20
Interior-Point Methods

In this chapter, we continue our discussion of polynomial-time algorithms for linear programming, and cover the high-level details of an interior-point algorithm. The runtime for these linear programs has recently been improved both qualitatively and quantitatively, so this is an active area of research that you may be interested in. Moreover, these algorithms contain sophisticated general ideas (duality and the method of Lagrange multipliers, and the use of barrier functions) that are important even beyond this context.

Another advantage of delving into the details of these methods is that we can work on getting better algorithms for special kinds of linear programs of interest to us. For instance, the line of work on faster max-flow algorithms for directed graphs, starting with the work of Madry, and currently resulting in the $O(m^{4/3+\varepsilon})$-time algorithms of Kathuria, and of Liu and Sidford, are based on a better understanding of interior-point methods.

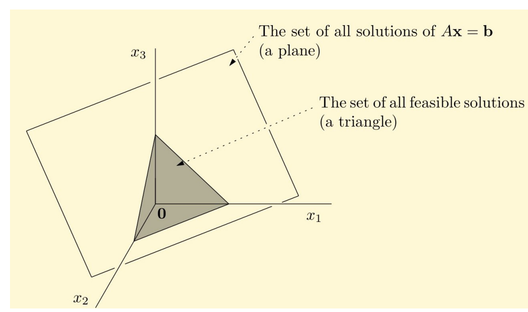

Figure 20.1: The feasible region for an LP in equational form (from the Matoušek and Gärtner book).

We will consider the following LP with *equality* constraints:

$$\begin{aligned} \min\ & c^\intercal x \\ Ax &= b \\ x &\geq 0 \end{aligned}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $c, x \in \mathbb{R}^n$. Let $K := \{x \mid Ax = b, x \geq 0\}$ be the polyhedron, and $x^* = \arg\min\{c^\intercal x \mid x \in K\}$ an optimal solution.

To get the main ideas across, we make some simplifying assumptions and skip over some portions of the algorithm. For more details, please refer to the book by Jiri Matoušek and Bernd Gärtner (which has more details), or the one by Steve Wright (which has most details).

20.1 Barrier Functions

The first step in solving the LP using an interior-point method will be to introduce a parameter $\eta > 0$ and exchange our constrained linear optimization problem for an unconstrained but *nonlinear* one:

$$f_\eta(x) := c^\intercal x + \eta\Big(\sum_{i=1}^{n} \log \frac{1}{x_i}\Big).$$

Let $x^*_\eta := \arg\min\{f_\eta(x) \mid Ax = b\}$ be the minimizer of this function over the subspace given by the equality constraints. Note that we've added in η times a ***barrier function***

$$B(x) := \sum_{i=1}^{n} \log \frac{1}{x_i}.$$

The intuition is that when x approaches the boundary $x \geq 0$ of the feasible region, the barrier function $B(x)$ will approach $+\infty$. The parameter η lets us control the influence of this barrier function. If η is sufficiently large, the contribution of the barrier function dominates in $f_\eta(x)$, and the minimizer x^*_η will be close to the "center" of the feasible region. However, as η gets close to 0, the effect of $B(x)$ will diminish and the term $c^\intercal x$ will now dominate, causing that x^*_η to approach x^*.

If we had *inequality constraints* $Ax \geq b$ as well, we would have added $\sum_{i=1}^{m} \log \frac{1}{a_i^\intercal x - b_i}$ to the barrier function.

Now consider the trajectory of the minimizer x^*_η as we lower η continuously, starting at some large value and tending to zero: this path is called the ***central path***. The idea of our ***path-following*** algorithm will be to approximately follow this path. In essence, such algorithms conceptually perform the following steps (although we will only approximate these steps in practice):

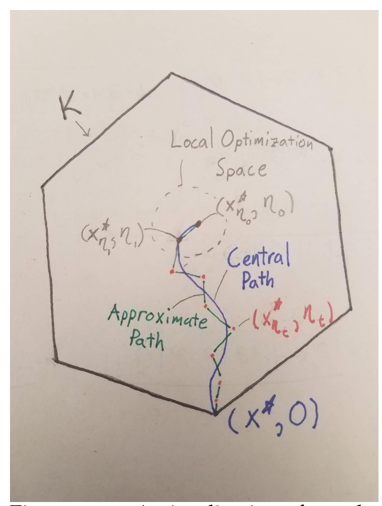

Figure 20.2: A visualization of a path-following algorithm.

1. Pick a sufficiently large η_0 and a starting point $x^{(0)}$ that is the minimizer of $f_{\eta_0}(x)$. (We will ignore this step in our discussion, for now.)

2. At step t, move to the corresponding minimizer $x^{(t+1)}$ for $f_{\eta_{t+1}}$, where
$$\eta_{t+1} := \eta_t \cdot (1 - \epsilon).$$
Since η_t is close to η_{t+1}, we hope that the previous minimizer $x^{(t)}$ is close enough to the current goal $x^{(t+1)}$ for us to find it efficiently.

3. Repeat until η is small enough that x^*_η is very close to an optimal solution x^*. At this point, round it to get a vertex solution, like in §19.4.1.

We will only sketch the high-level idea behind Step 1 (finding the starting solution), and will skip Step 2 (the rounding); our focus will be on the update step. To understand this step, let us look at the structure of the minimizers for $f_\eta(x)$.

20.1.1 *The Primal and Dual LPs, and the Duality Gap*

Recall the primal linear program:

$$\begin{aligned} (P)\ \min\ & c^\intercal x \\ Ax &= b \\ x &\geq 0, \end{aligned}$$

and its dual:

$$\begin{aligned} (D)\ \max\ & b^\intercal y \\ A^\intercal y &\leq c. \end{aligned}$$

We can rewrite the dual using non-negative *slack variables* s:

$$\begin{aligned} (D')\ \max\ & b^\intercal y \\ A^\intercal y + s &= c \\ s &\geq 0. \end{aligned}$$

We assume that both the primal (P) and dual (D) are *strictly feasible*: i.e., they have solutions even if we replace the inequalities with strict ones). Then we can prove the following result, which relates the optimizer for f_η to feasible primal and dual solutions:

Lemma 20.1 (Optimality Conditions)**.** *The point $x \in \mathbb{R}^n_{\geq 0}$ is a minimizer of $f_\eta(x)$ if and only if there exist $y \in \mathbb{R}^m$ and $s \in \mathbb{R}^n_{\geq 0}$ such that:*

$$Ax - b = 0 \tag{20.1}$$

$$A^\intercal y + s = c \tag{20.2}$$

$$\forall i \in [n] : s_i x_i = \eta \tag{20.3}$$

The conditions (20.1) and (20.2) show that x and (y, s) are feasible for the primal (P) and dual (D') respectively. The condition (20.3) is an analog of the usual complementary slackness result that arises when $\eta = 0$. To prove this lemma, we use the method of Lagrange multipliers.

Observe: we get that ***if*** there exists a maximum x^*, then x^* satisfies these conditions.

Theorem 20.2 (The Method of Lagrange Multipliers)**.** *Let functions f and $g_1, \cdots, g_m$ be continuously differentiable, and defined on some open subset of $\mathbb{R}^n$. If x^* is a local optimum of the following optimization problem*

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } \forall i \in [m] : g_i(x) &= 0 \end{aligned}$$

then there exists $y^* \in \mathbb{R}^m$ such that $\nabla f(x^*) = \sum_{i=1}^m y_i^* \cdot \nabla g_i(x^*)$.

Proof Sketch of Lemma 20.1. We need to show three things:

1. The function $f_\eta(x)$ achieves its maximum x^* in the feasible region.
2. The point x^* satisfies the conditions (20.1)–(20.3).
3. And that no other x satisfies these conditions.

The first step uses that if there are strictly feasible primal and dual solutions $(\hat{x}, \hat{y}, \hat{s})$, then the region $\{x \mid Ax = b, f_\mu(x) \leq f_\mu \hat{x}\}$ is bounded (and clearly closed) and hence the continuous function $f_\mu(x)$ achieves its minimum at some point x^* inside this region, by the Extreme Value theorem. (See Lemma 7.2.1 of Matoušek and Gärtner, say.)

For the second step, we use the functions $f_\mu(x)$, and $g_i(x) = a_i^\intercal x - b_i$ in Theorem 20.2 to get the existence of $y^* \in \mathbb{R}^m$ such that:

$$f_\eta(x^*) = \sum_{i=1}^{m} y_i^* \cdot \nabla(a_i^\intercal x^* - b_i) \quad \Longleftrightarrow \quad c - \eta \cdot \left(1/x_1^*, \cdots, 1/x_n^*\right)^\intercal = \sum_{i=1}^{m} y_i^* \, a_i.$$

Define a vector s^* with $s_i^* = \eta / x_i^*$. The above condition is now equivalent to setting $A^\intercal y^* + s^* = c$ and $s_i^* x_i^* = \eta$ for all i.

Finally, for the third step of the proof, the function $f_\eta(x)$ is strictly convex and has a unique local/global optimum. Finish this proof. □

By weak duality, the optimal value of the linear program lies between the values of any feasible primal and dual solution, so the ***duality gap*** $c^\intercal x - b^\intercal y$ bounds the suboptimality $c^\intercal x - OPT$ of our current solution. Lemma 20.1 allows us to relate the duality gap to η as follows.

$$c^\intercal x - b^\intercal y = c^\intercal x - (Ax)^\intercal y = x^\intercal c - x^\intercal (c - s) = x^\intercal s = n \cdot \eta.$$

If the representation size of the original LP is $L := \langle A \rangle + \langle b \rangle + \langle c \rangle$, then making $\eta \leq 2^{-\operatorname{poly}(L)}$ means we have primal and dual solutions whose values are close enough to optimal, and can be rounded (using the usual simultaneous Diophantine equations approach used for Ellipsoid).

20.2 *The Update Step*

Let us now return to the question of obtaining $x^{(t+1)}$ from $x^{(t)}$ at step t? Recall, we want $x^{(t+1)}$ to satisfy the optimality conditions (20.1)–(20.3) for $f_{\eta_{t+1}}$. The hurdles to finding this point directly are: (a) the non-negativity of the x, s variables means this is not just a linear system, there are *inequalities* to contend with. And more worryingly, (b) it is not a *linear* system at all: we have non-linearity in the constraints (20.3) because of multiplying x_i with s_i.

To get around this, we use a "local-search" method. We start with the solution $x^{(t)}$ "close to" the optimal solution $x^*_{\eta_t}$ for f_{η_t}, and take a small step, so that we remain non-negative, and also get "close to" the optimal solution $x^*_{\eta_{t+1}}$ for $f_{\eta_{t+1}}$. Then we lower η and repeat this process.

Let us make these precise. First, to avoid all the superscripts, we use (x, y, s) and η to denote $(x^{(t)}, y^{(t)}, s^{(t)})$ and η_t. Similarly, (x', y', s') and η' denote the corresponding values at time $t+1$. Now we assume we have (x, y, s) with $x, s > 0$, and also:

$$Ax = b \tag{20.4}$$

$$A^\intercal y + s = c \tag{20.5}$$

$$\sum_{i=1}^{n} \left(s_i x_i - \eta_t\right)^2 \le (\eta_t/4)^2. \tag{20.6}$$

The first two are again feasibility conditions for (P) and (D'). The third condition is new, and is an approximate version of (20.3). Suppose that

$$\eta' := \eta' \cdot (1 - \frac{1}{4\sqrt{n}}).$$

Our goal is a new solution $x' = x + \Delta x$, $y' = y + \Delta y$, $s' = s + \Delta s$, which satisfies non-negativity, and ideally also satisfies the original optimality conditions (20.1)–(20.3) for the new η'. (Of course, we will fail and only satisfy the weaker condition (20.6) instead of (20.3), but we should aim high.)

Let us write the goal explicitly, by substituting (x', y', s') into (20.4)–(20.6) and using the feasibility of (x, y, s). This means the increments $\Delta x, \Delta y, \Delta s$ satisfy

$$\begin{aligned} A\,(\Delta x) &= 0 \\ A^\intercal\,(\Delta y) + (\Delta s) &= 0 \\ s_i(\Delta x_i) + (\Delta s_i)x_i + (\Delta s_i)(\Delta x_i) &= \eta' - x_i s_i. \end{aligned}$$

Note the quadratic term in blue. Since we are aiming for an approximation anyways, and these increments are meant to be tiny, we drop the quadratic term to get a *system of linear equations* in these increments:

$$\begin{aligned} A\,(\Delta x) &= 0 \\ A^\intercal\,(\Delta y) + (\Delta s) &= 0 \\ s_i(\Delta x_i) + (\Delta s_i)x_i &= \eta' - x_i s_i. \end{aligned}$$

This is often written in the following matrix notation (which I am putting down just so that you recognize it the next time you see it):

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^\intercal & I \\ \operatorname{diag}(x) & 0 & \operatorname{diag}(s) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \eta' \mathbb{1} - x \circ s \end{bmatrix}.$$

Here $x \circ s$ stands for the component-wise product of the vectors x and s. The bottom line: this is a linear system and we can solve it, say using Gaussian elimination. Now we can set $x' = x + \Delta x$, etc., to get the new point (x', y', s'). It remains to check the non-negativity and also the weakened conditions (20.4)–(20.6) with respect to η'.

The goal of many modern algorithms is to get faster ways to solve this linear system. E.g., if it were a Laplacian system we could (approximately) solve it in near-linear time.

20.2.1 *Properties of the New Solution*

While discarding the quadratic terms means we do not satisfy $x_i s_i = \eta$ for each coordinate i, we can show that we satisfy it on average, allowing us to bound the duality gap.

Lemma 20.3. *The new duality gap is $\langle x', s' \rangle = n\, \eta'$.*

Proof. The last set of equalities in the linear system ensure that

$$s_i x_i + s_i(\Delta x_i) + (\Delta s_i) x_i = \eta', \tag{20.7}$$

so we get

$$\begin{aligned} \langle x', s' \rangle &= \langle x + \Delta x, s + \Delta s \rangle \\ &= \sum_i \big(s_i x_i + s_i(\Delta x_i) + (\Delta s_i) x_i \big) + \langle \Delta x, \Delta s \rangle \\ &= n\eta' + \langle \Delta x, -A^\intercal(\Delta y) \rangle \\ &= n \cdot \eta' - \langle A(\Delta x), \Delta y \rangle \\ &= n \cdot \eta', \end{aligned}$$

using the first two equality constraints of the linear system. □

We explicitly maintained the invariants given by (20.4), (20.5), so it remains to check (20.6). This requires just a bit of algebra (that also causes the $\sqrt{n}$ to pop out).

Lemma 20.4. $\sum_{i=1}^n \big(s_i' x_i' - \eta' \big)^2 \leq (\eta'/4)^2$.

Proof. As in the proof of Lemma 20.3, we get that $s_i' x_i' - \eta' = (\Delta s_i)(\Delta x_i)$, so it suffices to show that

$$\sqrt{\sum_{i=1}^n (\Delta s_i)^2 (\Delta x_i)^2} \leq \eta'/4.$$

We can use the inequality

$$\sqrt{\sum_i a_i^2 b_i^2} \leq \frac{1}{4} \sum_i (a_i + b_i)^2 = \frac{1}{4} \sum_i (a_i^2 + b_i^2 + 2a_i b_i),$$

where we set $a_i^2 = \frac{x_i(\Delta s_i)^2}{s_i}$ and $b_i^2 = \frac{s_i(\Delta x_i)^2}{x_i}$. Hence

$$\begin{aligned}
\sqrt{\sum_{i=1}^n (\Delta s_i \Delta x_i)^2} &\le \frac{1}{4}\sum_{i=1}^n \left(\frac{x_i}{s_i}\cdot(\Delta s_i)^2 + \frac{s_i}{x_i}\cdot(\Delta x_i)^2 + 2(\Delta s_i)(\Delta x_i)\right) \\
&= \frac{1}{4}\sum_{i=1}^n \frac{(x_i\Delta s_i)^2 + (s_i \Delta x_i)^2}{s_i x_i} \qquad \text{[since } (\Delta s)^\intercal \Delta x = 0 \text{ by Claim 20.3]} \\
&\le \frac{1}{4}\frac{\sum_{i=1}^n (x_i \Delta s_i + s_i \Delta x_i)^2}{\min_{i\in[n]} s_i x_i} \\
&= \frac{1}{4}\frac{\sum_{i=1}^n (\eta' - s_i x_i)^2}{\min_{i\in[n]} s_i x_i}. \qquad (20.8)
\end{aligned}$$

We now bound the numerator and denominator separately.

This claim and proof are incorrect. However, we can prove that $\min_i s_i x_i \ge 3\eta/4$. This weaker claim suffices for the rest of the proof. The details will come soon, sorry for the mistake!

Claim 20.5 (Denominator). $\min_i s_i x_i \ge \eta\left(1 - \frac{1}{4\sqrt{n}}\right)$.

Proof. By the inductive hypothesis, $\sum_i (s_i x_i - \eta)^2 \le (\eta/4)^2$. This means that $\max_i |s_i x_i - \eta| \le \frac{\eta}{4\sqrt{n}}$, which proves the claim. □

Claim 20.6 (Numerator). $\sum_{i=1}^n (\eta' - s_i x_i)^2 \le \eta^2/8$.

Proof. Let $\delta = \frac{1}{4\sqrt{n}}$. Then,

$$\begin{aligned}
\sum_{i=1}^n (\eta' - s_i x_i)^2 &= \sum_{i=1}^n ((1-\delta)\eta - s_i x_i)^2 \\
&= \sum_{i=1}^n (\eta - s_i x_i)^2 + \sum_{i=1}^n (\delta\eta)^2 + 2\delta\eta \sum_{i=1}^n (\eta - s_i x_i).
\end{aligned}$$

The first term is at most $(\eta/4)^2$, by the induction hypothesis. On the other hand, by Claim 20.3 we have

$$\sum_{i=1}^n (\eta - s_i x_i) = n\eta - \sum_{i=1}^n s_i x_i = 0.$$

Thus

$$\sum_{i=1}^n (\eta' - s_i x_i)^2 \le (\eta/4)^2 + n\frac{1}{(4\sqrt{n})^2}\eta^2 = \eta^2/8.$$ □

Substituting these results into (20.8), we get

$$\sqrt{\sum_{i=1}^n (\Delta s_i \Delta x_i)^2} \le \frac{1}{4}\frac{\eta^2/8}{(1 - \frac{1}{4\sqrt{n}})\eta} = \frac{1}{32}\frac{\eta'}{(1 - \frac{1}{4\sqrt{n}})^2}.$$

This expression is smaller than $\eta'/4$, which completes the proof. □

Lemma 20.7. *The new values x', s' are non-negative.*

Proof. By induction, we assume the previous point has $x_i > 0$ and $s_i > 0$. (For the base case we need to ensure that the starting solution $(x^{(0)}, s^{(0)})$ also satisfies this property.) Now for a scalar $\alpha \in [0,1]$ we define $x'' := x + \alpha\Delta x$, $s'' := s + \alpha\Delta s$, and $\eta'' := (1-\alpha)\eta + \alpha\eta'$, to linearly interpolate between the old values and the new ones. Then we can show $\langle x'', s''\rangle = n\,\eta''$, and also

$$\sum_i (s_i'' x_i'' - \eta'')^2 \leq (\eta''/4)^2, \tag{20.9}$$

which are analogs of Lemmas 20.3 and 20.4 respectively. The latter inequality means that $|s_i'' x_i'' - \eta''| \leq \eta''/4$ for each coordinate i, else that coordinate itself would violate inequality (20.9). Specifically, this means that neither x_i'' nor s_i'' ever becomes zero for any value of $\alpha \in [0,1]$. Now since (x_i'', s_i'') is a linear interpolation between (x_i, s_i) and (x_i', s_i'), and the former were strictly positive, the latter cannot be non-positive. □

Theorem 20.8. *Given an LP* $\min\{c^\intercal x \mid Ax = b, x \geq 0\}$ *with an initial feasible* $(x^{(0)}, \eta_0)$ *pair, the interior-point algorithm produces a primal-dual pair with duality gap at most* ε *in* $O(\sqrt{n}\log\frac{n\eta_0}{\varepsilon})$ *iterations, each involving solving one linear system.*

The proof of the above theorem follows immediately from the fact that the duality gap at the beginning is $n\eta_0$, and the value of η drops by $(1 - \frac{1}{4\sqrt{n}})$ in each iteration. If the LP has representation size $L := \langle A\rangle + \langle b\rangle + \langle c\rangle$, we can stop when $\varepsilon = \exp(-\operatorname{poly}(L))$, and then round this solution to an vertex solution of the LP.

The one missing piece is finding the initial $(x^{(0)}, \eta_0)$ pair: this is a somewhat non-trivial step. One possible approach is to run the interior-point algorithm "in reverse". The idea is that we can start with some vertex of the feasible region, and then to successively *increase* η through a similar mechanism as the one above, until the value of η is sufficiently large to begin the algorithm.

20.3 The Newton-Raphson Method

A more "modern" way of viewing interior-point methods is via the notion of *self-concordance*. To do this, let us revisit the classic Newton-Raphson method for finding roots.

20.3.1 Finding Zeros of Functions

The basic Newton-Raphson method for finding a zero of a univariate function is the following: given a function g, we start with a point x_1, and at each time t, set

$$x_{t+1} \leftarrow x_t - \frac{g(x_t)}{g'(x_t)}. \tag{20.10}$$

We now show that if f is "nice enough" and we are "close enough" to a zero x^*, then this process converges very rapidly to x^*.

Since we take $O(\log\log 1/\varepsilon)$ iterations to get to error ε, the number of bits of accuracy squares each time. This is called ***quadratic convergence*** in the optimization literature.

Theorem 20.9. *Suppose g has continuous second-derivatives, then if x^* is a zero of g, then if we start at x_1 "close enough" to X^*, the error goes to ε in $O(\log\log 1/\varepsilon)$ steps. Make this formal!*

Proof. By Taylor's theorem, the existence of continuous second derivatives means we can approximate f around x_t as:

$$f(x^*) = f(x_t) + f'(x_t)(x^* - x_t) + 1/2 f''(\xi_t)(x^* - x_t)^2,$$

where ξ_t is some point in the interval $[x^*, x_t]$. However, x^* is a zero of f, so $f(x^*) = 0$. Moreover, using (20.10) to replace $x_t f'(x_t) - f(x_t)$ by $x_{t+1} f'(x_t)$, and rearranging, we get

$$\underbrace{x^* - x_{t+1}}_{=:\delta_{t+1}} = \frac{-f''(\xi_t)}{2\, f'(x_t)} \cdot \underbrace{(x^* - x_t)^2}_{=:\delta_t^2}.$$

Above, we use δ_t to denote the error $x^* - x_t$. Taking absolute values

$$|\delta_{t+1}| = \left|\frac{f''(\xi_t)}{2\, f'(x_t)}\right| \cdot \delta_t^2.$$

Hence, if we can ensure that $|\frac{f''(\xi)}{2\, f'(x)}| \leq M$ for each x and each ξ that lies bewteen x^* and x, then once we have δ_0 small enough, then each subsequent error drops quadratically. This means the number of significant bits of accuracy double each step. More careful analysis. □

20.3.2 An Example

Given an n-bit integer $a \in \mathbb{Z}$, suppose we want to compute its reciprocal $1/a$ without using divisions. This reciprocal is a zero of the expression

$$g(x) = 1/x - a.$$

Hence, the Newton-Raphson method says, we can start with $x_1 = 1$, say, and then use (20.10) to get

$$x_{t+1} \leftarrow x_t - \frac{(1/x_t - a)}{(-1/x_t^2)} = x_t + x_t(1 - a\, x_t) = 2x_t - a\, x_t^2.$$

This method for computing reciprocals appears in the classic book of Aho, Hopcroft, Ullman, without any elaboration—it always mystified me until I realized the connection to the Newton-Raphson method. I guess they expected their readers to be familiar with these connections, since computer science used to have closer connections to numerical analysis in the 1970s.

If we define $\varepsilon_t := 1 - a\, x_t$, then

$$\varepsilon_{t+1} = 1 - a\, x_{t+1} = 1 - (2a\, x_t - a^2\, x_t^2) = (1 - a\, x_t)^2 = \varepsilon_t^2.$$

Hence, if $\varepsilon_1 \leq 1/2$, say, the number of bits of accuracy double at each step. Moreover, if we are careful, we can store x_t using integers (by instead keeping track of $2^k x_t$ for suitably chosen values $k \approx 2^t$).

20.3.3 *Minimizing Convex Functions*

To find the minimum of a function f (especially a convex function) we can focus on finding a stationary point, i.e., a point x such that $f'(x) = 0$. Setting $g = f'$, the update rule just changes to

$$x_{t+1} \leftarrow x_t - \frac{f'(x_t)}{f''(x_t)}. \tag{20.11}$$

20.3.4 *On To Higher Dimensions*

For general functions $f : \mathbb{R}^n \to \mathbb{R}$, the rule remains the same, with the natural changes:

$$x_{t+1} \leftarrow x_t - [H_f(x_t)]^{-1} \cdot \nabla f(x_t). \tag{20.12}$$

20.4 *Self-Concordance*

Analogy between self-concordance and the convergence conditions for the 1-d case?

Present the view using the "modern view" of self-concordance. Mention that the current bound is really $O(m)$-self-concordant. That universal barrier is $O(n)$ self-concordant, but not efficient. Vaidya's volumetric barrier? The entropic barrier? The Lee-Sidford barrier, based on leverage scores. What's the cleanest way, without getting lost in the algebra?