-\frac{1}{2}}e\|_2^2$$

$\|y\|_2^2 = y^T y$

$$\overset{\|y\|_2^2 = y^Ty}{=} 2^{-3/2}\Big[\underset{\|}{x^T s} - 2\sigma\mu\underbrace{e^T e}_{=n} + \sigma^2\mu^2\sum_{i=1}^{n}(x_i s_i)^{-1}\Big]$$

$$\le 2^{-3/2}\Big[n\mu - 2\sigma\mu n + \sigma^2\mu^2\frac{n}{\gamma\mu}\Big] \quad \text{since } x_i s_i \ge \gamma\mu$$

$$\le 2^{-3/2}\left[1 - 2\sigma + \sigma^2/\gamma\right] n\mu$$

$$\le 2^{-3/2}\left[1 + \gamma^{-1}\right] n\mu, \quad \text{as claimed.} \qquad \blacksquare$$

pretty crude... and it suffices for bounding the "most negligible" term