CS 435, 2018
Lecture 7, Date: 19 April 2018
Instructor: Nisheeth Vishnoi

Primal Path Following Interior Point Methods

In this lecture, we build on Newton's method and its convergence presented in the previous lecture to derive a polynomial time algorithm for Linear Programming. Key to the reduction from constrained to unconstrained optimization is the notion of barrier functions and the corresponding central path.

Contents

1 Linear Programming

Linear programming is the task of finding the minimum value of a linear function over a set of linear constraints, more formally, it is the following optimization problem.

> **Linear Programming.**
>
> *Input:* A matrix $A \in \mathbb{Q}^{m \times n}$ and vectors $b \in \mathbb{Q}^m$ and $c \in \mathbb{Q}^n$.
>
> *Goal:* Find $x^\star \in \operatorname{argmin}\left\{c^\top x : Ax \leq b\right\}$.

The ideas introduced in Lectures 3,4 and 5 (gradient descent, mirror descent and the concept of acceleration) can be applied to this problem. However, the running times of the resulting algorithms depends inverse polynomially on error parameter ε. For instance, the method based on the MWU scheme that we introduced for the bipartite matching problem, can be applied to general linear programs and yields an algorithm that given $\varepsilon > 0$ computes a solution x which has value at most $c^\top x^\star + \varepsilon$ (for an optimal solution $x^\star$) and violates all the constraints by at most ε (additively), in time proportional to $\frac{1}{\varepsilon^2}$. Not only the dependency on ε is unsatisfactory, but also, when no additional assumptions on A, b or c are made, these methods run in time exponential in the bit complexity[1] parameter L. As discussed in Lecture 2, for a method to be regarded as polynomial time, we require the dependency on the error parameter $\varepsilon > 0$ in the running time to be polynomial in $\log \frac{1}{\varepsilon}$, moreover the dependency on L should be also polynomial. In this lecture we derive a method which satisfies both these requirements for linear programming.

Theorem 1 (Polynomial time algorithm for solving LPs)**.** *There is an algorithm which given a description of a linear program (A, b, c) (with n variables, m constraints and bit complexity L) and an $\varepsilon > 0$ either outputs a feasible solution $\hat{x}$ with $c^\top \hat{x} \leq c^\top x^\star + \varepsilon$ or correctly decides that the program is infeasible in* $\operatorname{poly}\left(L, \log \frac{1}{\varepsilon}\right)$ *time.*

With additional work one can transform the above into an exact algorithm for Linear Programming – one that outputs an optimal point $x^\star$. This is achieved by picking $\varepsilon > 0$ small enough and rounding the output $\hat{x}$ to $x^\star$ (which is guaranteed to be rational and have bit complexity bounded by $O(L)$) – see [2] for details. This is where the poly-logarithmic dependency on $1/\varepsilon$ is crucial.

2 From Constrained to Unconstrained Optimization via Barrier Functions

So far we have largely discussed methods for unconstrained optimization problems. In this section we present a methodology to reduce constrained optimization problems to unconstrained optimization problems – via *barrier functions*.

[1]Recall that L is the number of bits required to specify all the rational numbers in the input. In particular $L \geq \max(n, m)$ where n is the number of variables and m is the number of constraints.

We consider constrained convex optimization problems of the form

$$\inf_{x \in K} f(x) \tag{1}$$

where f is a convex, real valued function and $K \subseteq \mathbb{R}^n$ is a convex set. To simplify our discussion, we assume that the objective function is linear, i.e., $f(x) = c^\top x$ and the convex body K is bounded and full-dimensional (it has positive volume).

Suppose we have a point $x_0 \in K$ and we want to perform an improvement step (with respect to the objective $c^\top x$) maintaining the condition of being inside K. The simplest idea would be to keep moving in the direction $-c$ to decrease our objective value as much as possible. Our step will then end up on the boundary of K. The second and subsequent iterates would then lie on to the boundary, which could force our steps to be short and potentially make this method inefficient. Indeed, such a method (when applied to polytopes) is equivalent to a variant of the "Simplex Method" [**?**] that is known to have an exponential worst case running time [**?**]. [2]

The key idea is to move the constraints captured by the convex set K to the objective function and consider

$$\inf_{x \in \mathbb{R}^n} c^\top x + F(x), \tag{2}$$

where $F(x)$ can be regarded as a "fee" for violating constraints – thus $F(x)$ should become big for x close to ∂K. One seemingly perfect choice of F would be a function which is 0 on K and $+\infty$ on the complement of K. This reduces our problem to unconstrained minimization of $c^\top x + F(x)$. However, note that we have not gained anything by this reformulation, even worse: the objective is not continuos anymore.

If one would like the methods for unconstrained optimization, that we developed in the previous lectures, to be applicable here, then F should satisfy certain properties such as convexity etc. To formalize this approach we introduce the notion of a *Barrier Function*. Instead of giving a precise definition, we list some properties, that is expected for a barrier function F:

- F is defined in the interior of K, i.e., $\mathrm{dom}(F) = \mathrm{int}(K)$,
- for every point $q \in \partial K$ it holds that: $\lim_{x \to q} F(x) = +\infty$,
- F is strictly convex.

Suppose F is such a barrier function. Note that solving (2) might give us some idea of what $\min_{x \in K} c^\top x$ is, but will certainly not give us the precise answer, since the presence of $F(x)$ alters the location of the minimum. For this reason one typically considers a whole family of perturbed objective functions f_η parametrized by $\eta > 0$

$$f_\eta(x) = \eta c^\top x + F(x) \tag{3}$$

[2] Spielman and Teng [**?**] showed that under certain assumptions on the data (A, b, c) a variant of the Simplex Method provably works.

We may imagine that f_η is defined on all of $\mathbb{R}^n$ but attains finite values only on $\operatorname{int}(K)$. Intuitively, making η bigger and bigger reduces the influence of $F(x)$ on the optimal value of $f_\eta(x)$. Let us jump into a more concrete setting: Linear Programming, where we have already seen a barrier function.

3 The Logarithmic Barrier Function for Polytopes and the Central Path

The linear programming introduced earlier can be cast as the problem of solving $\min_{x \in P} c^\top x$, where P is the following polytope

$$P := \{x \in \mathbb{R}^n : \langle a_i, x \rangle \le b_i, \text{ for } i = 1, 2, \dots, n\}, \tag{4}$$

and $a_1, a_2, \dots, a_m$ are the rows of A (treated as column vectors). We assume that P is a bounded polytope of nonzero volume[3] in $\mathbb{R}^n$ and that the considered linear program has a unique optimal solution $x^\star$. To implement the idea sketched in the previous section we use the following *Logarithmic Barrier* function.

Definition 2 (Logarithmic Barrier)**.** *For a matrix $A \in \mathbb{R}^{m \times n}$ (with rows $a_1, a_2, \dots, a_m \in \mathbb{R}^n$) and a vector $b \in \mathbb{R}^m$ we define the logarithmic barrier function $F : \operatorname{int}(P) \to \mathbb{R}$ as*

$$F(x) := -\sum_{i=1}^{m} \log(b_i - a_i^\top x).$$

As observed in Lecture 6, $F(x)$ is well defined and strictly convex on $\operatorname{int}(P)$ and tends to infinity when approaching the boundary of P. Given such a barrier function, let us go back to the parametrized family of perturbed objectives $\{f_\eta\}_{\eta \ge 0}$ introduced in (3). Observe that since $c^\top x$ is a linear function, the second order behavior of f_η is completely determined by F, that is $\nabla^2 f_\eta = \nabla^2 F$. In particular, f_η is strictly convex and has a unique minimizer. This motivates the following notion of a central path.

Definition 3 (Central Path)**.** *For every $\eta \ge 0$ we denote by $x_\eta^\star$ the unique minimizer of $f_\eta(x)$ over $x \in \operatorname{int}(P)$. The set of all these minimizers is called the Central Path (with respect to c) and is denoted by*

$$\Gamma_c := \{x_\eta^\star : \eta \ge 0\}.$$

The central path Γ_c can be seen to be continuous due to the Implicit Function Theorem. It originates at the analytic center $x_0^\star$ of P and approaches $x^\star$ – the optimal solution to the linear program when $\eta \to \infty$. In other words:

$$\lim_{\eta \to \infty} x_\eta^\star = x^\star.$$

The idea is now to initialize our algorithm at some point, say $x_1^\star$ (i.e., for $\eta = 1$) on the

[3]Of course this is not always the case for general linear programs, however if the program is feasible then we can perturb the constraints by an exponentially small $\varepsilon > 0$ to force the feasible set to be full dimensional.

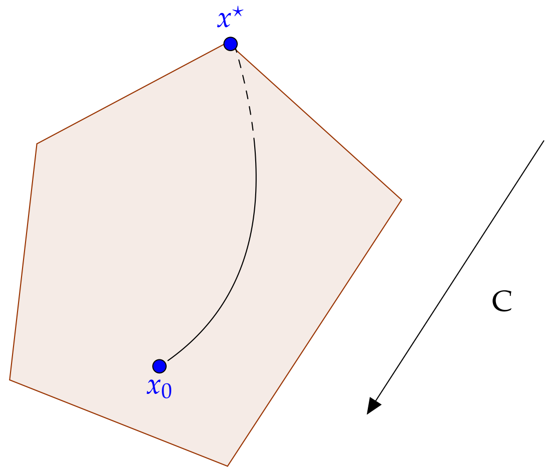

Figure 1: Example of a central path

central path, and progress along the path by gradually increasing η to infinity. A method that follows this general approach is referred to as a *path-following interior point method.* Now, the key questions from the algorithmic perspective are

1. How to find an initial point on the central path (the analytic centre)?
2. How to follow the central path using discrete steps?
3. How to decide when to stop following the central path (the goal is reach a close enough neighborhood of $x^\star$)?

We make this method precise in the next section and answer all these questions in our analysis of the path following IPM in Section 5.

4 A Path Following Algorithm for Linear Programming

4.1 The Algorithm

While the initialization and termination questions stated as 1. and 3. in the list at the end of the previous section are certainly important, we first focus on 2. – the way we progress along the path.

Suppose for a moment that we are given a point $x_0 \in P$ that is close to the central path, i.e., to $x^\star_{\eta_0}$ for some $\eta_0 > 0$. Then since (as we show later) the function f_{η_0} satisfies the **NL** condition from the previous lecture (Definition 5 in the notes for Lecture 6), we know that by performing a Newton step

$$x_1 := x_0 + n_{\eta_0}(x_0),$$

(where for any x and $\eta > 0$ we define $n_\eta(x) := -\left(\nabla^2 f_\eta(x)\right)^{-1} \nabla f_\eta(x)$), we make significant progress towards $x^\star_{\eta_0}$. The main idea is to use this progress and the opportunity it presents to increase the value of η_0 to $\eta_1 := \eta_0 \cdot (1+\delta)$ for some $\delta > 0$ so that x_1 is close enough to $x^\star_{\eta_1}$ and satisfies the **NL** condition; so that we can repeat this procedure and continue.

The key question becomes: how large a δ can we pick in order to make this scheme work and produce a sequence of pairs $(x_0, \eta_0), (x_1, \eta_1), \ldots$ such that η_k increases at a rate $(1+\delta)$ and that x_k is in the quadratic convergence region of f_{η_k}. We show in Section 5 that the right value for δ is roughly $1/\sqrt{m}$. We provide a precise description of this algorithm below and explain how to set η_0 and why $\eta_K > m/\varepsilon$ suffices in later sections.

Primal Path Following IPM for Linear Programming:

1. Find an initial $\eta_0 > 0$ and x_0 with $\left\|n_{\eta_0}(x_0)\right\|_{x_0} < \frac{1}{6}$.
2. Repeat for $k = 0, 1, 2, \ldots, T$ where T is s.t. $\eta_T > \frac{m}{\varepsilon}$
   - compute x_{k+1} according to the rule:
   $$x_{k+1} := x_k + n_{\eta_k}(x_k)$$
   - set $\eta_{k+1} := \eta_k \left(1 + \frac{1}{20\sqrt{m}}\right)$.
3. Calculate $\hat{x}$ by performing a constant number of Newton steps w.r.t. f_{η_T} (starting at x_T) and output $\hat{x}$.

The notion of closeness of a point x_k to the central path, i.e. to $x^\star_{\eta_k}$, that we employ here is

$$\left\|n_{\eta_k}(x_k)\right\|_{x_k} < 1/6.$$

It is directly motivated by the guarantee obtained in the previous lecture for Newton's method – this condition means precisely that x_k belongs to the quadratic convergence region for f_{η_k}.

4.2 Structure of the Analysis

We now provide an outline of the proof of Theorem 1 which is provided in the next section. We show that an appropriate implementation of the *Primal Path Following IPM* scheme yields a polynomial time algorithm for Linear Programming. This requires making the scheme a little bit more precise (explaining how to perform step 1.), however, the crucial part of the analysis is showing that the iteration in step 2. is indeed guaranteed to track the central path closely. To this end we want to prove that the following closeness invariant holds

Lemma 4 (Closeness Invariant)**.** *For every $k = 0, 1, 2, \dots, T$ it holds that*

$$\left\| n_{\eta_k}(x_k) \right\|_{x_k} < \frac{1}{6}.$$

To prove that this invariant holds, we consider one iteration and show that, if started with $\left\| n_{\eta_k}(x_k) \right\|_{x_k} < 1/6$, then after one iteration, still $\left\| n_{\eta_{k+1}}(x_{k+1}) \right\|_{x_{k+1}} < 1/6$. Every iteration consists of two steps: the Newton step w.r.t. f_{η_k} and the increase of η_k to η_{k+1}. We formulate another two lemmas that capture what happens when performing these steps.

Lemma 5 (Effect of a Newton step on centrality)**.** *The logarithmic barrier function F satisfies the* **NL** *condition. Therefore, for every $x \in \operatorname{int}(P)$ and every $\eta > 0$, such that $\left\| n_\eta(x) \right\|_x < \frac{1}{6}$, it holds*

$$\left\| n_\eta(x') \right\|_{x'} \le 3 \left\| n_\eta(x) \right\|_x^2$$

where $x' := x + n_\eta(x)$.

This implies that one Newton step brings always us significantly closer to the central path. The lemma below tells us that if we are very close to the central path, then it is "safe" to increase η_k by a factor of roughly $(1 + 1/\sqrt{m})$ so that the centrality is still preserved.

Lemma 6 (Effect of changing η on centrality)**.** *For every point $x \in \operatorname{int}(P)$ and every two positive $\eta, \eta' > 0$, we have*

$$\left\| n_{\eta'}(x) \right\|_x \le \frac{\eta'}{\eta} \left\| n_\eta(x) \right\|_x + \sqrt{m} \left| \frac{\eta'}{\eta} - 1 \right|.$$

This implies in particular that if we initialize the algorithm at, say, η_0 and terminate when η_T, then after $O(\sqrt{m} \log \frac{\eta_T}{\eta_0})$ iterations, each of which is just solving one $m \times m$ linear system, we recover an ε-approximate optimal solution.

This concludes the part of the analysis concerning the loop in step 2. of the *Primal Path Following IPM* and the issue of centrality. We now move to the part concerning initialization. We would like an algorithm to find a good starting point, i.e. one for which η_0 is not too small and for which the corresponding Newton step is "short".

Lemma 7 (Efficient Initialization)**.** *The step 1. of the the Primal Path Following IPM can be implemented in polynomial time to yield $\eta_0 = 2^{-\widetilde{O}(nL)}$ and $x_0 \in \operatorname{int}(P)$ such that $\left\| n_{\eta_0}(x_0) \right\|_{x_0} < \frac{1}{6}$.*

Note that it is crucial that the above Lemma provides a **lower bound** on η_0. This allows us to deduce that the number of iterations T in the *Primal Path Following IPM* is polynomially bounded and hence this scheme leads to a polynomial time algorithm for linear programming. It is also worth noting here that for well structured linear programs one can often find starting points with a larger value of η_0, such as $\eta_0 = \frac{1}{m}$ which then has only minor impact on the number of iterations.

Finally, we consider step 3. of the algorithm – termination, which says that by adjusting the point x_T just a little bit we can reach a point $\hat{x}$ very close to $x^\star$ in the objective value. More precisely we show.

Lemma 8 (Efficient Termination)**.** *By performing a constant number of Newton steps initialized at x_T in step 4. where T is such that $\eta_T \geq m/\varepsilon$ we obtain a point $\hat{x}$ that satisfies*

$$c^\top \hat{x} \leq c^\top x^\star + \varepsilon.$$

We are now ready to state the main theorem characterizing the performance of the *Primal Path Following IPM*.

Theorem 9 (Convergence of Path Following IPM)**.** *The Primal Path Following IPM, after $T = O\left(\sqrt{m}\log\frac{m}{\varepsilon\eta_0}\right)$ iterations, outputs a point $\hat{x} = x^\star_{\eta_T} \in P$ that satisfies*

$$c^\top x^\star_{\eta_T} \leq c^\top x^\star + \varepsilon.$$

Moreover, every iteration (step. 2) requires solving one linear system of the form $H(x)y = z$, where $x \in \operatorname{int}(P)$, $z \in \mathbb{R}^n$ is a given vector and y is the vector of variables. Thus it can be implemented in polynomial time.

Note now that by combining Theorem 9 with Lemma 7 the proof of Theorem 1 follows easily.

5 Analysis of the IPM algorithm for Linear Programming

5.1 Centrality

The purpose of this section is to prove Lemma 4, i.e., that the invariant

$$\left\| n_{\eta_k}(x_k) \right\|_{x_k} < \frac{1}{6}$$

holds for all $k = 1, 2, \ldots, T$. As explained in the outline, this just follows from Lemmas 5 and 6 – we now proceed with their proofs.

Effect of Newton's step on centrality

We recall the **NL** condition introduced in the previous lecture.

Definition 10 (Condition **NL**)**.** *Let f be a function; we say that f satisfies the* **NL** *condition if*

$$\forall x, y \text{ s.t. } \|y - x\|_x = \delta < 1, \quad (1-3\delta)H(x) \preceq H(y) \preceq (1+3\delta)H(x),$$

where $H(x) := \nabla^2 f(x)$ and $\|\cdot\|_x$ denotes the local norm.

Proof of Lemma 5. We need to prove that f_η (or equivalently F) satisfies the **NL** condition. To this end, recall that the Hessian $H(x) = \nabla^2 f_\eta(x) = \nabla^2 F(x)$ is given by

$$H(x) = \sum_{i=1}^{m} \frac{a_i a_i^\top}{s_i(x)^2}.$$

Consider now any two points x, y with $\|y - x\|_x = \delta < 1$. We have

$$\delta^2 = (y-x)^\top H(x)(y-x) = \sum_{i=1}^{m} \left| \frac{\langle a_i, y - x \rangle}{s_i(x)} \right|^2.$$

Thus in particular, every term in this sum is upper bounded by δ^2, hence for every $i = 1, 2, \ldots, m$ we have

$$\left| \frac{s_i(x) - s_i(y)}{s_i(x)} \right| = \left| \frac{\langle a_i, y - x \rangle}{s_i(x)} \right| \le \delta.$$

Consequently, for every $i = 1, 2, \ldots, m$ we have $(1-\delta)\,|s_i(x)| \le |s_i(y)| \le (1+\delta)\,|s_i(x)|$, and thus

$$\frac{(1+\delta)^{-2}}{s_i(x)^2} \le \frac{1}{s_i(y)^2} \le \frac{(1-\delta)^{-2}}{s_i(x)^2}.$$

It now follows that

$$\frac{(1+\delta)^{-2} a_i a_i^\top}{s_i(x)^2} \preceq \frac{a_i a_i^\top}{s_i(y)^2} \preceq \frac{(1-\delta)^{-2} a_i a_i^\top}{s_i(x)^2},$$

and thus, by summing the above over i we obtain

$$(1+\delta)^{-2} H(x) \preceq H(y) \preceq (1-\delta)^{-2} H(x).$$

To arrive at the **NL** condition it remains to observe that for every $\delta \in (0,1)$ we have $1 - 3\delta \le (1+\delta)^{-2}$ and $(1-\delta)^{-2} \le 1 + 3\delta$. □

Effect of the change of η on centrality

We proceed to the proof of Lemma 6 which asserts that changing η slightly does not increase $\|n_\eta(x)\|_x$ by much for a fixed point x. Let $g(x) := \nabla F(x)$.

Proof of Lemma 6. We have

$$\begin{aligned} n_{\eta'}(x) &= H(x)^{-1}(\eta' c + g(x)) \\ &= \frac{\eta'}{\eta} H(x)^{-1}(\eta c + g(x)) + \left(1 - \frac{\eta'}{\eta}\right) H(x)^{-1} g(x) \\ &= \frac{\eta'}{\eta} H(x)^{-1} \nabla f_\eta(x) + \left(1 - \frac{\eta'}{\eta}\right) H(x)^{-1} g(x). \end{aligned}$$

After taking norms and applying triangle inequality w.r.t. $\|\cdot\|_x$ we obtain

$$\left\|H(x)^{-1}\nabla f_{\eta'}(x)\right\|_x \leq \frac{\eta'}{\eta}\left\|H(x)^{-1}\nabla f_\eta(x)\right\|_x + \left|1-\frac{\eta'}{\eta}\right|\left\|H(x)^{-1}g(x)\right\|_x.$$

Let us stop here for a moment and try to understand what is the significance of the specific terms in the last expression. In our analysis of the algorithm, the term $\|H(x)^{-1}\nabla f_\eta(x)\|_x$ is a small constant. The goal is to show that the whole right hand side is bounded by a small constant as well. We should think of η' as $\eta(1+\delta)$ for some small $\delta > 0$. In such a setting $\frac{\eta'}{\eta}\|H(x)^{-1}(x)\nabla f_\eta(x)\|_x$ will be still a small constant, so what prevents us from choosing a large δ is the second term $|1-\frac{\eta'}{\eta}|\|H(x)^{-1}g(x)\|_x$. Thus what remains to do is to derive an upper bound on $\|H(x)^{-1}g(x)\|_x$. To this end we show that

$$\sup_{y\in\mathrm{int}(P)} \|H(y)^{-1}g(y)\|_y \leq \sqrt{m}.$$

To see this, let us pick any $y \in \mathrm{int}(P)$ and denote $z := H^{-1}g(y)$. From the Cauchy-Schwarz inequality we obtain

$$\|z\|_y^2 = g(y)^\top H(y)^{-1}g(y) = z^\top g(y) = \sum_{i=1}^m \frac{z^\top a_i}{s_i(y)} \leq \sqrt{m}\sqrt{\sum_{i=1}^m \frac{(z^\top a_i)^2}{s_i(y)^2}}. \tag{5}$$

Further, by inspecting the expression in the right hand side

$$\sum_{i=1}^m \frac{(z^\top a_i)^2}{s_i(y)^2} = z^\top\left(\sum_{i=1}^m \frac{a_i a_i^\top}{s_i(y)^2}\right)z = z^\top H(y)z = \|z\|_y^2. \tag{6}$$

Putting (5) and (6) together we obtain that $\|z\|_y^2 \leq \sqrt{m}\|z\|_y$, so in fact $\|z\|_y \leq \sqrt{m}$.

□

Concluding that the invariant holds

Proof of Lemma 4. Suppose that $\left\|n_{\eta_k}(x_k)\right\|_{x_k} < \frac{1}{6}$. Then by the quadratic convergence of Newton's method – Lemma 5, we obtain that x_{k+1} satisfies

$$\left\|n_{\eta_k}(x_{k+1})\right\|_{x_{k+1}} \leq 3\left\|n_{\eta_k}(x_k)\right\|_{x_k}^2 < \frac{1}{12}.$$

Further, by Lemma 6 it follows that

$$\left\|n_{\eta_{k+1}}(x_{k+1})\right\|_{x_{k+1}} \leq \frac{\eta_{k+1}}{\eta_k}\left\|n_{\eta_k}(x_{k+1})\right\|_{x_{k+1}} + \sqrt{m}\left|\frac{\eta_{k+1}}{\eta_k}-1\right| < (1+o(1))\cdot\frac{1}{12}+\frac{1}{20} < \frac{1}{6}.$$

□

5.2 Termination Condition

The discussion on the final point we divide into two parts. First we give a proof of Lemma 8 under the "ideal assumption" that in the 3rd step the IPM Path Following Algorithm, we actually reach $x^\star_{\eta_T}$ (and not only a point close-by). In a subsequent paragraph, by strenghtening this result we show that also an approximate minimum of f_{η_T} (which we get as a result of our algorithm) gives a suitable approximation guarantee.

Termination under "ideal assumption"

For simplicity here we make the "ideal assumption" that the point output by the algorithm in step 3. is really $\hat{x} := x^\star_{\eta_T}$. The lemma below explains our choice of $\eta \approx \frac{m}{\varepsilon}$ at which we stop the iteration in step 2.

Lemma 11 (Dependence of the approximation on the choice of η)**.** *For every $\eta > 0$ we have $c^\top x^\star_\eta - c^\top x^\star < \frac{m}{\eta}$.*

Proof. We start by writing down the derivative of $f_\eta(x)$

$$\nabla f_\eta(x) = \nabla(\eta c^\top x + F(x)) = \eta c + \nabla F(x) = \eta c + g(x).$$

The point $x^\star_\eta$ is the minimum of f_η, hence $\nabla f_\eta(x^\star_\eta) = 0$ and thus

$$g(x^\star_\eta) = -\eta c. \tag{7}$$

Using this observation we obtain that

$$c^\top x^\star_\eta - c^\top x^\star = -\left\langle c, x^\star - x^\star_\eta \right\rangle = \frac{1}{\eta}\left\langle g(x^\star_\eta), x^\star - x^\star_\eta \right\rangle.$$

To complete the proof it remains to argue that $\left\langle g(x^\star_\eta), x^\star - x^\star_\eta \right\rangle < m$. We show even more: for every two points x, y in the interior of P, we have $\langle g(x), y - x\rangle < m$. This follows by a simple calculation

$$\begin{aligned}
\langle g(x), y - x\rangle &= \sum_{i=1}^{m} \frac{a_i^\top (y - x)}{s_i(x)} \\
&= \sum_{i=1}^{m} \frac{(b_i - a_i^\top x) - (b_i - a_i^\top y)}{s_i(x)} \\
&= \sum_{i=1}^{m} \frac{s_i(x) - s_i(y)}{s_i(x)} \\
&= m - \sum_{i=1}^{m} \frac{s_i(y)}{s_i(x)} \\
&< m
\end{aligned}$$

Where in the last inequality we make use of the fact that our points x, y are strictly feasible, i.e. $s_i(x), s_i(y) > 0$ for all i.

$\square$

Dropping the "ideal assumption"

In this part we show that even after dropping the ideal assumption we still get an error which is $O(\varepsilon)$. For this, we perform a constant number of Newton iterations starting from x_T, so that the output $\hat{x}$ has $\left\|n_{\eta_T}(\hat{x})\right\|_{\hat{x}}$ extremely small.

To leverage on this fact, we derive a relation between the length of the Newton step at point x (with respect to the function f_η) and the distance to the optimum $x^\star_\eta$. We show that whenever $\|n(x)\|_x$ is sufficiently small, $\|x - x^\star_\eta\|_x$ is small as well. This together with a certain strenghtening of Lemma 11 imply that in the last step of *Primal Path Following IPM* we do not need to go with x_T to $x^\star_{\eta_T}$. In fact, only 2 additional Newton steps bring us (2ε)-close to the optimum.

Let us start by a certain extension of Lemma 11, which shows that to get a decent approximation of the optimum, we do not neccessarily need to be on the central path, but only close enough to it.

Lemma 12 (Approximation guarantee for points close to central path)**.** *For every point $x \in \operatorname{int}(P)$ and every $\eta > 0$, if $\|x - x^\star_\eta\|_x < 1$ then:*

$$c^\top x - c^\top x^\star \leq \frac{m}{\eta}\left(1 - \|x - x^\star_\eta\|_x\right)^{-1}.$$

Proof. For every $y \in \operatorname{int}(P)$ we have:

$$\begin{aligned} c^\top x - c^\top y &= c^\top (x - y) \\ &= \langle c, x - y \rangle \\ &= \langle c_x, x - y \rangle_x \\ &\leq \|c_x\|_x \|x - y\|_x \end{aligned}$$

Where $c_x = H(x)^{-1} c$ and the last inequality follows from Cauchy-Schwarz. Now, we want to bound $\|c_x\|_x$. Imagine we are at point x and we move in the direction of $-c_x$ until hitting the boundary of unit ball (in the local norm) around x. We will land in the point $x - \frac{c_x}{\|c_x\|_x}$, which is still inside P (as proved in Problem 1 of Homework 1). Therefore

$$\left\langle c, x - \frac{c_x}{\|c_x\|_x} \right\rangle \geq \langle c, x^\star \rangle.$$

Since $\langle c, c_x \rangle = \|c_x\|_x^2$, we get

$$\|c_x\|_x \leq \langle c, x \rangle - \langle c, x^\star \rangle.$$

We have obtained:

$$c^\top x - c^\top y \leq \|x-y\|_x (c^\top x - c^\top x^\star). \tag{8}$$

Now, let us express

$$c^\top x - c^\top x^\star = (c^\top x - c^\top x^\star_\eta) + (c^\top x^\star_\eta - c^\top x^\star)$$

and use (8) with $y = x^\star_\eta$. We obtain

$$c^\top x - c^\top x^\star \leq (c^\top x - c^\top x^\star)\|x-y\|_x + (c^\top x^\star_\eta - c^\top x^\star).$$

Thus

$$(c^\top x - c^\top x^\star)(1 - \|x-y\|_x) \leq c^\top x^\star_\eta - c^\top x^\star.$$

By applying Lemma 11 the result follows. □

Note that in the algorithm we never literally mention the condition that $\|x - x^\star_\eta\|_x$ is small. However, we show that it follows from $\|n_\eta(x)\|_x$ being small. In fact, we prove the following more general lemma which holds for any function f satisfying the **NL** condition.

Lemma 13 (Distance to the optimum and the Newton's step). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be any strictly convex function satisfying the* **NL** *condition. Let x be any point in the domain of f. Consider the Newton step $n(x)$ at a point x. If $\|n(x)\|_x < 1$, then $\|x - x^\star\|_x \leq 8\|n(x)\|_x$, where $x^\star$ is the minimizer of f.*

Proof. Pick any h such that $\|h\|_x \leq \frac{1}{4}$. Expand $f_\eta(x+h)$ into a Taylor series around x:

$$f(x+h) = f(x) + h^\top \nabla f(x) + \frac{1}{2} h^\top \nabla^2 f(\theta) h \tag{9}$$

for some point θ lying on the segment $[x, x+h]$. We proceed by lower-bounding the linear term. Note that:

$$\left| h^\top \nabla f(x) \right| = |\langle h, n(x)\rangle_x| \leq \|h\|_x \|n(x)\|_x \tag{10}$$

by Cauchy-Schwarz. Next:

$$h^\top \nabla^2 f(\theta) h = h^\top H(\theta) h \geq \frac{1}{4} h^\top H(x) h \tag{11}$$

where we used the **NL** condition for f. Applying bounds (10), (11) to the expansion (9) results in:

$$f(x+h) \geq f(x) - \|h\|_x \|n(x)\|_x + \frac{1}{8}\|h\|_x^2. \tag{12}$$

Set $r = 8\|n(x)\|_x$ and consider points y satysfing $\|x-y\|_x = r$, i.e. points on the boundary of the local norm ball of radius r centered at x. Then 12 simplifies to:

$$f(x+h) \geq f(x)$$

which implies that $x^\star$, the unique minimizer of f, belongs to the above mentioned ball and the theorem follows.

□

Proof of Lemma 8. Combining Lemma 12 with Lemma 13 we obtain that if $\eta \geq \frac{m}{\varepsilon}$ and $\|n_\eta(x)\|_x \leq \frac{1}{16}$ then $c^\top x - c^\top x^\star \leq 2\varepsilon$. □

5.3 Starting Point

In this section we give a method for finding an appropriate starting point. More precisely, we show how to find efficiently some $\eta_0 > 0$ and x_0 such that $\|n(x_0)\|_{x_0} < 1/6$.

Before we start, we would like to remark that this discussion provides a very small η_0, of order $2^{-\mathrm{poly}(L)}$. This enables us to prove that in fact IPM can solve linear programming in polynomial time, but does not seem promising when trying to apply IPM to devise fast algorithms for combinatorial problems. Indeed, there is a factor of $\log \eta_0^{-1}$ in the bound on the number of iterations, which translates to L for such a tiny η_0. To make an algorithm fast, we need to have ideally $\eta_0 = \Omega(1/\mathrm{poly(m)})$. It turns out that for specific problems (such as maximum flow) we can often devise some specialized methods to find such η_0 and x_0.

Finding (η_0, x_0) given a point in the interior of P

First, we show how given a point $x' \in \mathrm{int}(P)$ we can find some starting pair (η_0, x_0). Let us assume now that such a point x' is given. Furthermore, we assume that each constraint is satisfied with slack at least $2^{-\widetilde{O}(nL)}$ at x', that is $b_i - a_i^\top x' \geq 2^{-\widetilde{O}(nL)}$. Our procedure for finding a point in P will provide such an x' based on our assumption that P is full dimensional. [4]

Recall that we want to find a point x_0 close to the central path

$$\Gamma_c := \{x_\eta^\star : \eta \geq 0\},$$

which corresponds to the objective function $c^\top x$. Note that as $\eta \to 0$, $x_\eta^\star \to x_0^\star = x^\circ$, the analytic center of P. Hence, finding a point x_0, very close to the analytic center and choosing η_0 to be some tiny number should be a good strategy. In fact it is, but how to find a point close to x°?

[4]In this presentation we do not discuss all the details, e.g. the full-dimensionality assumption. These are easy to deal with, yet tedious. We refer the reader to [2] for a thorough treatment.

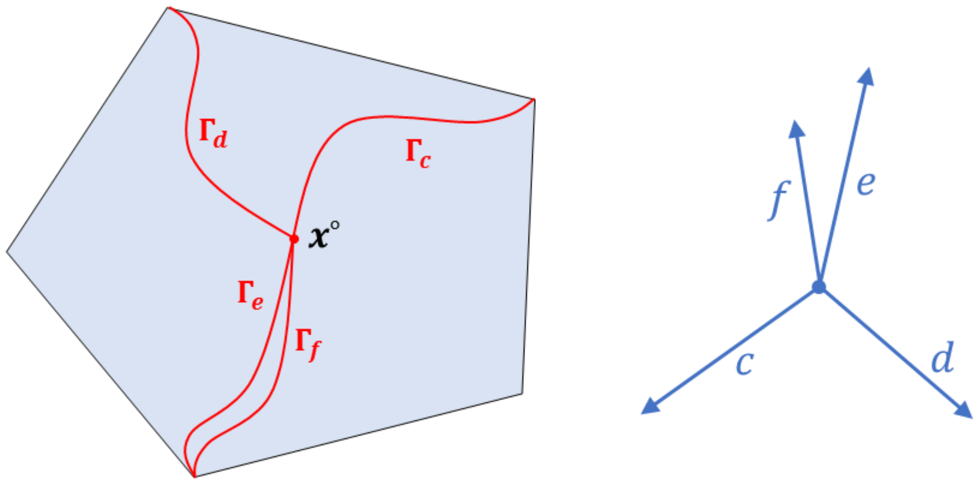

Figure 2: Illustration of central paths $\Gamma_c, \Gamma_d, \Gamma_e, \Gamma_f$ for four different objective vectors c, d, e and f. All the paths originate at the analytic center x° and converge to some boundary point of P. Note that because we minimize the linear objective, the cost vectors point in the "opposite" direction to the direction traversed by the corresponding central path.

The path Γ_c is of main interest to us because it approaches the optimal solution to our linear program. However, in general, one might also define other central paths. If $d \in \mathbb{R}^n$ is any vector, we may define

$$\Gamma_d := \left\{ \operatorname*{argmin}_{x \in \text{int}(P)} \left(\eta d^\top x + F(x) \right) : \eta \geq 0 \right\},$$

i.e., the path consisting of minimizers to perturbed objective $d^\top x$. What do all the paths Γ_d have in common? The origin! They all start at the same point: analytic center of P, see Figure 2 for an illustration. Our strategy will be to pick one such path on which x' lies and traverse it backwards to reach a point very close to the origin of this path, which at the same time will be a good choice for x_0.
Recall that g is the gradient of the logarithimic barrier F and define $d = -g(x')$. Now it turns out that $x' \in \Gamma_d$. Why is this? Denote $f'_\eta(x) = \eta d^\top x + F(x)$ and let $x'^\star_\eta$ be the minimizer of f'_η. Then $x'^\star_1 = x'$, since $\nabla f'_1(x') = 0$. We will use $n'_\eta(x)$ for the Newton step at x with respect to f'_η. We see in particular that $n'_1(x') = 0$.

As mentioned above, our goal is to move along the Γ_d path in the direction of decreasing η. We will use exactly the same method as in the *Primal Path Following*. We perform steps, in each of them we make one Newton step w.r.t. current η and then decrease η by a factor of $(1 - 1/20\sqrt{m})$. At each step it holds that $\|n'_\eta(x)\|_x < 1/6$, by an argument identical to the proof of Lemma 4. It remains to see, how small η is required (this will determine the number of iterations we need to perform) so that we can jump from Γ_d to Γ_c.

For this we first prove that by taking η small enough we reach a point x very close to the analytic center (i.e. for which $\left\|H(x)^{-1}g(x)\right\|_x$ is small). Next we show that having such a point x, with an appropriate η_0 satisfies $\left\|n_{\eta_0}(x)\right\| < \frac{1}{6}$. The first part is formalized in the lemma below.

Lemma 14. *Suppose that x' is a point such that*

$$\forall i = 1, 2, \dots, m \qquad s_i(x') \geq \delta \cdot \max_{x \in P} s_i(x).$$

For any $\eta > 0$ denote by $n'_\eta(x)$ the Newton step $n'_\eta(x) := -H(x)^{-1}(\eta g(x') + g(x))$. Then, whenever $\eta < \frac{\delta}{24\sqrt{m}}$ and $\left\|n'_\eta(x)\right\|_x < \frac{1}{24}$, it holds

$$\left\|H(x)^{-1}g(x)\right\|_x < \frac{1}{12}.$$

Proof. From the triangle inequality we have

$$\left\|H(x)^{-1}g(x)\right\|_x \leq \left\|n'_\eta(x)\right\|_x + \eta \left\|H(x)^{-1}g(x')\right\|_x.$$

Thus it remains to show a suitable upper bound on $\left\|H(x)^{-1}g(x')\right\|_x$. Denote by S_x and $S_{x'}$ the diagonal matrices with $s(x)$ and $s(x')$ (the slack vectors at x and x' respectively) on the diagonals. Then, one can compactly write

$$H(x) = A^\top S_x^{-2} A, \qquad g(x) = A^\top S_x^{-1} 1 \qquad g(x') = A^\top S_{x'}^{-1} 1,$$

where $1 \in \mathbb{R}^m$ is the all-one vector. We have

$$\left\|H(x)^{-1}g(x')\right\|_x^2 = g(x')^\top H(x)^{-1} g(x') = 1^\top S_{x'}^{-1} A (A^\top S_x^{-2} A)^{-1} A^\top S_{x'}^{-1} 1.$$

This can be also rewritten as

$$\left\|H(x)^{-1}g(x')\right\|_x^2 = v^\top \Pi v$$

where $\Pi := S_x^{-1} A (A^\top S_x^{-2} A)^{-1} A^\top S_x^{-1}$ and $v := S_{x'}^{-1} S_x 1$. One can note now that Π is an orthogonal projection matrix: indeed Π is orthogonal and $\Pi^2 = \Pi$. Therefore

$$v^\top \Pi v = \|\Pi v\|^2 \leq \|v\|^2.$$

Further

$$\|v\|^2 = \sum_{i=1}^m \frac{s_i(x)^2}{s_i(x')^2} \leq \frac{m}{\delta^2}.$$

Thus, concluding the above derivation, we obtain

$$\left\|H(x)^{-1}g(x')\right\|_x \leq \frac{\sqrt{m}}{\delta},$$

and the lemma follows. □

As mentioned above, the next lemma shows that given a point close to the analytic center, we can use it to initialize our algorithm, assuming that η_0 is small enough.

Lemma 15. *Suppose that $x \in \operatorname{int}(P)$ is a point such that $\left\|H(x)^{-1}g(x)\right\|_x < \frac{1}{12}$ and*

$$\eta_0 \leq \frac{1}{12} \cdot \frac{1}{c^\top x - c^\top x^\star},$$

where $x^\star$ is the optimal solution to the linear program. Then $\left\|n_{\eta_0}(x)\right\|_x < \frac{1}{6}$.

Proof. We have

$$\left\|n_{\eta_0}(x)\right\|_x = \left\|H(x)^{-1}(\eta_0 c + g(x))\right\|_x \leq \eta_0 \left\|H(x)^{-1}c\right\|_x + \left\|H(x)^{-1}g(x)\right\|_x.$$

Since, from assumption $\left\|H(x)^{-1}g(x)\right\|_x < \frac{1}{12}$, to arrive at the conclusion it suffices to prove that

$$\left\|H(x)^{-1}c\right\|_x \leq c^\top x - c^\top x^\star.$$

To see this, denote $c_x = H(x)^{-1}c$ and observe that since the Dikin Ellipsoid $E_x = \{y : (y-x)^\top H(x)(y-x) \leq 1\}$ is a subset of P (Exercise 1 in Problem Set 1), we know that $x - \frac{c_x}{\|c_x\|_x} \in P$ (because this point belongs to E_x). Therefore

$$\left\langle c, x - \frac{c_x}{\|c_x\|_x} \right\rangle \geq \langle c, x^\star \rangle,$$

since $x^\star$ minimizes this linear objective over P. The above, after rewriting gives

$$\left\langle c, \frac{c_x}{\|c_x\|_x} \right\rangle \leq \langle c, x \rangle - \langle c, x^\star \rangle.$$

By observing that

$$\left\langle c, \frac{c_x}{\|c_x\|_x} \right\rangle = \left\|H(x)^{-1}c\right\|,$$

the lemma follows. □

Given Lemmas 14 and 15 we are now ready to prove that a good starting point can be found in polynomial time.

Lemma 16 (Finding a starting point given interior point). *There is an algorithm which given a point $x' \in \operatorname{int}(P)$ such that*

$$\forall i = 1, 2, \ldots, m \qquad s_i(x') \geq \delta \cdot \max_{x \in P} s_i(x),$$

outputs a point $x_0 \in \operatorname{int}(P)$ and $\eta_0 > 0$ such that $\left\| n_{\eta_0}(x_0) \right\|_{x_0} < \frac{1}{6}$ and $\eta_0 \geq \frac{1}{\|c\|_2 \cdot \operatorname{diam}(P)} \geq 2^{-\widetilde{O}(nL)}$. The running time of this algorithm is polynomial in n, m, L and $\log \delta$.

Proof. The lemma is a straightforward combination of Lemmas 14 and 15. It remains to observe that for $x \in P$, using the Cauchy-Schwarz inequality

$$c^\top x - c^\top x^\star = \langle c, x - x^\star \rangle \leq \|c\|_2 \cdot \|x - x^\star\| \leq \|c\|_2 \cdot \operatorname{diam}(P).$$

Further $\operatorname{diam}(P) \leq 2^{\widetilde{O}(nL)}$ using Exercise 1 in Problem Set 7. $\square$

Finding a point in the interior of P

To complete our considerations we show how to find a suitable $x' \in \operatorname{int}(P)$. To this end, we consider an auxiliary linear program:

$$\begin{aligned} \min_{(t,x) \in \mathbb{R}^{n+1}} \quad & t \\ & a_i^\top x \leq b_i + t, \text{ for all } i \\ & -2^{\widetilde{O}(nL)} \leq t \leq 2^{\widetilde{O}(nL)} \end{aligned} \tag{13}$$

Taking $x = 0$ and t big enough ($t = 1 + \sum_i |b_i|$), we obtain a strictly feasible solution with at least $O(1)$ slack at every constraint. Thus we can use the interior point method we derived to solve it up to precision $2^{-\widetilde{\Omega}(nL)}$ in polynomial time. If P is full-dimensional and non-empty, then the optimal solution t_{opt} to (13) is negative and in fact $t_{opt} \leq -2^{-\widetilde{O}(nL)}$ (see Exercise 1 in Problem Set 7). Thus by solving (13) up to $2^{-\widetilde{\Omega}(nL)}$ precision, we obtain a feasible point x' whose all slacks are at least $2^{-\widetilde{O}(nL)}$, hence δ in Lemma 16 is $2^{-\widetilde{O}(nL)}$.

5.4 Conclusion: proof of Theorem 9

Theorem 9 can be now easily concluded from Lemmas 11 and 4. Indeed, Lemma 11 says that $\hat{x}$ is an ε-approximate solution whenever $\eta \geq \frac{m}{\varepsilon}$. Since $\eta_T = \eta_0 \left(1 + \frac{1}{20\sqrt{m}}\right)^T$, it is enough to take $T = \Omega\left(\sqrt{m} \log \frac{m}{\varepsilon \eta_0}\right)$ for $\eta_T \geq \frac{m}{\varepsilon}$ to hold.

Since the invariant $\left\| n_{\eta_k}(x_k) \right\|_{x_k} < \frac{1}{6}$ is satisfied at every step k (by Lemma 4), it is in particular satisfied for $k = T$ and thus x_T lies in the region of quadratic convergence of Newton's method for f_{η_T}, hence $x^\star_{\eta_T}$ can be computed from x_T. Note that we are not entirely precise here: we make a simplifying assumption that once we arrive at the region of quadratic convergence around $x^{\star}_{\eta_T}$ then we can actually reach this point – we show how to get around this issue in Section 5.2.

It remains to observe that in every iteration, the only nontrivial operation is computing the Newton step $n_{\eta_k}(x_k)$, which in turn boils down to solving the following linear system

$$H(x_k)y = \nabla f_{\eta_k}(x_k).$$

Note in particular that computing the inverse of $H(x_k)$ is not necessary! Just solving the above linear system might be much faster than inverting a matrix – indeed, when $H(x)$ is a Laplacian matrix, it follows from the result of Spielman and Teng [10] that such a system is solvable in time nearly linear in the number of non-zero entries in $H(x)$.

A A Brief History of IPMs

Interior point methods (IPMs), first appear in a recognizable form in Dikin's Ph.D. thesis in the 60s, yet with no analysis of the convergence time. In 1984 Karmarkar announced a polynomial-time algorithm to solve linear programs using an IPM based on projective scaling [4]. At that point there was already a known polynomial time algorithm for solving LPs, namely the Ellipsoid Algorithm from by Khachiyan [5]. Furthermore, the method of choice in practice, despite known to be inefficient in the worst case, was the Simplex method. Karmarkar, in his paper, presented also empirical evidence showing that his algorithm was consistently 50 times faster than the Simplex method. This event, which received publicity around the world throughout the popular press and media, started a new line of work on algorithms following the "interior point" idea. For a comprehensive historical perspective on this line of work, we refer to the survey [11].

Karmarkar's algorithm needed roughly $O(m \log 1/\varepsilon)$ iterations[5] to find a solution (optimal up to an additive error of ε) to a linear program, where m is the number of constraints. Each such iteration involved solving a linear system of equations, which could be easily performed in polynomial time. Thanks to the logarithmic dependence on the error ε it can be used to find the exact, optimal solution to a linear program in time polynomial with respect to the encoding size of the problem. Thus, Karmarkar's algorithm was the first efficient algorithm with provable worst case polynomial running time. Subsequently, Renegar proposed an interior point algorithm with reduced number of iterations: $O(\sqrt{m}\log(1/\varepsilon))$ [9]. Around the same time, Nesterov and Nemirovski abstracted out the essence of interior point methods and came up with the the notion of *self-concordance* [8], which in turn was used to provide efficient, polynomial time algorithms for many non-linear convex problems such as semi-definite programming.

[5]For clarity, the dependency on the bit complexity parameter L is hidden, when discussing the running times of various algorithms.

References

[1] Sébastien Bubeck and Ronen Eldan. The entropic barrier: a simple and optimal universal self-concordant barrier. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, page 279, 2015.

[2] Martin Grötschel, Lászlo Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[3] Roland Hildebrand. Canonical barriers on convex cones. *Mathematics of operations research*, 39(3):841–850, 2014.

[4] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[5] Leonid Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53 – 72, 1980.

[6] Yin Tat Lee and Aaron Sidford. Matching the universal barrier without paying the costs : Solving linear programs with õ(sqrt(rank)) linear system solves. *CoRR*, abs/1312.6677, 2013.

[7] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in õ(vrank) iterations and faster algorithms for maximum flow. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 424–433, 2014.

[8] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

[9] James Renegar. A polynomial-time algorithm, based on newton's method, for linear programming. *Math. Program.*, 40(1-3):59–93, 1988.

[10] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *STOC'04: Proceedings of the 36th Annual ACM Symposium on the Theory of Computing*, pages 81–90, 2004.

[11] Margaret H. Wright. The interior-point revolution in optimization: history, recent developments, and lasting consequences. *Bull. Amer. Math. Soc. (N.S*, 42:39–56, 2005.