In the previous lecture, in order to find \( \min_{x \in \mathbb{R}^n} f(x) \), assuming that \( f(x) \) has continuous first and second derivatives, we used Taylor approximation.

\[
f(x + \delta) = f(x) + \nabla f(x)^T \delta + \delta^T \nabla^2 f(y) \delta \quad \text{where } y \in [x, x + \delta]
\]

We assumed that we have query access to \( f(x) \) and \( \nabla f(x) \). We also considered the following bound assumptions on \( \nabla^2 f \):

1. \( \delta^T \nabla^2 f(y) \delta \leq \beta \| \delta \|^2 \) or, equivalently, \( \lambda_{\text{max}}(\nabla^2) \leq \beta \), which implies that the progress is at least \( \frac{1}{2\beta} \| \nabla f(x) \|^2 \) at every step;

2. \( f \) is convex, which means that \( \delta^T \nabla^2 f(y) \delta \geq 0 \) and that if \( \nabla f = 0 \) we have reached optimality;

3. \( \delta^T \nabla^2 f(y) \delta \geq \alpha \| \delta \|^2 \)

Convergence occurs in \( O\left(\frac{\beta}{\alpha} \log \frac{f(x^0) - f(x^*)}{\epsilon}\right) \) where \( \beta \) is the biggest eigenvalue and \( \alpha \) is the smallest eigenvalue of \( \nabla^2 f \), and \( x^* \) is the optimal solution. We are looking for \( x^T \) such that:

\[
f(x^T) - f(x^*) \leq \epsilon
\]

Define the condition number \( k = \frac{\beta}{\alpha} \).

1 Newton’s Method

Define \( Q = \delta^T \nabla^2 f(y) \delta \). Using linear changes of variables, we have:

\[
z := Ax \quad \text{where } A \text{ is a full rank } n \times n \text{ matrix}
\]

\[
\Delta := A\delta \implies \delta = A^{-1} \Delta
\]

\[
Q = \Delta^T (A^{-1})^T \nabla^2 f(y) A^{-1} \Delta
\]

We want to set \( A \) such that:

\[
(A^{-1})^T \nabla^2 f(y) A^{-1} = I
\]

since then \( \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}} = 1 \). Therefore,

\[
\nabla^2 f(y) = A^T A \implies A = (\nabla^2 f(y))^{\frac{1}{2}}
\]
Now, fix $A$. We look for a step which is:

$$\arg\min_{\delta : \Delta = A\delta, \|\Delta\| = \epsilon} \nabla f(x)^T \delta + \frac{1}{2} \|\Delta\|^2$$

$$= \arg\min_{\delta : \Delta = A\delta, \|\Delta\| = \epsilon} \nabla f(x)^T A^{-1} \Delta$$

$$= -\eta A^{-1}(A^{-1})^T \nabla f(x)$$

$$= -\eta (\nabla^2 f(y))^{-1} \nabla f(x)$$

Because the minimum is achieved for $\Delta \propto - (\nabla f(x)^T A^{-1})^T = -(A^{-1})^T \nabla f(x)$. Therefore, the minimization occurs at step $\delta = -\eta (\nabla^2 f(y))^{-1} \nabla f(x)$.

**Note 1.** We need query access to $\nabla^2 f(y)$, which is why this is called a second-order method.

**Note 2.** We need to invert a matrix, or equivalently a linear system of equations: $\nabla^2 f(y) \delta = -\eta \nabla f(x)$.

**Note 3.** We don’t have $y$, which is why Newton’s method uses $\delta = -\eta (\nabla^2 f(x))^{-1} \nabla f(x)$. But in general, $\nabla^2 f(x) \neq \nabla^2 f(y)$.

**Note 4.** Assuming that $\nabla^2 f(x) = \nabla^2 f(y)$, convergence takes $O((\log \log \|x^0 - x^*\|) / \epsilon)$.

### 1.1 Alternative view on Newton’s method

$$f(x + \delta) = \underbrace{f(x) + \nabla f(x)^T \delta + \delta^T \nabla^2 f(x) \delta}_{\delta \text{ is minimizer of}} + O(|\delta|^3)$$

**Theorem 1.** Suppose there exists $r > 0$ such that for all $x, y$ at distance $\leq r$ from $x^*$ we have:

1. $\lambda_{\min}(\nabla^2 f(x)) \geq \mu$

2. $\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L \|x - y\|$

Then $\|x - x^*\| \leq \frac{L}{\mu \|x^0 - x^*\|^2}$, where $x^0$ is at distance $\leq r$ from $x^*$ and $x^1$ is $x^0$ plus a Newton’s step.

The norm we use for matrices is the spectral norm, i.e., $\|X\| = \lambda_{\max}(X)$.

**Intuition:** under the right conditions, it converges in $O((\log \log \|x^0 - x^*\|) / \epsilon)$.

### 2 Back to linear programming

#### 2.1 The interior point method

Consider a linear programming problem of the following form:

$$\min c^T x$$

s.t. $Ax \leq b$

on $n$ coordinates with $m$ constraints. Call $K$ the feasible region, i.e., $K = \{x \in \mathbb{R}^n \mid Ax \leq b\}$.

We have already seen one way to turn this into an unconstrained problem, by replacing the objective function with one that evaluates to $c^T x$ for $x \in K$ and to $+\infty$ otherwise. But such a function isn’t
continuous and doesn’t work well with the gradient descent method or Newton’s method. We need a smoother function.

We will instead replace the objective function with \( f_\eta(x) = \eta c^T x + F(x) \), for \( \eta \geq 0 \), where \( F(x) \) is called a barrier function and has the following properties:

\[
F(x) < +\infty \text{ for } x \in K \\
F(x) \to +\infty \text{ for } x \to \partial K
\]

One possible barrier function is:

\[
F(x) = \log \left( \prod_{i=1}^{m} \frac{1}{b_i - A_i x} \right) = -\sum_{i=1}^{m} \log (b_i - A_i x)
\]

Call \( x^*_\eta = \arg \min \eta c^T x + F(x) \). It is a continuous function of \( \eta \). When \( \eta = 0 \) we have that \( x^*_0 \) is independent of \( c \), and this point is called analytic center.

In the above drawing we see the polytope \( K \) with \( c \) pointing from left to right. The optimal point \( x^* \) is therefore the leftmost vertex of \( K \). The point \( x^*_0 \) is the analytic center. The path connecting the two is the central path, i.e., \( \{x^*_\eta, \eta \geq 0\} \). This means that \( x^* \) is \( \lim_{\eta \to +\infty} x^*_\eta \).

This reformulation of linear programming leads to a few algorithm ideas:

**Idea 1**

- start from a point \( x^0 \);
- compute \( x^*_\eta \) for a “very large” \( \eta \) using Newton’s method or gradient descent starting at \( x^0 \).

The problem with gradient descent is that it depends on the condition number, which depends on \( F(x) \) and may be very large, whereas Newton’s method requires \( x^0 \) to be “close” to \( x^*_\eta \) in order for the theorem we saw earlier to apply.
Let $s_i$ be $b_i - A_ix$, and call these slack variables. We can use them to express $\nabla f_\eta(x)$ and $\nabla^2 f_\eta(x)$:

$$
\nabla f_\eta(x) = \eta c + \sum_{i=1}^{m} \frac{A_i}{s_i(x)}
$$

$$
\nabla^2 f_\eta(x) = \nabla^2 F(x) = \sum_{i=1}^{m} \frac{A_i A_i^T}{s_i^2(x)}
$$

This means that close to the boundary of $K$ the coefficients of the Hessian of the barrier function will increase rapidly and this may affect negatively the condition number.

Remark: we assume $K$ has $> 0$ volume.

**Idea 2**

- start at $x^0 = x^*_{\eta_0}$ for some $\eta_0 > 0$;
- “walk the central path”, meaning that at time $t + 1$:
  - increase $\eta$: $\eta_{t+1} = \eta_t (1 + \alpha)$ (we will decide the value of $\alpha$ later);
  - run Newton’s method to find $x^*_{\eta_{t+1}}$ starting at $x^*_{\eta_t}$ (which works correctly and efficiently as long as $x^*_{\eta_{t+1}}$ is “close” to $x^*_{\eta_t}$).

**Idea 3** This idea is just a performance improvement of idea 2, based on the observation that when running Newton’s method to find $x^*_{\eta_{t+1}}$ we don’t need to run it until it reaches optimality, we can stop it early. In particular stopping it after just one iteration yields the following algorithm:

**Algorithm**

- start at $x_0 \approx x^*_{\eta_0}$ for some $\eta_0 > 0$;
- at step $t + 1$ define $\eta_{t+1}$ as $\eta_t (1 + \alpha)$ and find $x_{t+1}$ by performing one step of Newton’s method for $f_{\eta_{t+1}}$ starting at $x_t$;
- once at time $t = T$ such that $\eta_T$ is “large enough” run Newton’s method to optimality and obtain $x^*_{\eta_T}$;
- output $x^*_{\eta_T}$.

**Lemma 2.** For all $\eta$ we have $c^T x^*_\eta - c^T x^* \leq \frac{m}{\eta}$, which implies that $\eta_T$ has to be larger than $\frac{m}{\epsilon}$ if we want $c^T x^*_\eta - c^T x^* \leq \epsilon$. This means that the number of steps is

$$
T = O \left( \frac{1}{\alpha} \log \frac{m/\epsilon}{\eta_0} \right) = O \left( \log_{1+\alpha} \frac{m/\epsilon}{\eta_0} \right)
$$