ETH Zurich

Advanced Graph Algorithms and Optimization

Rasmus Kyng & Maximilian Probst Gutenberg

Spring 2025

These notes will be updated throughout the course. They are likely to contain typos, and they may have mistakes or lack clarity in places. Feedback and comments are welcome. Please send to kyng@inf.ethz.ch or, even better, submit a pull request at

`https://github.com/mesimon/agao25_script`.

We want to thank scribes from the 2020 edition of the course who contributed to these notes: Hongjie Chen, Meher Chaitanya, Timon Knigge, and Tim Taubner – and we're grateful to all the readers who've submitted corrections, including Martin Kucera, Alejandro Cassis, and Luke Volpatti.

A important note: If you're a student browsing these notes to decide whether to take this course, please note that the current notes are incomplete. We will release parts later in the semester. You can take a look at last year's notes for an impression of the what the rest of the course will look like. Find them here:

`https://github.com/rjkyng/agao24_script/raw/main/agao24_script.pdf`

There will, however, be some changes to the content compared to last year.

Contents

Chapter 1

Course Introduction

1.1 Overview

This course will take us quite deep into modern approaches to graph algorithms using convex optimization techniques. By studying convex optimization through the lens of graph algorithms, we'll try to develop an understanding of fundamental phenomena in optimization. Much of our time will be devoted to flow problems on graphs. We will not only be studying these problems for their own sake, but also because they often provide a useful setting for thinking more broadly about optimization.

The course will cover some traditional discrete approaches to various graph problems, especially flow problems, and then contrast these approaches with modern, asymptotically faster methods based on combining convex optimization with spectral and combinatorial graph theory.

1.2 Electrical Flows and Voltages - a Graph Problem from Middle School?

We will dive right into graph problems by considering how electrical current moves through a network of resistors.

First, let us recall some middle school physics. If some of these things don't make sense to you, don't worry, in less than a paragraph from here, we'll be back to safely doing math.

Recall that a typical battery that one buys from Migros has two endpoints, and produces what is called a *voltage difference* between these endpoints.

One end of the battery will have a positive charge (I think that means an excess of positrons[1]), and the other a negative charge. If we connect the two endpoints with a wire, then a current will flow from one end of the battery to the other in an attempt to even out this imbalance of charge.

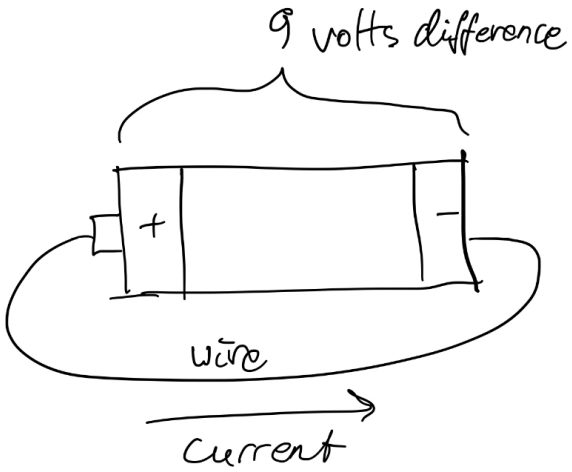

Figure 1.1: A 9 volts battery with a wire attached.

We can also imagine a kind of battery that tries to send a certain amount of current through the wires between its endpoints, e.g. 1 unit of charge per unit of time. This will be a little more convenient to work with, so let us focus on that case.

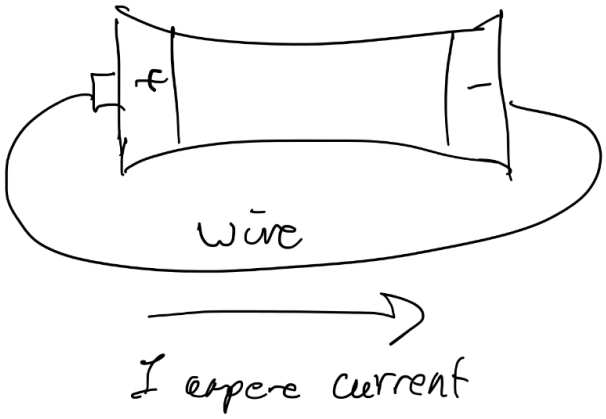

Figure 1.2: A 1 ampere battery with a wire attached.

A *resistor* is a piece of wire that connects two points u and v, and is completely described by a single number r called its *resistance*.

[1] I'm joking, of course! Try Wikipedia if you want to know more. However, you will not need it for this class.

If the voltage difference between the endpoints of the resistor is x, and the resistance is r then this will create a flow of charge per unit of time of $f = x/r$. This is called Ohm's Law.

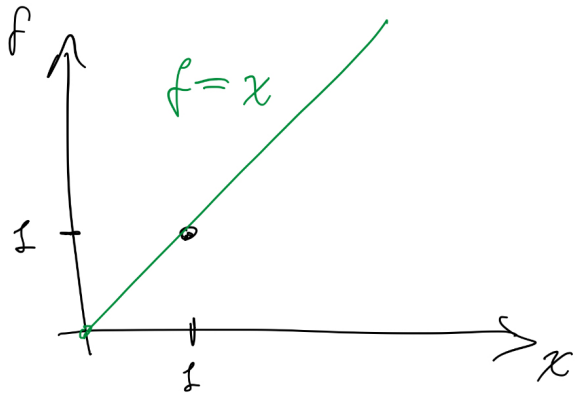

Figure 1.3: Ohm's Law for a resistor with resistance $r = 1$.

Suppose we set up a bunch of wires that route electricity from our current source s to our current sink t in some pattern:

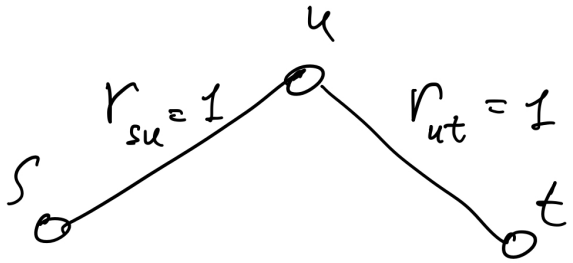

Figure 1.4: A path of two resistors.

We have one unit of charge flowing out of s per unit of time, and one unit coming into t. Because charge is conserved, the current flowing into any other point u must equal the amount flowing out of it. This is called Kirchhoff's Current Law.

To send one unit of current from s to t, we must be sending it first from s to u and then from u to t. So the current on edge (s, u) is 1 and the current on (u, t) is 1. By Ohm's Law, the voltage difference must also be 1 across each of the two wires. Thus, if the voltage is x at s, it must be $x + 1$ at u and $x + 2$ at t. What is x? It turns out it doesn't matter: We only care about the differences. So let us set $x = 0$.

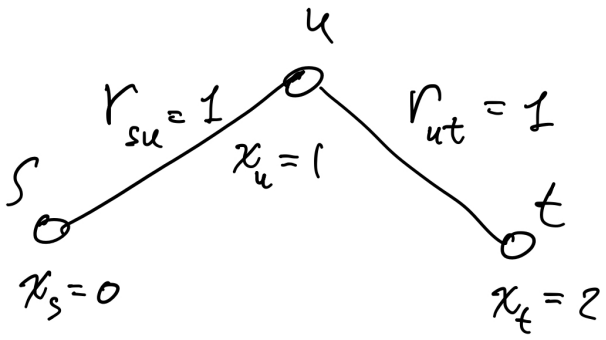

Figure 1.5: A path of two resistors.

Let us try one more example:

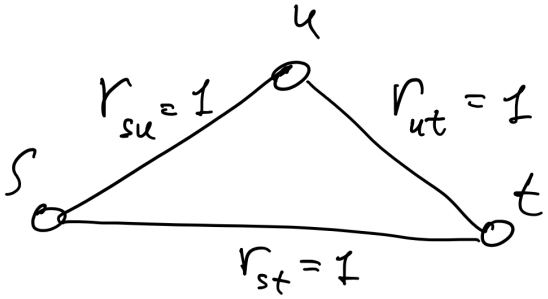

Figure 1.6: A network with three resistors.

How much flow will go directly from s to t and how much via u?

Well, we know what the net current flowing into and out of each vertex must be, and we can use that to set up some equations. Let us say the voltage at s is x_s, at u is x_u and at t is x_t.

- Net current at s: $-1 = (x_s - x_t) + (x_s - x_u)$
- Net current at u: $0 = (x_u - x_s) + (x_u - x_t)$
- Net current at t: $1 = (x_t - x_s) + (x_t - x_u)$

The following is a solution: $x_s = 0$, $x_u = \frac{1}{3}$, $x_t = \frac{2}{3}$. And as before, we can shift all the voltages by some constant x and get another solution $x_s = x + 0$, $x_u = x + \frac{1}{3}$, $x_t = x + \frac{2}{3}$. You might want to convince yourself that these are the only solutions.

Electrical flows in general graphs. Do we know enough to calculate the electrical flow in some other network of resistors? To answer this, let us think about the network as a graph. Consider an undirected graph $G = (V, E)$ with $|V| = n$ vertices and $|E| = m$ edges, and let us assume G is connected. Let's associate a resistance $\boldsymbol{r}(e) > 0$ with every edge $e \in E$.

To keep track of the direction of the flow on each edge, it will be useful to assign an arbitrary direction to every edge. So let's do that, but remember that this is just a bookkeeping tool that helps us track where flow is going.

A *flow* in the graph is a vector $\boldsymbol{f} : \mathbb{R}^E$. The *net flow* of $\boldsymbol{f}$ at a vertex $u \in V$ is defined as $\sum_{v \to u} \boldsymbol{f}(v,u) - \sum_{u \to v} \boldsymbol{f}(u,v)$.

We say a flow routes the demands $\boldsymbol{d} \in \mathbb{R}^V$ if the net flow at every vertex v is $\boldsymbol{d}(v)$.

We can assign a voltage to every vertex $\boldsymbol{x} \in R^V$. Ohm's Law says that the electrical flow induced by these voltages will be $\boldsymbol{f}(u,v) = \frac{1}{\boldsymbol{r}(u,v)}(\boldsymbol{x}(v) - \boldsymbol{x}(u))$.

Say we want to route one unit of current from vertex $s \in V$ to vertex $t \in V$. As before, we can write an equation for every vertex saying that the voltage differences must produce the desired net current:

- Net current at s: $-1 = \sum_{(s,v)} \frac{1}{\boldsymbol{r}(s,v)}(\boldsymbol{x}(v) - \boldsymbol{x}(s))$
- Net current at $u \in V \setminus \{s,t\}$: $0 = \sum_{(u,v)} \frac{1}{\boldsymbol{r}(u,v)}(\boldsymbol{x}(v) - \boldsymbol{x}(u))$
- Net current at t: $1 = \sum_{(t,v)} \frac{1}{\boldsymbol{r}(t,v)}(\boldsymbol{x}(v) - \boldsymbol{x}(t))$

This gives us n constraints, exactly as many as we have voltage variables. However we have to be a little careful when trying to conclude that a solution exists, yielding voltages $\boldsymbol{x}$ that gives induce an electrical flow routing the desired demand.

You will prove in the exercises (Week 1, Exercise 2) that a solution $\boldsymbol{x}$ exists. The proof requires two important observations: Firstly that the graph is connected, and secondly that summed over all vertices, the net demand is zero, i.e. as much flow is coming into the network as is leaving it.

The incidence matrix and the Laplacian matrix. To have a more compact notation for net flow constraints, we also introduce the *edge-vertex incidence matrix* of the graph, $\boldsymbol{B} \in \mathbb{R}^{V \times E}$.

$$\boldsymbol{B}(v,e) = \begin{cases} 1 & \text{if } e = (u,v) \\ -1 & \text{if } e = (v,u) \\ 0 & \text{o.w.} \end{cases}$$

Now we can express the net flow constraint that $\boldsymbol{f}$ routes $\boldsymbol{d}$ by

$$\boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}.$$

This is also called a conservation constraint. In our examples so far, we have $\boldsymbol{d}(s) = -1$, $\boldsymbol{d}(t) = 1$ and $\boldsymbol{d}(u) = 0$ for all $u \in V \setminus \{s,t\}$.

If we let $\boldsymbol{R} = \mathrm{diag}_{e \in E}\, \boldsymbol{r}(e)$ then Ohm's law tells us that $\boldsymbol{f} = \boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{x}$. Putting these observations together, we have $\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{x} = \boldsymbol{d}$. The voltages $\boldsymbol{x}$ that induce $\boldsymbol{f}$ must solve this system of linear equations, and we can use that to compute both $\boldsymbol{x}$ and $\boldsymbol{f}$. It is exactly

the same linear equation as the one we considered earlier. We can show that for a connected graph, a solution $\boldsymbol{x}$ exists if and only if the flow into the graph equals the net flow out, which we can express as $\sum_v \boldsymbol{d}(v) = 0$ or $\mathbf{1}^\top \boldsymbol{d} = 0$. You will show this as part of Exercise 2. This also implies that an electrical flow routing $\boldsymbol{d}$ exists if and only if the net flow into the graph equals the net flow out, which we can express as $\mathbf{1}^\top \boldsymbol{d} = 0$.

The matrix $\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top$ is called the *Laplacian* of the graph and is usually denoted by $\boldsymbol{L}$.

An optimization problem in disguise. So far, we have looked at electrical voltages and flows as arising from a set of linear equations – and it might not be apparent that this has anything to do with optimization. But transporting current through a resistor requires energy, which will be dissipated as heat by the resistor (i.e. it will get hot!). If we send a current of f across a resistor with a potential drop of x, then the amount of energy spent per unit of time by the resistor will be $f \cdot x$. This is called Joule's Law. Applying Ohm's law to a resistor with resistance r, we can also express this energy per unit of time as $f \cdot x = x^2/r = r \cdot f^2$. Since we aren't bothering with units, we will even forget about time, and refer to these quantities as "energy", even though a physicist would call them "power".

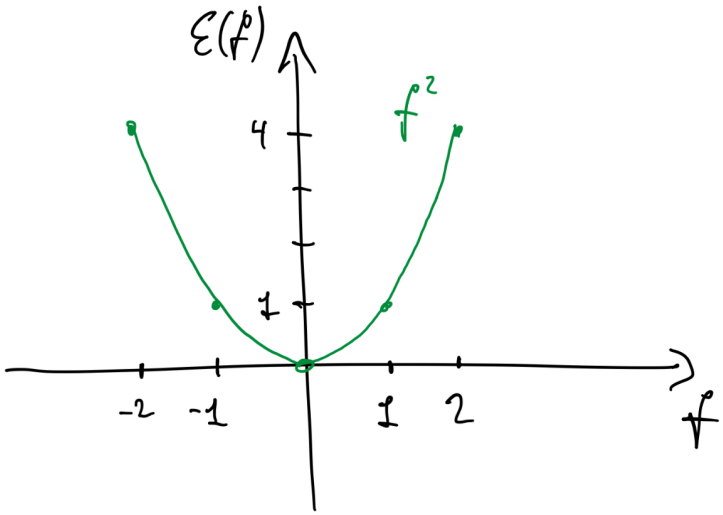

Figure 1.7: Energy as a function of flow in a resistor with resistance $r = 1$.

Now, another interesting question would seem to be: If we want to find a flow routing a certain demand $\boldsymbol{d}$, how should the flow behave in order to minimize the electrical energy spent routing the flow? The electrical energy of a flow vector $\boldsymbol{f}$ is $\mathcal{E}(\boldsymbol{f}) \stackrel{\text{def}}{=} \sum_e \boldsymbol{r}(e)\boldsymbol{f}(e)^2$. We can phrase this as an optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{f} \in \mathbb{R}^E} \mathcal{E}(\boldsymbol{f}) \\ &\text{s.t. } \boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}. \end{aligned}$$

We call this problem *electrical energy-minimizing flow*. As we will prove later, the flow $\boldsymbol{f}^*$ that minimizes the electrical energy among all flows that satisfy $\boldsymbol{Bf} = \boldsymbol{d}$ is precisely the electrical flow.

A pair of problems. What about our voltages, can we also get them from some optimization problem? Well, we can work backwards from the fact that our voltages solve the equation $\boldsymbol{Lx} = \boldsymbol{d}$. Consider the function $c(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Lx} - \boldsymbol{x}^\top \boldsymbol{d}$. We should ask ourselves some questions about this function $c : \mathbb{R}^V \to \mathbb{R}$. Is it continuous and continuously differentiable? The answer to this is yes, and that is not hard to see. Does the function have a minimum? This is maybe not immediately clear, but the minimum does indeed exist.

When this is minimized, the derivative of $c(\boldsymbol{x})$ with respect to each coordinate of $\boldsymbol{x}$ must be zero. This condition yields exactly the system of linear equations $\boldsymbol{Lx} = \boldsymbol{d}$. You will confirm this in Exercise 4 of the first exercise sheet.

Based on our derivative condition for the optimum, we can also express the electrical voltages as the solution to an optimization problem, namely

$$\min_{\boldsymbol{x} \in \mathbb{R}^V} c(\boldsymbol{x})$$

As you are probably aware, having the derivative of each coordinate equal zero is not a sufficient condition for being at the optimum of a function[2].

It is also interesting to know whether *all* solutions to $\boldsymbol{Lx} = \boldsymbol{d}$ are in fact minimizers of c. The answer is yes, and we will see some very general tools for proving statements like this in Chapter 2.

Altogether, we can see that routing electrical current through a network of resistors leads to a *pair* of optimization problems, let's call them $\boldsymbol{f}^*$ and $\boldsymbol{x}^*$, and that the solutions to the two problems are related, in our case through the equation $\boldsymbol{f}^* = \boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{x}^*$ (Ohm's Law). But why and how are these two optimization problems related?

Instead of minimizing $c(\boldsymbol{x})$, we can equivalently think about maximizing $-c(\boldsymbol{x})$, which gives the following optimization problem: $\max_{\boldsymbol{x} \in \mathbb{R}^V} -c(\boldsymbol{x})$. In fact, as you will show in the exercises for Week 1, we have $\mathcal{E}(\boldsymbol{f}^*) = -c(\boldsymbol{x}^*)$, so the minimum electrical energy is exactly the maximum value of $-\boldsymbol{c}(\boldsymbol{x})$. More generally for *any* flow that routes $\boldsymbol{d}$ and *any* voltages $\boldsymbol{x}$, we have $\mathcal{E}(\boldsymbol{f}) \geq -c(\boldsymbol{x})$. So, for any $\boldsymbol{x}$, the value of $-c(\boldsymbol{x})$ is a lower bound on the minimum energy $\mathcal{E}(\boldsymbol{f}^*)$.

This turns out to be an instance of a much broader phenomenon, known as Lagrangian duality, which allows us to learn a lot about many optimization problems by studying two related pairs of problems, a minimization problem, and a related maximization problem that gives lower bounds on the optimal value of the minimization problem.

[2]Consider the function in one variable $c(x) = x^3$.

Solving $\boldsymbol{Lx} = \boldsymbol{d}$. Given a graph G with resistances for the edges, and some net flow vector $\boldsymbol{d}$, how quickly can we compute $\boldsymbol{x}$? Broadly speaking, there are two very different families of algorithms we could use to try to solve this problem.

We could solve the linear equation using something like *Gaussian Elimination* to compute an exact solution.

Alternatively, we could start with a guess at a solution, e.g. $\boldsymbol{x}_0 = \mathbf{0}$, and then we could try to make a change to $\boldsymbol{x}_0$ to reach a new point $\boldsymbol{x}_1$ with a lower value of $c(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} - \boldsymbol{x}^\top \boldsymbol{d}$, i.e. $c(\boldsymbol{x}_1) < c(\boldsymbol{x}_0)$. If we repeat a process like that for enough steps, say t, hopefully we eventually reach $\boldsymbol{x}_t$ with $c(\boldsymbol{x}_t)$ close to $c(\boldsymbol{x}^*)$, where $\boldsymbol{x}^*$ is a minimizer of $c(\boldsymbol{x})$ and hence $\boldsymbol{L}\boldsymbol{x}^* = \boldsymbol{d}$. Now, we also need to make sure that $c(\boldsymbol{x}_t) \approx c(\boldsymbol{x}^*)$ implies that $\boldsymbol{L}\boldsymbol{x}_t \approx \boldsymbol{d}$ in some useful sense.

One of the most basic algorithms in this framework of "guess and adjust" is called *Gradient Descent*, which we will study in Week 2. The rough idea is the following: if we make a very small step from $\boldsymbol{x}$ to $\boldsymbol{x} + \boldsymbol{\delta}$, then a multivariate Taylor expansion suggests that $c(\boldsymbol{x} + \boldsymbol{\delta}) - c(\boldsymbol{x}) \approx \sum_{v \in V} \boldsymbol{\delta}(v) \frac{\partial c(\boldsymbol{x})}{\partial \boldsymbol{x}(v)}$.

If we are dealing with smooth convex function, this quantity is negative if we let $\boldsymbol{\delta}(v) = -\epsilon \cdot \frac{\partial c(\boldsymbol{x})}{\partial \boldsymbol{x}(v)}$ for some small enough ϵ so the approximation holds well. So we should be able to make progress by taking a small step in this direction. That's Gradient Descent! The name comes from the vector of partial derivatives, which is called the gradient.

As we will see later in this course, understanding electrical problems from an optimization perspective is crucial to develop fast algorithms for computing electrical flows and voltages, but to do very well, we also need to borrow some ideas from Gaussian Elimination.

What running times do different approaches get?

1. Using Gaussian Elimination, we can find $\boldsymbol{x}$ s.t. $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{d}$ in $O(n^3)$ time and with asymptotically faster algorithms based on matrix multiplication, we can bring this down to roughly $O(n^{2.372})$.

2. Meanwhile Gradient Descent will get a running time of $O(n^3 m)$ or so – at least this is a what a simple analysis suggests.

3. However, we can do much better: By combining ideas from both algorithms, and a bit more, we can get $\boldsymbol{x}$ up to very high accuracy in time $O(m \log^c n)$ where c is some small constant.

1.3 Convex Optimization

Recall our plot in Figure 1.7 of the energy required to route a flow f across a resistor with resistance r, which was $\mathcal{E}(f) = r \cdot f^2$. We see that the function has a special structure: the

graph of the function sits below the line joining any two points $(f, \mathcal{E}(f))$ and $(g, \mathcal{E}(g))$. A function $\mathcal{E} : \mathbb{R} \to \mathbb{R}$ that has this property is said to be convex.

Figure 1.8 shows the energy as a function of flow, along with two points $(f, \mathcal{E}(f))$ and $(g, \mathcal{E}(g))$. We see the function sits below the line segment between these points.

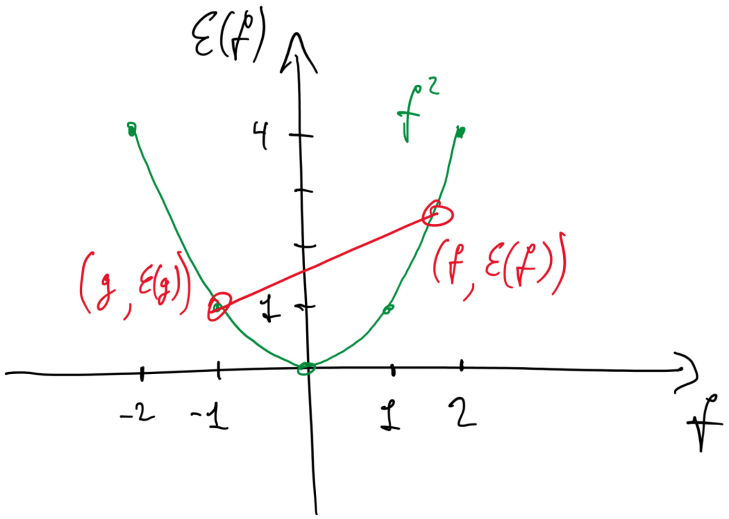

Figure 1.8: Energy as a function of flow in a resistor with resistance $r = 1$. The function is convex.

We can also interpret this condition as saying that for all $\theta \in [0, 1]$

$$\mathcal{E}(\theta f + (1 - \theta)g) \leq \theta \mathcal{E}(f) + (1 - \theta)\mathcal{E}(g).$$

This immediately generalizes to functions $\mathcal{E} : \mathbb{R}^m \to \mathbb{R}$.

A *convex set* is a subset of $S \subseteq \mathbb{R}^m$ s.t. if $\boldsymbol{f}, \boldsymbol{g} \in S$ then for all $\theta \in [0, 1]$ we have $\theta \boldsymbol{f} + (1-\theta)\boldsymbol{g} \in S$.

Figure 1.9 shows some examples of sets that are and aren't convex.

Convex functions and convex sets are central to optimization, because for most problems of minimization of a convex function over a convex set, we can develop fast algorithms [3].

So why convex functions and convex sets? One important reason is that for a convex function defined over a convex feasible set, any local minimum is also a global minimum, and this fact makes searching for an optimal solution computationally easier. In fact, this is closely related to why Gradient Descent works well on many convex functions.

Notice that the set $\{\boldsymbol{f} : \boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}\}$ is convex, i.e. the set of all flows that route a fixed demand

[3] There are some convex optimization problems that are NP-hard. That said, polynomial time algorithms exist for almost any convex problem you can come up with. The most general polynomial time algorithm for convex optimization is probably the Ellipsoid Method.

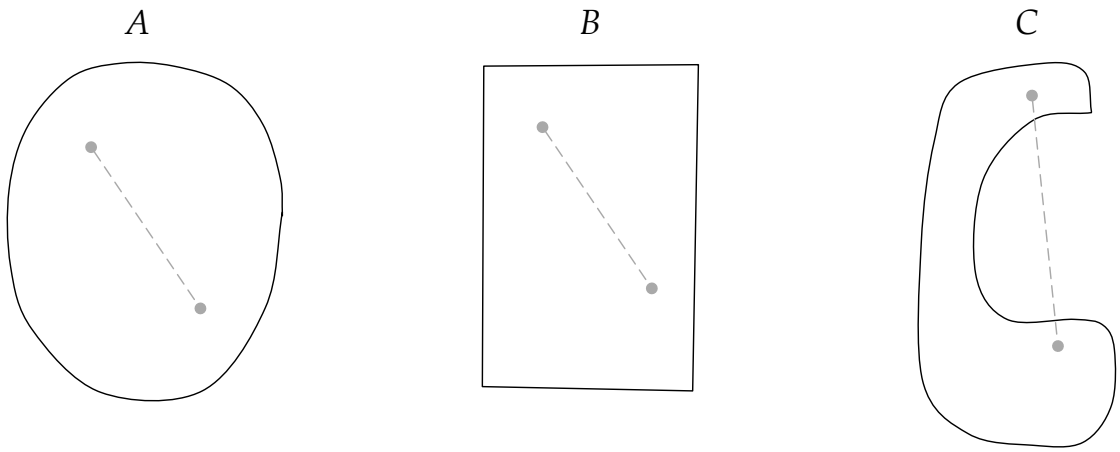

Figure 1.9: A depiction of convex and non-convex sets. The sets A and B are convex since the straight line between any two points inside them is also in the set. The set C is not convex.

$\boldsymbol{d}$ is convex. It is also easy to verify that $\mathcal{E}(\boldsymbol{f}) = \sum_e \boldsymbol{r}(e)\boldsymbol{f}(e)^2$ is a convex function, and hence finding an electrical flow is an instance of convex minmization:

1.4 More Graph Optimization Problems

Maximum flow. Again, let $G = (V, E)$ be an undirected, connected graph with n vertices and m edges. Suppose we want to find a flow $\boldsymbol{f} \in \mathbb{R}^E$ that routes $\boldsymbol{d}$, but instead of trying to minimize electrical energy, we try to pick an $\boldsymbol{f}$ that minimizes the largest amount of flow on any edge, i.e. $\max_e |\boldsymbol{f}_e|$ – which we also denote by $\|\boldsymbol{f}\|_\infty$. We can write this problem as

$$\begin{aligned} &\min_{\boldsymbol{f} \in \mathbb{R}^E} \|\boldsymbol{f}\|_\infty \\ &\text{s.t. } \boldsymbol{B}\boldsymbol{f} = \boldsymbol{d} \end{aligned}$$

This problem is known as the Minimum Congested Flow Problem[4]. It is equivalent to the more famous Maximum Flow Problem.

The behavior of this kind of flow is very different than electrical flow. Consider the question of whether a certain demand can be routed $\|\boldsymbol{f}\|_\infty \leq 1$. Imagine sending goods from a source s to a destination t using a network of train lines that all have the same capacity and asking whether the network is able to route the goods at the rate you want: This boils down to whether routing exists with $\|\boldsymbol{f}\|_\infty \leq 1$, if we set it up right.

We have a very fast, convex optimization-based algorithm for Minimum Congested Flow: In $m\epsilon^{-1}\log^{O(1)} n$ time, we can find a flow $\tilde{\boldsymbol{f}}$ s.t. $\boldsymbol{B}\tilde{\boldsymbol{f}} = \boldsymbol{d}$ and $\left\|\tilde{\boldsymbol{f}}\right\|_\infty \leq (1+\epsilon)\|\boldsymbol{f}^*\|_\infty$, where $\boldsymbol{f}^*$

[4]This version is called undirected, because the graph is undirected, and *uncapacitated* because we are aiming for the same bound on the flow on all edges.

is an optimal solution, i.e. an actual minimum congestion flow routing $\boldsymbol{d}$.

But what if we want ϵ to be very small, e.g. $1/m$? Then this running time isn't so good anymore. But, in this case, we can use other algorithms that find flow $\boldsymbol{f}^*$ *exactly*. Unfortunately, these algorithms take time roughly $m^{4/3+o(1)}$.

Just as the electrical flow problem had a dual voltage problem, so maximum flow has a dual voltage problem, which is know as the s-t minimum cut problem.

Maximum flow, with directions and capacities. We can make the maximum flow problem harder by introducing directed edges: To do so, we allow edges to exist in both directions between vertices, and we require that the flow on a directed edge is always non-negative. So now $G = (V, E)$ is a directed graph. We can also make the problem harder by introducing capacities. We define a capacity vector $\boldsymbol{c} \in \mathbb{R}^E \geq \mathbf{0}$ and try to minimize $\left\| \boldsymbol{C}^{-1}\boldsymbol{f} \right\|_\infty$, where $\boldsymbol{C} = \operatorname{diag}_{e \in E} \boldsymbol{c}(e)$. Then our problem becomes

$$\begin{aligned} \min_{\boldsymbol{f} \in \mathbb{R}^E} & \left\| \boldsymbol{C}^{-1}\boldsymbol{f} \right\|_\infty \\ \text{s.t. } & \boldsymbol{B}\boldsymbol{f} = \boldsymbol{d} \\ & \boldsymbol{f} \geq \mathbf{0}. \end{aligned}$$

For this capacitated, directed maximum flow problem, our best algorithms run in about $O(m\sqrt{n})$ time in sparse graphs and $O(m^{1.483})$ in dense graphs[5], even if we are willing to accept fairly low accuracy solution. If the capacities are allowed to be exponentially large, the best running time we can get is $O(mn)$. For this problem, we do not yet know how to improve over classical combinatorial algorithms using convex optimization.

Multi-commodity flow. We can make the problem even harder still, by simultaneously trying to route two types of flow (imagine pipes with Coke and Pepsi). Our problem now looks like

$$\begin{aligned} \min_{\boldsymbol{f}_1, \boldsymbol{f}_2 \in \mathbb{R}^E} & \left\| \boldsymbol{C}^{-1}(\boldsymbol{f}_1 + \boldsymbol{f}_2) \right\|_\infty \\ \text{s.t. } & \boldsymbol{B}\boldsymbol{f}_1 = \boldsymbol{d}_1 \\ & \boldsymbol{B}\boldsymbol{f}_2 = \boldsymbol{d}_2 \\ & \boldsymbol{f}_1, \boldsymbol{f}_2 \geq \mathbf{0}. \end{aligned}$$

Solving this problem to high accuracy is essentially as hard as solving a general linear program! We should see later in the course how to make this statement precise.

If we in the above problem additionally require that our flows must be integer valued, i.e. $\boldsymbol{f}_1, \boldsymbol{f}_2 \in \mathbb{N}_0$, then the problem becomes NP-complete.

[5]Provided the capacities are integers satisfying a condition like $\boldsymbol{c} \leq n^{100}\mathbf{1}$.

Random walks in a graph. Google famously uses[6] the PageRank problem to help decide how to rank their search results. This problem essentially boils down to computing the *stable distribution* of a random walk on a graph. Suppose $G = (V, E)$ is a directed graph where each outgoing edge (v, u), which we will define as going from u to v, has a transition probability $p_{(v,u)} > 0$ s.t. $\sum_{z \leftarrow u} p_{(z,u)} = 1$. We can take a step of a random walk on the vertex set by starting at some vertex $u_0 = u$, and then randomly pick one of the outgoing edges (v, u) with probability $p_{(v,u)}$ and move to the chosen vertex $u_1 = v$. Repeating this procedure, to take a step from the next vertex u_1, gives us a *random walk* in the graph, a sequence of vertices $u_0, u_1, u_2 \ldots, u_k$.

We let $\boldsymbol{P} \in \mathbb{R}^{V \times V}$ be the matrix of transition probabilities given by

$$\boldsymbol{P}_{vu} = \begin{cases} p_{(v,u)} & \text{for } (u, v) \in E \\ 0 & \text{o.w.} \end{cases}$$

Any probability distribution over the vertices can be specified by a vector $\boldsymbol{p} \in \mathbb{R}^V$ where $\boldsymbol{p} \geq \boldsymbol{0}$ and $\sum_v \boldsymbol{p}(v) = 1$. We say that probability distribution $\boldsymbol{\pi}$ on the vertices is a *stable distribution* of the random walk if $\boldsymbol{\pi} = \boldsymbol{P}\boldsymbol{\pi}$. A strongly connected graph always has exactly one stable distribution.

How quickly can we compute the stable distribution of a general random walk? Under some mild conditions on the stable distribution[7], we can find a high accuracy approximation of $\boldsymbol{\pi}$ in time $O(m \log^c n)$ for some constant c.

This problem does not easily fit in a framework of convex optimization, but nonetheless, our fastest algorithms for it use ideas from convex optimization.

Topics in this Course

In this course, we will try to address the following questions.

1. What are the fundamental tools of fast convex optimization?
2. What are some problems we can solve quickly on graphs using optimization?
3. What can graphs teach us about convex optimization?
4. What algorithm design techniques are good for getting algorithms that quickly find a crude approximate solution? And what techniques are best when we need to get a highly accurate answer?
5. What is special about flow problems?

[6] At least they did at some point.

[7] Roughly something like $\max_v 1/\boldsymbol{\pi}(v) \leq n^{100}$.

Part I

Introduction to Convex Optimization

Chapter 2

Some Basic Optimization, Convex Geometry, and Linear Algebra

2.1 Overview

In this chapter, we will

1. Start with an overview (i.e. this list).
2. Learn some basic terminology and facts about optimization.
3. Recall our definition of convex functions and see how convex functions can also be understood in terms of a characterization based on first derivatives.
4. See how the first derivatives of a convex function can certify that we are at a global minimum.

2.2 Optimization Problems

Focusing for now on optimization over $\boldsymbol{x} \in \mathbb{R}^n$, we usually write optimization problems as:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \text{ (or } \max) \; & f(\boldsymbol{x}) \\ s.t. \; & g_1(\boldsymbol{x}) \leq b_1 \\ & . \\ & . \\ & . \\ & g_m(\boldsymbol{x}) \leq b_m \end{aligned}$$

where $\{g_i(\boldsymbol{x})\}_{i=1}^m$ encode the constraints. For example, in the following optimization problem from the previous chapter

$$\min_{\boldsymbol{f} \in \mathbb{R}^E} \sum_e \boldsymbol{r}(e)\boldsymbol{f}(e)^2$$
$$\text{s.t. } \boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}$$

we have the constraint $\boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}$. Notice that we can rewrite this constraint as $\boldsymbol{B}\boldsymbol{f} \leq \boldsymbol{d}$ and $-\boldsymbol{B}\boldsymbol{f} \leq -\boldsymbol{d}$ to match the above setting. The set of points which respect the constraints is called the *feasible set*.

Definition 2.2.1. For a given optimization problem the set $\mathcal{F} = \{\boldsymbol{x} \in \mathbb{R}^n \ : \ g_i(\boldsymbol{x}) \leq b_i, \forall i \in [m]\}$ is called the **feasible set**. A point $\boldsymbol{x} \in \mathcal{F}$ is called a **feasible point**, and a point $\boldsymbol{x}' \notin \mathcal{F}$ is called an **infeasible point**.

Ideally, we would like to find optimal solutions for the optimization problems we consider. Let's define what we mean exactly.

Definition 2.2.2. For a *maximization* problem $\boldsymbol{x}^\star$ is called an **optimal solution** if $f(\boldsymbol{x}^\star) \geq f(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{F}$. Similarly, for a *minimization* problem $\boldsymbol{x}^\star$ is an optimal solution if $f(\boldsymbol{x}^\star) \leq f(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{F}$.

What happens if there are *no feasible points*? In this case, an optimal solution cannot exist, and we say the problem is infeasible.

Definition 2.2.3. If $\mathcal{F} = \emptyset$ we say that the optimization problem is **infeasible**. If $\mathcal{F} \neq \emptyset$ we say the optimization problem is **feasible**.

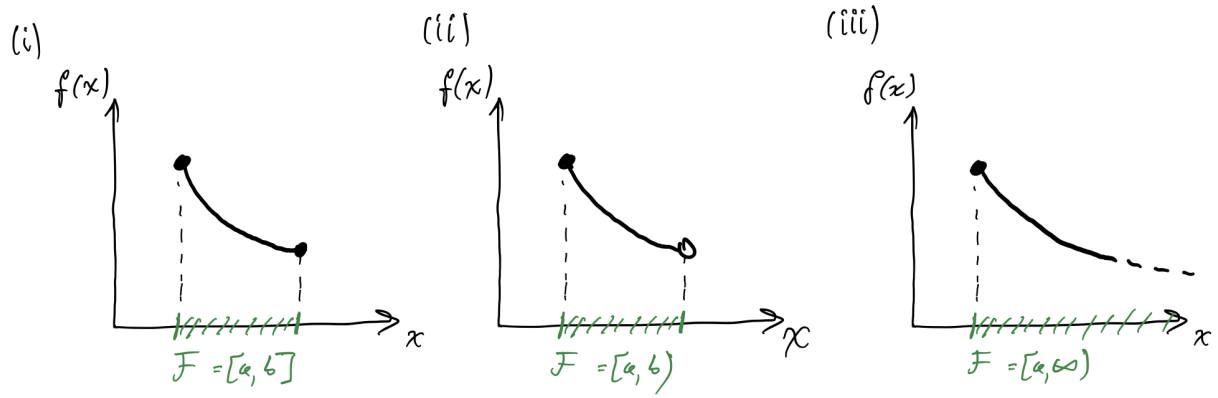

Figure 2.1

Consider three examples depicted in Figure 2.1:

(i) $\mathcal{F} = [a, b]$

(ii) $\mathcal{F} = [a, b)$

(iii) $\mathcal{F} = [a, \infty)$

In the first example, the minimum of the function is attained at b. In the second case the region is open and therefore there is no minimum function value, since for every point we will choose, there will always be another point with a smaller function value. Lastly, in the third example, the region is unbounded and the function decreasing, thus again there will always be another point with a smaller function value.

Sufficient Condition for Optimality. The following theorem, which is a fundamental theorem in real analysis, gives us a sufficient (though not necessary) condition for optimality.

Theorem (Extreme Value Theorem)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function and $\mathcal{F} \subseteq \mathbb{R}^n$ be nonempty, bounded, and closed. Then, the optimization problem $\min f(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{F}$ has an optimal solution.

2.3 A Characterization of Convex Functions

Recall the definitions of convex sets and convex functions that we introduced in Chapter 1:

Definition 2.3.1. A set $S \subseteq \mathbb{R}^n$ is called a **convex set** if any two points in S contain their line, i.e. for any $\boldsymbol{x}, \boldsymbol{y} \in S$ we have that $\theta\boldsymbol{x} + (1-\theta)\boldsymbol{y} \in S$ for any $\theta \in [0, 1]$.

Definition 2.3.2. For a convex set $S \subseteq \mathbb{R}^n$, we say that a function $f : S \to \mathbb{R}$ is **convex on** S if for any two points $\boldsymbol{x}, \boldsymbol{y} \in S$ and any $\theta \in [0, 1]$ we have that:

$$f\left(\theta\boldsymbol{x} + (1-\theta)\boldsymbol{y}\right) \leq \theta f(\boldsymbol{x}) + \Big(1-\theta\Big) f(\boldsymbol{y}).$$

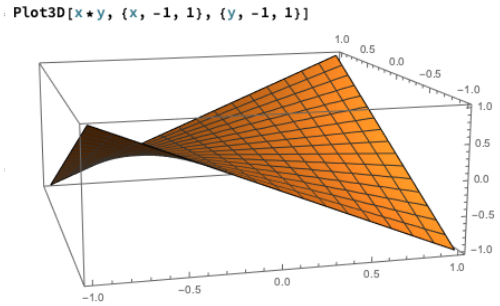

Figure 2.2: This plot shows the function $f(x, y) = xy$. For any fixed y_0, the function $h(x) = f(x, y_0) = xy_0$ is linear in x, and so is a convex function in x. But is f convex?

We will first give an important characterization of convex function. To do so, we need to characterize multivariate functions via their Taylor expansion.

Notation for this section. In the rest of this section, we frequently consider a multivariate function f whose domain is a set $S \subseteq \mathbb{R}^n$, which we will require to be open. When we additionally require that S is convex, we will specify this. Note that $S = \mathbb{R}^n$ is both open and convex and it suffices to keep this case in mind. Things sometimes get more complicated if S is not open, e.g. when the domain of f has a boundary. We will leave those complications for another time.

2.3.1 First-order Taylor Approximation

Definition 2.3.3. The **gradient** of a function $f : S \to \mathbb{R}$ at point $\boldsymbol{x} \in S$ is denoted $\boldsymbol{\nabla} f(\boldsymbol{x})$ and is defined as

$$\boldsymbol{\nabla} f(\boldsymbol{x}) = \left[\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}(1)}, \ldots, \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}(n)}\right]^\top$$

First-order Taylor expansion. For a function $f : \mathbb{R} \to \mathbb{R}$ of a single variable, differentiable at $x \in \mathbb{R}$

$$f(x + \delta) = f(x) + f'(x)\delta + o(|\delta|)$$

where by definition:

$$\lim_{\delta \to 0} \frac{o(|\delta|)}{|\delta|} = 0.$$

Similarly, a multivariate function $f : S \to \mathbb{R}$ is said to be *(Fréchet) differentiable* at $\boldsymbol{x} \in S$ when there exists $\boldsymbol{\nabla} f(\boldsymbol{x}) \in \mathbb{R}^n$ s.t.

$$\lim_{\boldsymbol{\delta} \to \mathbf{0}} \frac{\left\| f(\boldsymbol{x} + \boldsymbol{\delta}) - f(\boldsymbol{x}) - \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{\delta} \right\|_2}{\|\boldsymbol{\delta}\|_2} = 0.$$

Note that this is equivalent to saying that $f(\boldsymbol{x} + \boldsymbol{\delta}) = f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{\delta} + o(\|\boldsymbol{\delta}\|_2)$.

We say that f is *continuously differentiable* on a set $S \subseteq \mathbb{R}^n$ if it is differentiable and in addition the gradient is continuous on S. A differentiable convex function whose domain is an open convex set $S \subseteq \mathbb{R}^n$ is always continuously differentiable[1].

Remark. In this course, we will generally err on the side of being informal about functional analysis when we can afford to, and we will not worry too much about the details of different notions of differentiability (e.g. Fréchet and Gateaux differentiability), except when it turns out to be important.

Theorem 2.3.4 (Taylor's Theorem, multivariate first-order remainder form)**.** *If $f : S \to \mathbb{R}$ is continuously differentiable over $[\boldsymbol{x}, \boldsymbol{y}]$, then for some $\boldsymbol{z} \in [\boldsymbol{x}, \boldsymbol{y}]$,*

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\boldsymbol{y} - \boldsymbol{x}).$$

[1] See p. 248, Corollary 25.5.1 in *Convex Analysis* by Rockafellar (my version is the Second print, 1972). Rockefellar's corollary concerns finite convex functions, because he otherwise allows convex functions that may take on the values $\pm\infty$.

This theorem is useful for showing that the function f can be approximated by the affine function $\boldsymbol{y} \to f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$ when $\boldsymbol{y}$ is "close to" $\boldsymbol{x}$ in some sense.

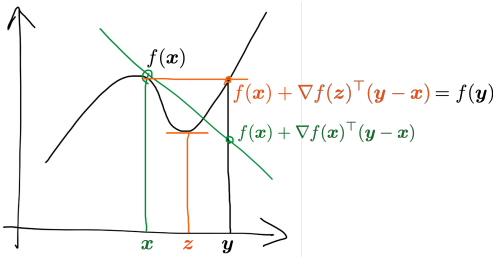

Figure 2.3: The function $f(\boldsymbol{y})$ equals its linear approximation based at x, with gradient coming from an intermediate point z, i.e. $f(\boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\boldsymbol{y} - \boldsymbol{x})$.

2.3.2 Directional Derivatives

Definition 2.3.5. Let $f : S \to \mathbb{R}$ be a function differentiable at $\boldsymbol{x} \in S$ and let us consider $\boldsymbol{d} \in \mathbb{R}^n$. We define the **derivative of f at $\boldsymbol{x}$ in direction $\boldsymbol{d}$** as:

$$Df(\boldsymbol{x})[\boldsymbol{d}] = \lim_{\lambda \to \boldsymbol{0}} \frac{f(\boldsymbol{x} + \lambda \boldsymbol{d}) - f(\boldsymbol{x})}{\lambda}$$

Proposition 2.3.6. $Df(\boldsymbol{x})[\boldsymbol{d}] = \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{d}$.

Proof. Using the first order expansion of f at $\boldsymbol{x}$:

$$f(\boldsymbol{x} + \lambda \boldsymbol{d}) = f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\lambda \boldsymbol{d}) + o(\|\lambda \boldsymbol{d}\|_2)$$

hence, dividing by λ (and noticing that $\|\lambda \boldsymbol{d}\|_2 = \lambda \|\boldsymbol{d}\|_2$):

$$\frac{f(\boldsymbol{x} + \lambda \boldsymbol{d}) - f(\boldsymbol{x})}{\lambda} = \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{d} + \frac{o(\lambda \|\boldsymbol{d}\|_2)}{\lambda}$$

letting λ go to 0 concludes the proof. □

2.3.3 Lower Bounding Convex Functions with Affine Functions

In order to prove the characterization of convex functions in the next section we will need the following lemma. This lemma says that any differentiable convex function can be lower bounded by an affine function.

Theorem 2.3.7. *Let S be an open convex subset of $\mathbb{R}^n$, and let $f : S \to \mathbb{R}$ be a differentiable function. Then, f is convex if and only if for any $\boldsymbol{x}, \boldsymbol{y} \in S$ we have that $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$.*

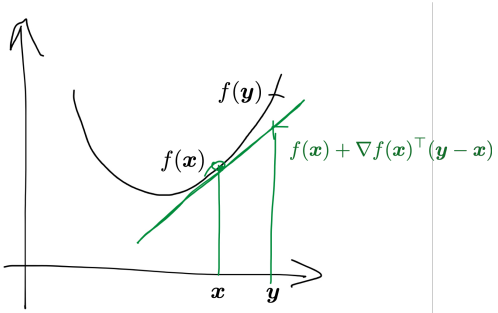

Figure 2.4: The convex function $f(\boldsymbol{y})$ sits above the linear function in $\boldsymbol{y}$ given by $f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top(\boldsymbol{y} - \boldsymbol{x})$.

Proof. [$\implies$] Assume f is convex, then for all $\mathbf{x}, \mathbf{y} \in S$ and $\theta \in [0, 1]$, if we let $\mathbf{z} = \theta\boldsymbol{y} + (1 - \theta)\boldsymbol{x}$, we have that

$$f(\mathbf{z}) = f((1 - \theta)\boldsymbol{x} + \theta\boldsymbol{y}) \leq (1 - \theta)f(\boldsymbol{x}) + \theta f(\boldsymbol{y})$$

and therefore by subtracting $f(\boldsymbol{x})$ from both sides we get:

$$\begin{aligned} f\left(\boldsymbol{x} + \theta(\boldsymbol{y} - \boldsymbol{x})\right) - f(\boldsymbol{x}) &\leq \theta f(\boldsymbol{y}) + (1 - \theta)f(\boldsymbol{x}) - f(\boldsymbol{x}) \\ &= \theta f(\boldsymbol{y}) - \theta f(\boldsymbol{x}). \end{aligned}$$

Thus we get that (for $\theta > 0$):

$$\frac{f\left(\boldsymbol{x} + \theta(\boldsymbol{y} - \boldsymbol{x})\right) - f(\boldsymbol{x})}{\theta} \leq f(\boldsymbol{y}) - f(\boldsymbol{x})$$

Applying Proposition 2.3.6 with $\boldsymbol{d} = \boldsymbol{y} - \boldsymbol{x}$ we have that:

$$\boldsymbol{\nabla} f(\boldsymbol{x})^\top(\boldsymbol{y} - \boldsymbol{x}) = \lim_{\theta \to 0^+} \frac{f(\boldsymbol{x} + \theta(\boldsymbol{y} - \boldsymbol{x})) - f(\boldsymbol{x})}{\theta} \leq f(\boldsymbol{y}) - f(\boldsymbol{x}).$$

[$\impliedby$] Assume that $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top(\boldsymbol{y} - \boldsymbol{x})$ for all $\mathbf{x}, \mathbf{y} \in S$ and show that f is convex. Let $\mathbf{x}, \mathbf{y} \in S$ and $\mathbf{z} = \theta\boldsymbol{y} + (1 - \theta)\boldsymbol{x}$. By our assumption we have that:

$$f(\boldsymbol{y}) \geq f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top(\boldsymbol{y} - \mathbf{z}) \tag{2.1}$$

$$f(\boldsymbol{x}) \geq f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top(\boldsymbol{x} - \mathbf{z}) \tag{2.2}$$

Observe that $\boldsymbol{y} - \boldsymbol{z} = (1 - \theta)(\boldsymbol{y} - \boldsymbol{x})$ and $\boldsymbol{x} - \boldsymbol{z} = \theta(\boldsymbol{x} - \boldsymbol{y})$. Thus adding θ times (2.1) to $(1 - \theta)$ times (2.2) gives cancellation of the vectors multiplying the gradient, yielding

$$\begin{aligned} \theta f(\boldsymbol{y}) + (1 - \theta)f(\boldsymbol{x}) &\geq f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top \mathbf{0} \\ &= f(\theta\boldsymbol{y} + (1 - \theta)\boldsymbol{x}) \end{aligned}$$

This is exactly the definition of convexity. □

2.4 Conditions for Optimality

We now want to find necessary and sufficient conditions for local optimality.

Definition 2.4.1. Consider a differentiable function $f : S \to \mathbb{R}$. A point $\boldsymbol{x} \in S$ at which $\boldsymbol{\nabla} f(\boldsymbol{x}) = \mathbf{0}$ is called a **stationary point**.

Proposition 2.4.2. *If $\boldsymbol{x}$ is a local extremum of a differentiable function $f : S \to \mathbb{R}$ then $\boldsymbol{\nabla} f(\boldsymbol{x}) = \mathbf{0}$.*

Proof. Let us assume that $\boldsymbol{x}$ is a local minimum for f. Then for all $\boldsymbol{d} \in \mathbb{R}^n$, $f(\boldsymbol{x}) \leq f(\boldsymbol{x}+\lambda \boldsymbol{d})$ for λ small enough. Hence:

$$0 \leq f(\boldsymbol{x} + \lambda \boldsymbol{d}) - f(\boldsymbol{x}) = \lambda \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{d} + o(\|\lambda \boldsymbol{d}\|)$$

dividing by $\lambda > 0$ and letting $\lambda \to 0^+$, we obtain $0 \leq \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{d}$. But, taking $\boldsymbol{d} = -\boldsymbol{\nabla} f(\boldsymbol{x})$, we get $0 \leq -\|\boldsymbol{\nabla} f(\boldsymbol{x})\|_2^2$. This implies that $\boldsymbol{\nabla} f(\boldsymbol{x}) = \mathbf{0}$.

The case where $\boldsymbol{x}$ is a local maximum can be dealt with similarly. □

Remark 2.4.3. For this proposition to hold, it is important that S is open.

For convex functions however it turns out that a stationary point necessarily implies that the function is at its minimum. Together with the proposition above, this says that for a convex function on $\mathbb{R}^n$ a point is optimal if and only if it is stationary.

Proposition 2.4.4. *Let $S \subseteq \mathbb{R}^n$ be an open convex set and let $f : S \to \mathbb{R}$ be a differentiable and convex function. If $\boldsymbol{x}$ is a stationary point then $\boldsymbol{x}$ is a global minimum.*

Proof. From Theorem 2.3.7 we know that for all $\boldsymbol{x}, \boldsymbol{y} \in S$: $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x})$. Since $\boldsymbol{\nabla} f(\boldsymbol{x}) = \mathbf{0}$ this implies that $f(\boldsymbol{y}) \geq f(\boldsymbol{x})$. As this holds for any $\boldsymbol{y} \in S$, $\boldsymbol{x}$ is a global minimum.

□

Chapter 3

Convexity, Second Derivatives and Gradient Descent

Notation for this chapter. In this chapter, we sometimes consider a multivariate function f whose domain is a set $S \subseteq \mathbb{R}^n$, which we will require to be open. When we additionally require that S is convex, we will specify this. Note that $S = \mathbb{R}^n$ is both open and convex and it suffices to keep this case in mind. Things sometimes get more complicated if S is not open, e.g. when the domain of f has a boundary. We will leave those complications for another time.

3.1 A Review of Linear Algebra

Semi-definiteness of a matrix. The following classification of symmetric matrices will be useful.

Definition 3.1.1. Let $\boldsymbol{A}$ by a symmetric matrix in $\mathbb{R}^{n\times n}$. We say that $\boldsymbol{A}$ is:

1. *positive definite* iff $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} > 0$ for all $x \in \mathbb{R}^n \setminus \{0\}$;
2. *positive semidefinite* iff $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} \geq 0$ for all $x \in \mathbb{R}^n$;
3. If neither $\boldsymbol{A}$ nor $-\boldsymbol{A}$ is positive semi-definite, we say that $\boldsymbol{A}$ is *indefinite*.

Example: indefinite matrix. Consider the following matrix $\boldsymbol{A}$:

$$\boldsymbol{A} := \begin{bmatrix} +4 & -1 \\ -1 & -2 \end{bmatrix}$$

For $\boldsymbol{x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, we have $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} = 4 > 0$. For $\boldsymbol{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ we have $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} = -2 < 0$. $\boldsymbol{A}$ is therefore indefinite.

The following theorem gives a useful characterization of (semi)definite matrices.

Theorem 3.1.2. *Let $\boldsymbol{A}$ be a symmetric matrix in $\mathbb{R}^{n\times n}$.*

1. *$\boldsymbol{A}$ is positive definite iff all its eigenvalues are positive;*
2. *$\boldsymbol{A}$ is positive semidefinite iff all its eigenvalues are non-negative;*

In order to prove this theorem, let us first recall the Spectral Theorem for symmetric matrices.

Theorem 3.1.3 (The Spectral Theorem for Symmetric Matrices)**.** *For all symmetric $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ there exist $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ and a diagonal matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$ s.t.*

1. *$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$.*
2. *$\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}$ (the $n \times n$ identity matrix). I.e. the columns of $\boldsymbol{V}$ form an orthonormal basis. Furthermore, $\boldsymbol{v}_i$ is an eigenvector of $\lambda_i(\boldsymbol{A})$, the ith eigenvalue of $\boldsymbol{A}$.*
3. *$\boldsymbol{\Lambda}_{ii} = \lambda_i(\boldsymbol{A})$.*

Using the Spectral Theorem, we can show the following result:

Theorem 3.1.4 (The Courant-Fischer Theorem)**.** *Let $\boldsymbol{A}$ be a symmetric matrix in $\mathbb{R}^{n\times n}$, with eigenvalues $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$. Then*

1. $$\lambda_i = \min_{\substack{\text{subspace } W\subseteq\mathbb{R}^n \\ \dim(W)=i}} \max_{\boldsymbol{x}\in W, \boldsymbol{x}\neq\boldsymbol{0}} \frac{\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}$$
2. $$\lambda_i = \max_{\substack{\text{subspace } W\subseteq\mathbb{R}^n \\ \dim(W)=n+1-i}} \min_{\boldsymbol{x}\in W, \boldsymbol{x}\neq\boldsymbol{0}} \frac{\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}$$

Theorem 3.1.2 is an immediate corollary of Theorem 3.1.4, since we can see that the minimum value of the quadratic form $\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}$ over $\boldsymbol{x} \in W = \mathbb{R}^n$ is $\lambda_1(\boldsymbol{A})\|\boldsymbol{x}\|_2^2$.

Proof of Theorem 3.1.4. We start by showing Part 1.

Consider letting $W = \operatorname{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_i\}$, and normalize $\boldsymbol{x} \in W$ so that $\|\boldsymbol{x}\|_2 = 1$. Then $\boldsymbol{x} = \sum_{j=1}^{i} \boldsymbol{c}(j)\boldsymbol{v}_j$ for some vector $\boldsymbol{c} \in \mathbb{R}^i$ with $\|\boldsymbol{c}\|_2 = 1$.

Using the decomposition from Theorem 3.1.3 $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues of $\boldsymbol{A}$, which we take to be sorted in increasing order. Then $\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^\top\boldsymbol{V}^\top\boldsymbol{\Lambda}\boldsymbol{V}\boldsymbol{x} = (\boldsymbol{V}\boldsymbol{x})^\top\boldsymbol{\Lambda}(\boldsymbol{V}\boldsymbol{x}) = \sum_{j=1}^{i}\lambda_j\boldsymbol{c}(j)^2 \le \lambda_i\|\boldsymbol{c}\|_2^2 = \lambda_i$. So this choice of W ensures the maximizer cannot achieve a value above λ_i.

But is it possible that the "minimizer" can do better by choosing a different W? Let $T = \text{span}\{\boldsymbol{v}_i, \ldots, \boldsymbol{v}_n\}$. As $\dim(T) = n + 1 - i$ and $\dim(W) = i$, we must have $\dim(W \cap T) \geq 1$, by a standard property of subspaces. Hence for any W of this dimension,

$$\begin{aligned}\max_{\boldsymbol{x} \in W, \boldsymbol{x} \neq \boldsymbol{0}} \frac{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} &\geq \max_{\boldsymbol{x} \in W \cap T, \boldsymbol{x} \neq \boldsymbol{0}} \frac{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \\ &\geq \min_{\substack{\text{subspace } V \subseteq T \\ \dim(V) = 1}} \max_{\boldsymbol{x} \in V, \boldsymbol{x} \neq \boldsymbol{0}} \frac{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \lambda_i,\end{aligned}$$

where the last equality follows from a similar calculation to our first one. Thus, λ_i can always be achieved by the "maximizer" for all W of this dimension.

Part 2 can be dealt with similarly.

□

Example: a positive semidefinite matrix. Consider the following matrix $\boldsymbol{A}$:

$$\boldsymbol{A} := \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

For $\boldsymbol{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, we have $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$, so $\lambda = 0$ is an eigenvalue of $\boldsymbol{A}$. For $\boldsymbol{x} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, we have $\boldsymbol{A}\boldsymbol{x} = \begin{pmatrix} 2 \\ -2 \end{pmatrix} = 2\boldsymbol{x}$, so $\lambda = 2$ is the other eigenvalue of $\boldsymbol{A}$. As both are non-negative, by the theorem above, $\boldsymbol{A}$ is positive semidefinite.

Since we are learning about symmetric matrices, there is one more fact that everyone should know about them. We'll use $\lambda_{\max}(\boldsymbol{A})$ denote maximum eigenvalue of a matrix $\boldsymbol{A}$, and $\lambda_{\min}(\boldsymbol{A})$ the minimum.

Claim 3.1.5. *For a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, $\|\boldsymbol{A}\| = \max(|\lambda_{\max}(\boldsymbol{A})|, |\lambda_{\min}(\boldsymbol{A})|)$.*

3.2 Characterizations of Convexity and Optimality via Second Derivatives

We will now use the second derivatives of a function to obtain characterizations of convexity and optimality. We will begin by introducing the *Hessian*, the matrix of pairwise second derivatives of a function. We will see that it plays a role in approximating a function via a second-order Taylor expansion. We will then use *semi-definiteness* of the Hessian matrix to characterize both conditions of optimality as well as the convexity of a function.

Definition 3.2.1. Given a function $f : S \to \mathbb{R}$ its **Hessian** matrix at point $\boldsymbol{x} \in S$ denoted $\boldsymbol{H}_f(\boldsymbol{x})$ (also sometimes denoted $\boldsymbol{\nabla}^2 f(\boldsymbol{x})$) is:

$$\boldsymbol{H}_f(\boldsymbol{x}) := \begin{bmatrix} \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(1)^2} & \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(1)\partial \boldsymbol{x}(2)} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(1)\partial \boldsymbol{x}(n)} \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(2)\partial \boldsymbol{x}(1)} & \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(2)^2} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(2)\partial \boldsymbol{x}(n)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(n)\partial \boldsymbol{x}(1)} & \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(n)\partial \boldsymbol{x}(2)} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}(n)^2} \end{bmatrix}$$

Second-order Taylor expansion. When f is twice differentiable it is possible to obtain an approximation of f by quadratic functions. Our definition of $f : S \to \mathbb{R}$ being twice (Fréchet) differentiable at $\boldsymbol{x} \in S$ is that there exists $\boldsymbol{\nabla} f(\boldsymbol{x}) \in \mathbb{R}^n$ and $\boldsymbol{H}_f(\boldsymbol{x}) \in R^{n \times n}$ s.t.

$$\lim_{\boldsymbol{\delta} \to \boldsymbol{0}} \frac{\left\| f(\boldsymbol{x} + \boldsymbol{\delta}) - f(\boldsymbol{x}) - \left(\boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^\top \boldsymbol{H}_f(\boldsymbol{x}) \boldsymbol{\delta} \right) \right\|_2}{\|\boldsymbol{\delta}\|_2^2} = 0.$$

This is equivalent to saying that for all $\boldsymbol{\delta}$

$$f(\boldsymbol{x} + \boldsymbol{\delta}) = f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^\top \boldsymbol{H}_f(\boldsymbol{x}) \boldsymbol{\delta} + o(\|\boldsymbol{\delta}\|_2^2).$$

where by definition:

$$\lim_{\boldsymbol{\delta} \to \boldsymbol{0}} \frac{o(\|\boldsymbol{\delta}\|^2)}{\|\boldsymbol{\delta}\|_2^2} = 0$$

We say that f is *twice continuously differentiable* on a set $S \subseteq \mathbb{R}^n$ if it is twice differentiable and in addition the gradient and Hessian are continuous on S.

As for first order expansions, we have a Taylor's Theorem, which we state in the so-called remainder form.

Theorem 3.2.2 (Taylor's Theorem, multivariate second-order remainder form)**.** *If $f : S \to \mathbb{R}$ is twice continuously differentiable over $[\boldsymbol{x}, \boldsymbol{y}]$, then for some $\boldsymbol{z} \in [\boldsymbol{x}, \boldsymbol{y}]$,*

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{1}{2} (\boldsymbol{y} - \boldsymbol{x})^\top \boldsymbol{H}_f(\boldsymbol{z}) (\boldsymbol{y} - \boldsymbol{x})$$

3.2.1 A Necessary Condition for Local Extrema

Recall that in the previous chapter, we show the following proposition.

Proposition 3.2.3. *If $\boldsymbol{x}$ is a local extremum of a differentiable function $f : S \to \mathbb{R}$ then $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$.*

We can now give the second-order necessary conditions for local extrema via the Hessian.

Theorem 3.2.4. *Let $f : S \to \mathbb{R}$ be a function twice differentiable at $\boldsymbol{x} \in S$. If $\boldsymbol{x}$ is a local minimum, then $\boldsymbol{H}_f(\boldsymbol{x})$ is positive semidefinite.*

Proof. Let us assume that $\boldsymbol{x}$ is a local minimum. We know from Proposition 3.2.3 that $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$, hence the second-order expansion at $\boldsymbol{x}$ takes the form:

$$f(\boldsymbol{x} + \lambda \boldsymbol{d}) = f(\boldsymbol{x}) + \lambda^2 \frac{1}{2} \boldsymbol{d}^\top \boldsymbol{H}_f(\boldsymbol{x}) \boldsymbol{d} + o(\lambda^2 \left\| \boldsymbol{d} \right\|_2^2)$$

Because $\boldsymbol{x}$ is a local minimum, we can then derive

$$0 \leq \lim_{\lambda \to 0^+} \frac{f(\boldsymbol{x} + \lambda \boldsymbol{d}) - f(\boldsymbol{x})}{\lambda^2} = \frac{1}{2} \boldsymbol{d}^\top \boldsymbol{H}_f(\boldsymbol{x}) \boldsymbol{d}$$

This is true for any $\boldsymbol{d}$, hence $\boldsymbol{H}_f(\boldsymbol{x})$ is positive semidefinite. □

Remark 3.2.5. Again, for this proposition to hold, it is important that S is open.

3.2.2 A sufficient condition for local extrema

A local minimum thus is a stationary point and has a positive semi-definite Hessian. The converse is almost true, but we need to strengthen the Hessian condition slightly.

Theorem 3.2.6. *Let $f : S \to \mathbb{R}$ be a function twice differentiable at a stationary point $\boldsymbol{x} \in S$. If $\boldsymbol{H}_f(\boldsymbol{x})$ is positive definite then $\boldsymbol{x}$ is a local minimum.*

Proof. Let us assume that $\boldsymbol{H}_f(\boldsymbol{x})$ is positive definite. We know that $\boldsymbol{x}$ is a stationary point. We can write the second-order expansion at $\boldsymbol{x}$:

$$f(\boldsymbol{x} + \boldsymbol{\delta}) = f(\boldsymbol{x}) + \frac{1}{2} \boldsymbol{\delta}^\top \boldsymbol{H}_f(\boldsymbol{x}) \boldsymbol{\delta} + o(\left\| \boldsymbol{\delta} \right\|_2^2)$$

Because the Hessian is positive definite, it has a strictly positive minimum eigenvalue $\lambda_{\min}$, we can conclude that $\boldsymbol{\delta}^\top \boldsymbol{H}_f(\boldsymbol{x}) \boldsymbol{\delta} \geq \lambda_{\min} \left\| \boldsymbol{\delta} \right\|_2^2$. From this, we conclude that when $\left\| \boldsymbol{\delta} \right\|_2^2$ is small enough, $f(\boldsymbol{x} + \boldsymbol{\delta}) - f(\boldsymbol{x}) \geq \frac{1}{4} \lambda_{\min} \left\| \boldsymbol{\delta} \right\|_2^2 > 0$. This proves that $\boldsymbol{x}$ is a local minimum. □

Remark 3.2.7. When $\boldsymbol{H}_f(\boldsymbol{x})$ is indefinite at a stationary point $\boldsymbol{x}$, we have what is known as a *saddle point*: $\boldsymbol{x}$ will be a minimum along the eigenvectors of $\boldsymbol{H}_f(\boldsymbol{x})$ for which the eigenvalues are positive and a maximum along the eigenvectors of $\boldsymbol{H}_f(\boldsymbol{x})$ for which the eigenvalues are negative.

3.2.3 Characterization of convexity

Definition 3.2.8. For a convex set $S \subseteq R^n$, we say that a function $f : S \to \mathbb{R}$ is **strictly convex on** S if for any two points $\boldsymbol{x}_1, \boldsymbol{x}_2 \in S$ and any $\theta \in (0, 1)$ we have that:

$$f\left(\theta \boldsymbol{x}_1 + (1 - \theta) \boldsymbol{x}_2\right) < \theta f(\boldsymbol{x}_1) + \Big(1 - \theta\Big) f(\boldsymbol{x}_2).$$

Theorem 3.2.9. *Let $S \subseteq \mathbb{R}^n$ be open and convex, and let $f : S \to \mathbb{R}$ be twice continuously differentiable.*

1. *If $H_f(\boldsymbol{x})$ is positive semi-definite for any $\boldsymbol{x} \in S$ then f is convex on S.*
2. *If $H_f(\boldsymbol{x})$ is positive definite for any $\boldsymbol{x} \in S$ then f is **strictly** convex on S.*
3. *If f is convex, then $H_f(\boldsymbol{x})$ is positive semi-definite $\forall \boldsymbol{x} \in S$.*

Proof.

1. By applying Theorem 3.2.2, we find that for some $\boldsymbol{z} \in [\boldsymbol{x}, \boldsymbol{y}]$:

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{1}{2}\Big((\boldsymbol{y} - \boldsymbol{x})^\top H_f(\boldsymbol{z})(\boldsymbol{y} - \boldsymbol{x})\Big)$$

   If $H_f(\boldsymbol{z})$ is positive semi-definite, this necessarily implies that:

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$$

   and from Theorem 2.3.7 we get that f is convex.

2. if $H_f(\boldsymbol{x})$ is positive definite, we have that:

$$f(\boldsymbol{y}) > f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}).$$

   Applying the same idea as in Theorem 2.3.7 we can show that in this case f is **strictly** convex.

3. Let f be a convex function. For $\boldsymbol{x} \in S$, and some small $\lambda > 0$, for any $\mathbf{d} \in \mathbb{R}^n$ we have that $\boldsymbol{x} + \lambda \mathbf{d} \in S$. From the Taylor expansion of f we get:

$$f(\boldsymbol{x} + \lambda \mathbf{d}) = f(\boldsymbol{x}) + \lambda \boldsymbol{\nabla} f(\boldsymbol{x})^\top \mathbf{d} + \frac{\lambda^2}{2} \mathbf{d}^\top H_f(\boldsymbol{x}) \mathbf{d} + o(\lambda^2 \left\| \boldsymbol{d} \right\|_2^2).$$

   From Lemma 2.3.7 we get that if f is convex then:

$$f(\boldsymbol{x} + \lambda \mathbf{d}) \geq f(\boldsymbol{x}) + \lambda \boldsymbol{\nabla} f(\boldsymbol{x})^\top \mathbf{d}.$$

   Therefore, we have that for any $\mathbf{d} \in \mathbb{R}^n$:

$$\frac{\lambda^2}{2} \mathbf{d}^\top H_f(\boldsymbol{x}) \mathbf{d} + o(||\lambda \mathbf{d}||^2) \geq 0$$

   Dividing by λ^2 and taking $\lambda \to 0^+$ gives us that for any $\mathbf{d} \in \mathbb{R}^n$: $\mathbf{d}^\top H_f(\boldsymbol{x}) \mathbf{d} \geq 0$. □

Remark 3.2.10. It is important to note that if S is open and f is strictly convex, then $\boldsymbol{H}_f(\boldsymbol{x})$ may still (only) be positive semi-definite $\forall \boldsymbol{x} \in S$. Consider $f(x) = x^4$ which is strictly convex, then the Hessian is $\boldsymbol{H}_f(x) = 12x^2$ which equals 0 at $x = 0$.

3.3 Gradient Descent - An Approach to Optimization?

We have begun to develop an understanding of convex functions, and what we have learned already suggests a way for us to try to find an approximate minimizer of a given convex function.

Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, and we want to solve

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$$

We would like to find $\boldsymbol{x}^*$, a global minimizer of f. Suppose we start with some initial guess $\boldsymbol{x}_0$, and we want to update it to $\boldsymbol{x}_1$ with $f(\boldsymbol{x}_1) < f(\boldsymbol{x}_0)$. If we can repeatedly make updates like this, maybe we eventually find a point with nearly minimum function value, i.e. some $\tilde{\boldsymbol{x}}$ with $f(\tilde{\boldsymbol{x}}) \approx f(\boldsymbol{x}^*)$?

Recall that $f(\boldsymbol{x}_0 + \boldsymbol{\delta}) = f(\boldsymbol{x}_0) + \boldsymbol{\nabla} f(\boldsymbol{x}_0)^\top \boldsymbol{\delta} + o(\|\boldsymbol{\delta}\|_2)$. This means that if we choose $\boldsymbol{x}_1 = \boldsymbol{x}_0 - \lambda \boldsymbol{\nabla} f(\boldsymbol{x}_0)$, we get

$$f(\boldsymbol{x}_0 - \lambda \boldsymbol{\nabla} f(\boldsymbol{x}_0)) = f(\boldsymbol{x}_0) - \lambda \|\boldsymbol{\nabla} f(\boldsymbol{x}_0)\|_2^2 + o(\lambda \|\boldsymbol{\nabla} f(\boldsymbol{x}_0)\|_2)$$

And because f is convex, we know that $\boldsymbol{\nabla} f(\boldsymbol{x}_0) \neq \mathbf{0}$ unless we are already at a global minimum. So, for some small enough $\lambda > 0$, we should get $f(\boldsymbol{x}_1) < f(\boldsymbol{x}_0)$ unless we're already at a global minimizer. This idea of taking a step in the direction of $-\boldsymbol{\nabla} f(\boldsymbol{x}_0)$ is what is called *Gradient Descent*. But how do we choose λ each time? And does this lead to an algorithm that quickly reaches a point with close to minimal function value? To get good answers to these questions, we need to assume more about the function f that we are trying to minimize.

In the following subsection, we will see some conditions that suffice. But there are also many other settings where one can show that some form of gradient descent converges.

3.3.1 A Quantitative Bound on Changes in the Gradient

Definition 3.3.1. Let $f : S \to \mathbb{R}$ be a differentiable function, where $S \subseteq \mathbb{R}^n$ is convex and open. We say that f is *β-gradient Lipschitz* iff for all $\boldsymbol{x}, \boldsymbol{y} \in S$

$$\|\boldsymbol{\nabla} f(\boldsymbol{x}) - \boldsymbol{\nabla} f(\boldsymbol{y})\|_2 \leq \beta \|\boldsymbol{x} - \boldsymbol{y}\|_2 .$$

We also refer to this as f being *β-smooth*.

Proposition 3.3.2. *Consider a twice continuously differentiable $f : S \to \mathbb{R}$. Then f is β-gradient Lipschitz if and only if for all $\boldsymbol{x} \in S$, $\|\boldsymbol{H}_f(\boldsymbol{x})\| \leq \beta$.*

You will prove this in Exercise 3 (Week 2) of the second exercise set.

Proposition 3.3.3. *Let $f : S \to \mathbb{R}$ be a β-gradient Lipschitz function. Then for all $\boldsymbol{x}, \boldsymbol{y} \in S$,*

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{\beta}{2} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

To prove this proposition, we need the following result from multi-variate calculus. This is a restricted form of the fundamental theorem of calculus for line integrals.

Proposition 3.3.4. *Let $f : S \to \mathbb{R}$ be a differentiable function, and consider $\boldsymbol{x}, \boldsymbol{y}$ such that $[\boldsymbol{x}, \boldsymbol{y}] \in S$. Let $\boldsymbol{x}_\theta = \boldsymbol{x} + \theta(\boldsymbol{y} - \boldsymbol{x})$. Then*

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \int_{\theta=0}^{1} \boldsymbol{\nabla} f(\boldsymbol{x}_\theta)^\top (\boldsymbol{y} - \boldsymbol{x}) d\theta$$

Now, we're in a position to show Proposition 3.3.3

Proof of Proposition 3.3.3. Let $f : S \to \mathbb{R}$ be a β-gradient Lipschitz function. Consider arbitrary $\boldsymbol{x}, \boldsymbol{y} \in S$ such that $[\boldsymbol{x}, \boldsymbol{y}] \in S$

$$\begin{aligned}
f(\boldsymbol{y}) &= f(\boldsymbol{x}) + \int_{\theta=0}^{1} \boldsymbol{\nabla} f(\boldsymbol{x}_\theta)^\top (\boldsymbol{y} - \boldsymbol{x}) d\theta \\
&= f(\boldsymbol{x}) + \int_{\theta=0}^{1} \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) d\theta + \int_{\theta=0}^{1} (\boldsymbol{\nabla} f(\boldsymbol{x}_\theta) - \boldsymbol{\nabla} f(\boldsymbol{x}))^\top (\boldsymbol{y} - \boldsymbol{x}) d\theta \\
&\qquad \text{Next we use Cauchy-Schwarz pointwise.} \\
&\qquad \text{We also evaluate the first integral.} \\
&\leq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \int_{\theta=0}^{1} \|\boldsymbol{\nabla} f(\boldsymbol{x}_\theta) - \boldsymbol{\nabla} f(\boldsymbol{x})\| \, \|\boldsymbol{y} - \boldsymbol{x}\| \, d\theta \\
&\qquad \text{Then we apply } \beta\text{-gradient Lipschitz and note } \boldsymbol{x}_\theta - \boldsymbol{x} = \theta(\boldsymbol{y} - \boldsymbol{x}). \\
&\leq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \int_{\theta=0}^{1} \beta\theta \, \|\boldsymbol{y} - \boldsymbol{x}\|^2 \, d\theta. \\
&= f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{\beta}{2} \|\boldsymbol{y} - \boldsymbol{x}\|^2 .
\end{aligned}$$

□

3.3.2 Analyzing Gradient Descent

It turns out that just knowing a function $f : \mathbb{R}^n \to \mathbb{R}$ is convex and β-gradient Lipschitz is enough to let us figure out a reasonable step size for Gradient Descent and let us analyze its convergence.

We start at a point $\boldsymbol{x}_0 \in \mathbb{R}^n$, and we try to find a point $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{\delta}$ with lower function value. We will let our upper bound from Proposition 3.3.3 guide us, in fact, we could ask, what is the *best* update for minimizing this upper bound, i.e. a $\boldsymbol{\delta}$ solving

$$\min_{\boldsymbol{\delta} \in \mathbb{R}^n} f(\boldsymbol{x}_0) + \boldsymbol{\nabla} f(\boldsymbol{x}_0)^\top \boldsymbol{\delta} + \frac{\beta}{2} \|\boldsymbol{\delta}\|^2$$

We can compute the best according to this upper bound by noting first that function is convex and continuously differentiable, and hence will be minimized at any point where the gradient is zero. Thus we want $\mathbf{0} = \boldsymbol{\nabla}_{\boldsymbol{\delta}} \left(f(\boldsymbol{x}_0) + \boldsymbol{\nabla} f(\boldsymbol{x}_0)^\top \boldsymbol{\delta} + \frac{\beta}{2} \|\boldsymbol{\delta}\|^2 \right) = \boldsymbol{\nabla} f(\boldsymbol{x}_0) + \beta \boldsymbol{\delta}$, which occurs at $\boldsymbol{\delta} = -\frac{1}{\beta} \boldsymbol{\nabla} f(\boldsymbol{x}_0)$.

Plugging in this value into the upper bound, we get that $f(\boldsymbol{x}_1) \leq f(\boldsymbol{x}_0) - {\|\boldsymbol{\nabla} f(\boldsymbol{x}_0)\|_2^2}/{2\beta}$.

Now, as our algorithm, we will start with some guess $\boldsymbol{x}_0$, and then at every step we will update our guess using the best step based on our Proposition 3.3.3 upper bound on f at $\boldsymbol{x}_i$, and so we get

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \frac{1}{\beta} \boldsymbol{\nabla} f(\boldsymbol{x}_i) \text{ and } f(\boldsymbol{x}_{i+1}) \leq f(\boldsymbol{x}_i) - \frac{\|\boldsymbol{\nabla} f(\boldsymbol{x}_i)\|_2^2}{2\beta}. \tag{3.1}$$

Let us try to prove that our algorithm converges toward an $\boldsymbol{x}$ with low function value.

We will measure this by looking at

$$\text{gap}_i = f(\boldsymbol{x}_i) - f(\boldsymbol{x}^*)$$

where $\boldsymbol{x}^*$ is a global minimizer of f (note that there may not be a unique minimizer of f). We will try to show that this function value gap grows small. Using $f(\boldsymbol{x}_{i+1}) - f(\boldsymbol{x}_i) = \text{gap}_{i+1} - \text{gap}_i$, we get

$$\text{gap}_{i+1} - \text{gap}_i \leq -\frac{\|\boldsymbol{\nabla} f(\boldsymbol{x}_i)\|_2^2}{2\beta} \tag{3.2}$$

If the gap_i value is never too much bigger than $\frac{\|\boldsymbol{\nabla} f(\boldsymbol{x}_i)\|_2^2}{2\beta}$, then this should help us show we are making progress. But how much can they differ? We will now try to show a limit on this.

Recall that in the previous chapter we showed the following theorem.

Theorem 3.3.5. *Let S be an open convex subset of $\mathbb{R}^n$, and let $f : S \to \mathbb{R}$ be a differentiable function. Then, f is convex if and only if for any $\boldsymbol{x}, \boldsymbol{y} \in S$ we have that $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$.*

Using the convexity of f and the lower bound on convex functions given by Theorem 3.3.5, we have that

$$f(\boldsymbol{x}^*) \geq f(\boldsymbol{x}_i) + \boldsymbol{\nabla} f(\boldsymbol{x}_i)^\top (\boldsymbol{x}^* - \boldsymbol{x}_i) \tag{3.3}$$

Rearranging gets us

$$\begin{aligned} \text{gap}_i &\leq \nabla f(\boldsymbol{x}_i)^\top (\boldsymbol{x}_i - \boldsymbol{x}^*) && (3.4) \\ &\leq \|\nabla f(\boldsymbol{x}_i)\|_2 \|\boldsymbol{x}_i - \boldsymbol{x}^*\|_2 && \text{by Cauchy-Schwarz.} \end{aligned}$$

At this point, we are essentially ready to connect Equation (3.2) with Equation (3.4) and analyze the convergence rate of our algorithm.

However, at the moment, we see that the change $\text{gap}_{i+1} - \text{gap}_i$ in how close we are to the optimum function value is governed by the norm of the gradient $\|\nabla f(\boldsymbol{x}_i)\|_2$, while the size of the gap is related to *both* this quantity and the distance $\|\boldsymbol{x}_i - \boldsymbol{x}^*\|_2$ between the current solution $\boldsymbol{x}_i$ and an optimum $\boldsymbol{x}^*$. Do we need both or can we get rid of, say, the distance? Unfortunately, with this algorithm and for this class of functions, a dependence on the distance is necessary. However, we can simplify things considerably using the following observation, which you will prove in the exercises (Week 2, Exercise 2):

Claim 3.3.6. *When running Gradient Descent as given by the step in Equation* (3.1)*, for all* i $\|\boldsymbol{x}_i - \boldsymbol{x}^*\|_2 \leq \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2$.

Combining this Claim with Equation (3.2) and Equation (3.4),

$$\text{gap}_{i+1} - \text{gap}_i \leq -\frac{1}{2\beta} \cdot \left(\frac{\text{gap}_i}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2} \right)^2 \tag{3.5}$$

At this point, a simple induction will complete the proof of following result.

Theorem 3.3.7. *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be a* β*-gradient Lipschitz, convex function. Let* $\boldsymbol{x}_0$ *be a given starting point, and let* $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$ *be a minimizer of* f*. The Gradient Descent algorithm given by*

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \frac{1}{\beta} \nabla f(\boldsymbol{x}_i)$$

ensures that the k*th iterate satisfies*

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) \leq \frac{2\beta \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2^2}{k+1}.$$

Carrying out this induction is Exercise 4 (Week 2).

3.4 Accelerated Gradient Descent

It turns out that we can get an algorithm that converges substantially faster than Gradient Descent, using an approach known as *Accelerated Gradient Descent*, which was developed by Nesterov [Nes83]. This algorithm in turn improved on some earlier results by Nemirovski and Yudin [NY83]. The phenomenon of acceleration was perhaps first understood in the

context of quadratic functions, minimizing $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^\top \boldsymbol{b}$ when $\boldsymbol{A}$ is positive definite – for this case, the Conjugate Gradient algorithm was developed independently by Hestenes and Stiefel [HS+52] (here at ETH!), and by Lanczos [Lan52]. In the past few years, providing more intuitive explanations of acceleration has been a popular research topic. This presentation is based on an analysis of Nesterov's algorithm developed by Diakonikolas and Orecchia [DO19].

We will adopt a slightly different approach to analyzing this algorithm than what we used in the previous section for Gradient Descent.

We will use $\boldsymbol{x}_0$ to denote the starting point of our algorithm, and we will produce a sequence of iterates $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k$. At each iterate $\boldsymbol{x}_i$, we will compute the gradient $\boldsymbol{\nabla} f(\boldsymbol{x}_i)$. However, the way we choose $\boldsymbol{x}_{i+1}$ based on what we know so far will now be a little more involved than what we did for Gradient Descent.

To help us understand the algorithm, we are going to introduce two more sequences of iterates $\boldsymbol{y}_0, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_k \in \mathbb{R}^n$ and $\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k \in \mathbb{R}^n$.

The sequence of $\boldsymbol{y}_i$'s will be constructed to help us get as low a function value as possible at $f(\boldsymbol{y}_i)$, which we will consider our current solution and the last iterate $\boldsymbol{y}_k$ will be the output solution of our algorithm.

The sequence of $\boldsymbol{v}_i$'s will be constructed to help us get a lower bound on $f(\boldsymbol{x}^*)$.

By combining the upper bound on the function value of our current solution $f(\boldsymbol{y}_i)$ with a lower bound on the function value at an optimal solution $f(\boldsymbol{x}^*)$, we get an upper bound on the gap $f(\boldsymbol{y}_i) - f(\boldsymbol{x}^*)$ between the value of our solution and the optimal one. Finally, each iterate $\boldsymbol{x}_i$, which will be where we evaluate gradient $\boldsymbol{\nabla} f(\boldsymbol{x}_i)$, is chosen through a trade-off between wanting to reduce the upper bound and wanting to increase the lower bound.

The upper bound sequence: $\boldsymbol{y}_i$'s. The point $\boldsymbol{y}_i$ will be chosen from $\boldsymbol{x}_i$ to minimize an upper bound on f based at $\boldsymbol{x}_i$. This is similar to what we did in the previous section. We let $\boldsymbol{y}_i = \boldsymbol{x}_i + \boldsymbol{\delta}_i$ and choose $\boldsymbol{\delta}_i$ to minimize the upper bound $\boldsymbol{f}(\boldsymbol{y}_i) \leq f(\boldsymbol{x}_i) + \boldsymbol{\nabla} f(\boldsymbol{x}_i)^\top \boldsymbol{\delta}_i + \frac{\beta}{2} \|\boldsymbol{\delta}_i\|^2$, which gives us

$$\boldsymbol{y}_i = \boldsymbol{x}_i - \frac{1}{\beta} \boldsymbol{\nabla} f(\boldsymbol{x}_i) \text{ and } f(\boldsymbol{y}_i) \leq f(\boldsymbol{x}_i) - \frac{\|\boldsymbol{\nabla} f(\boldsymbol{x}_i)\|_2^2}{2\beta}.$$

We will introduce a notation for this upper bound

$$U_i = f(\boldsymbol{y}_i) \leq f(\boldsymbol{x}_i) - \frac{\|\boldsymbol{\nabla} f(\boldsymbol{x}_i)\|_2^2}{2\beta}. \tag{3.6}$$

Philosophizing about lower bounds[1]. A crucial ingredient to establishing an upper bound on gap_i was a lower bound on $f(\boldsymbol{x}^*)$.

[1]YMMV. People have a lot of different opionions about how to understand acceleration, and you should take my thoughts with a grain of salt.

In our analysis of Gradient Descent, in Equation (3.4), we used the lower bound $f(\boldsymbol{x}^*) \geq f(\boldsymbol{x}_i) - \|\boldsymbol{\nabla} f(\boldsymbol{x}_i)\|_2 \|\boldsymbol{x}_i - \boldsymbol{x}^*\|_2$. We can think of the Gradient Descent analysis as being based on a tension between two statements: Firstly that "a large gradient implies we quickly approach the optimum" and secondly "the function value gap to optimum cannot exceed the magnitude of the current gradient (scaled by distance to opt)".

This analysis does not use that we have seen many different function values and gradients, and each of these can be used to construct a lower bound on the optimum value $f(\boldsymbol{x}^*)$, and, in particular, it is not clear that the last gradient provides the best bound. To do better, we will try to use lower bounds that take advantage of all the gradients we have seen.

Definition 3.4.1. We will adopt a new notation for inner products that sometimes is more convenient when dealing with large expressions: $\langle \boldsymbol{a}, \boldsymbol{b} \rangle \stackrel{\text{def}}{=} \boldsymbol{a}^\top \boldsymbol{b}$.

The lower bound sequence: $\boldsymbol{v}_i$'s. We can introduce weights $a_i > 0$ for each step and combine the gradients we have observed into one lower bound based on a weighted average. Let us use $A_i = \sum_{j \leq i} a_j$ to denote the sum of the weights. Now a general lower bound on the function value at any $\boldsymbol{v} \in \mathbb{R}^n$ is :

$$f(\boldsymbol{v}) \geq \frac{1}{A_i} \sum_{j \leq i} a_j \left(f(\boldsymbol{x}_j) + \langle \boldsymbol{\nabla} f(\boldsymbol{x}_j), \boldsymbol{v} - \boldsymbol{x}_j \rangle \right)$$

However, to use Cauchy-Schwarz on each individual term here to instantiate this bound at $\boldsymbol{x}^*$ does not give us anything useful. Instead, we will employ a somewhat magical trick: we introduce a regularization term

$$\phi(\boldsymbol{v}) \stackrel{\text{def}}{=} \frac{\sigma}{2} \|\boldsymbol{v} - \boldsymbol{x}_0\|_2^2 .$$

We will choose the value $\sigma > 0$ later. Now we derive our lower bound L_i

$$\begin{aligned} f(\boldsymbol{x}^*) &\geq \frac{1}{A_i} \left(\phi(\boldsymbol{x}^*) + \sum_{j \leq i} a_j f(\boldsymbol{x}_j) + \langle a_j \boldsymbol{\nabla} f(\boldsymbol{x}_j), \boldsymbol{x}^* - \boldsymbol{x}_j \rangle \right) - \frac{\phi(\boldsymbol{x}^*)}{A_i} \\ &\geq \min_{\boldsymbol{v} \in \mathbb{R}^n} \left\{ \frac{1}{A_i} \left(\phi(\boldsymbol{v}) + \sum_{j \leq i} a_j f(\boldsymbol{x}_j) + \langle a_j \boldsymbol{\nabla} f(\boldsymbol{x}_j), \boldsymbol{v} - \boldsymbol{x}_j \rangle \right) \right\} - \frac{\phi(\boldsymbol{x}^*)}{A_i} \\ &= L_i \end{aligned}$$

We will let $\boldsymbol{v}_i$ be the $\boldsymbol{v}$ obtaining the minimum in the optimization problem appearing in the definition of L_i, so that

$$L_i = \frac{1}{A_i} \left(\phi(\boldsymbol{v}_i) + \sum_{j \leq i} a_i f(\boldsymbol{x}_i) + \langle a_i \boldsymbol{\nabla} f(\boldsymbol{x}_i), \boldsymbol{v}_i - \boldsymbol{x}_i \rangle \right) - \frac{\phi(\boldsymbol{x}^*)}{A_i}$$

How we will measure convergence. We have designed the upper bound U_i and the lower bound L_i such that $\text{gap}_i = f(\boldsymbol{y}_i) - f(\boldsymbol{x}^*) \leq U_i - L_i$.

As you will show in the exercises for Week 2, we can prove the convergence of Gradient Descent directly by an induction that establishes $1/\text{gap}_i \leq C \cdot i$ for some constant C depending on the Lipschitz gradient parameter β and the distance $\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2$.

To analyze Accelerated Gradient Descent, we will adopt a similar, but slightly different strategy, namely trying to show that $(U_i - L_i)r(i)$ is non-increasing for some positive "rate function" $r(i)$. Ideally $r(i)$ should grow quickly, which would imply that gap_i quickly gets small. We will also need to show that $(U_0 - L_0)r(0) \leq C$ for some constant C again depending on β and $\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2$. Then, we'll be able to conclude that

$$\text{gap}_i \cdot r(i) \leq (U_i - L_i)r(i) \leq (U_{i-1} - L_{i-1})r(i-1) \leq \cdots \leq (U_0 - L_0)r(0) \leq C,$$

and hence $\text{gap}_i \leq C/r(i)$.

This framework is fairly general. We could have also used it to analyze Gradient Descent, and it works for many other optimization algorithms too.

We are going to choose our rate function $r(i)$ to be exactly A_i, which of course is no accident! As we will see, this interacts nicely with our lower bound L_i. Hence, our goals are to

1. provide an upper bound on $A_0(U_0 - L_0)$,
2. and show that $A_{i+1}(U_{i+1} - L_{i+1}) \leq A_i(U_i - L_i)$,

Establishing the convergence rate. Let's start by looking at the change in the upper bound scaled by our rate function:

$$A_{i+1}U_{i+1} - A_iU_i = A_{i+1}\left(f(\boldsymbol{y}_{i+1}) - f(\boldsymbol{x}_{i+1})\right) - A_i\left(f(\boldsymbol{y}_i) - f(\boldsymbol{x}_{i+1})\right) + (A_{i+1} - A_i)f(\boldsymbol{x}_{i+1}) \tag{3.7}$$

$$\begin{aligned}
&\leq A_{i+1}\left(-\frac{\|\boldsymbol{\nabla} f(\boldsymbol{x}_{i+1})\|_2^2}{2\beta}\right) && \text{First term controlled by Equation (3.6).}\\
&\quad - A_i\left\langle \boldsymbol{\nabla} f(\boldsymbol{x}_{i+1}), \boldsymbol{y}_i - \boldsymbol{x}_{i+1}\right\rangle && \text{Second term bounded by Theorem 3.3.5.}\\
&\quad + a_{i+1}f(\boldsymbol{x}_{i+1}) && \text{Third term uses } a_{i+1} = A_{i+1} - A_i.
\end{aligned}$$

The solution $\boldsymbol{v}_i$ to the minimization in the lower bound L_i turns out to be relatively simple to characterize. By using derivatives to find the optimum, we first analyze the initial value of the lower bound L_0.

Claim 3.4.2.

1. $\boldsymbol{v}_0 = \boldsymbol{x}_0 - \frac{a_0}{\sigma}\boldsymbol{\nabla} f(\boldsymbol{x}_0)$
2. $L_0 = f(\boldsymbol{x}_0) - \frac{a_0}{2\sigma}\left\|\boldsymbol{\nabla} f(\boldsymbol{x}_0)\right\|_2^2 - \frac{\sigma}{2a_0}\left\|\boldsymbol{x}^* - \boldsymbol{x}_0\right\|_2^2$.

You will prove Claim 3.4.2 in Exercise 1 of the Week 3 exercise sheet. Noting $A_0 = a_0$, we see from Equation (3.6) and Part 2 of Claim 3.4.2, that

$$A_0(U_0 - L_0) \leq \left(\frac{a_0^2}{2\sigma} - \frac{a_0}{2\beta}\right)\left\|\boldsymbol{\nabla} f(\boldsymbol{x}_0)\right\|_2^2 + \frac{\sigma}{2}\left\|\boldsymbol{x}^* - \boldsymbol{x}_0\right\|_2^2 \tag{3.8}$$

It will be convenient to introduce notation for the rescaled lower bound $A_i L_i$ *without* optimizing over $\boldsymbol{v}$.

$$m_i(\boldsymbol{v}) = \phi(\boldsymbol{v}) - \phi(\boldsymbol{x}^*) + \sum_{j \leq i} a_j f(\boldsymbol{x}_j) + \langle a_j \boldsymbol{\nabla} f(\boldsymbol{x}_j), \boldsymbol{v} - \boldsymbol{x}_j \rangle$$

Thus $A_i L_i - A_{i+1} L_{i+1} = m_i(\boldsymbol{v}_i) - m_{i+1}(\boldsymbol{v})$. Now, it is not too hard to show the following relationships.

Claim 3.4.3.

1. $m_i(\boldsymbol{v}) = m_i(\boldsymbol{v}_i) + \frac{\sigma}{2}\left\|\boldsymbol{v} - \boldsymbol{v}_i\right\|_2^2$
2. $m_{i+1}(\boldsymbol{v}) = m_i(\boldsymbol{v}) + a_{i+1} f(\boldsymbol{x}_{i+1}) + \langle a_{i+1}\boldsymbol{\nabla} f(\boldsymbol{x}_{i+1}), \boldsymbol{v} - \boldsymbol{x}_{i+1}\rangle$
3. $\boldsymbol{v}_{i+1} = \boldsymbol{v}_i - \frac{a_{i+1}}{\sigma}\boldsymbol{\nabla} f(\boldsymbol{x}_{i+1})$

And again, you will prove Claim 3.4.3 in Exercise 2 (Week 3 sheet). *Hint for Part 1: note that $m_i(v)$ is a quadratic function, miminimized at v_i and its Hessian equals σI at all v.*

Given Claim 3.4.3, we see that

$$\begin{aligned} A_i L_i - A_{i+1} L_{i+1} & \\ &= m_i(\boldsymbol{v}_i) - m_{i+1}(\boldsymbol{v}_{i+1}) && (3.9)\\ &= -a_{i+1} f(\boldsymbol{x}_{i+1}) - \langle a_{i+1}\boldsymbol{\nabla} f(\boldsymbol{x}_{i+1}), \boldsymbol{v}_{i+1} - \boldsymbol{x}_{i+1}\rangle - \frac{\sigma}{2}\left\|\boldsymbol{v}_{i+1} - \boldsymbol{v}_i\right\|_2^2 && (3.10)\\ &= -a_{i+1} f(\boldsymbol{x}_{i+1}) - \langle a_{i+1}\boldsymbol{\nabla} f(\boldsymbol{x}_{i+1}), \boldsymbol{v}_i - \boldsymbol{x}_{i+1}\rangle + \frac{a_{i+1}^2}{2\sigma}\left\|\boldsymbol{\nabla} f(\boldsymbol{x}_{i+1})\right\|_2^2 && (3.11) \end{aligned}$$

This means that by combining Equation (3.7) and (3.11) we get

$$\begin{aligned} A_{i+1}(U_{i+1} - L_{i+1}) - A_i(U_i - L_i) \leq \left(\frac{-A_{i+1}}{2\beta} + \frac{a_{i+1}^2}{2\sigma}\right)\left\|\boldsymbol{\nabla} f(\boldsymbol{x}_{i+1})\right\|_2^2 & \\ + \langle \boldsymbol{\nabla} f(\boldsymbol{x}_{i+1}), A_{i+1}\boldsymbol{x}_{i+1} - a_{i+1}\boldsymbol{v}_i - A_i \boldsymbol{y}_i\rangle\,. & \end{aligned}$$

Now, this means that $A_{i+1}(U_{i+1} - L_{i+1}) - A_i(U_i - L_i) \leq 0$ if

$$A_{i+1}\boldsymbol{x}_{i+1} - a_{i+1}\boldsymbol{v}_i - A_i\boldsymbol{y}_i = \mathbf{0} \text{ and } A_{i+1}/\beta \geq a_{i+1}^2/\sigma$$

We can get this by letting $\boldsymbol{x}_{i+1} = \frac{A_i\boldsymbol{y}_i + a_{i+1}\boldsymbol{v}_i}{A_{i+1}}$, and $\sigma = \beta$ and $a_i = \frac{i+1}{2}$, which implies that $A_i = \frac{(i+1)(i+2)}{4} > a_i^2$.

By Equation (3.8), these parameter choices also imply that

$$A_0(U_0 - L_0) \leq \frac{\beta}{2}\left\|\boldsymbol{x}^* - \boldsymbol{x}_0\right\|_2^2.$$

Finally, by induction, we get $A_i(U_i - L_i) \leq \frac{\beta}{2}\left\|\boldsymbol{x}^* - \boldsymbol{x}_0\right\|_2^2$. Dividing through by A_i and using $\text{gap}_i \leq U_i - L_i$ results in the following theorem.

Theorem 3.4.4. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a β-gradient Lipschitz, convex function. Let $\boldsymbol{x}_0$ be a given starting point, and let $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$ be a minimizer of f.*

The Accelerated Gradient Descent algorithm given by

$$\begin{aligned}
a_i &= \frac{i+1}{2}, A_i = \frac{(i+1)(i+2)}{4} \\
\boldsymbol{v}_0 &= \boldsymbol{x}_0 - \frac{1}{2\beta}\boldsymbol{\nabla} f(\boldsymbol{x}_0) \\
\boldsymbol{y}_i &= \boldsymbol{x}_i - \frac{1}{\beta}\boldsymbol{\nabla} f(\boldsymbol{x}_i) \\
\boldsymbol{x}_{i+1} &= \frac{A_i\boldsymbol{y}_i + a_{i+1}\boldsymbol{v}_i}{A_{i+1}} \\
\boldsymbol{v}_{i+1} &= \boldsymbol{v}_i - \frac{a_{i+1}}{\beta}\boldsymbol{\nabla} f(\boldsymbol{x}_{i+1})
\end{aligned}$$

ensures that the kth iterate satisfies

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) \leq \frac{2\beta\left\|\boldsymbol{x}_0 - \boldsymbol{x}^*\right\|_2^2}{(k+1)(k+2)}.$$

Bibliography

[DO19] Jelena Diakonikolas and Lorenzo Orecchia. Conjugate gradients and accelerated methods unified: The approximate duality gap view. *arXiv preprint arXiv:1907.00289*, 2019.

[HS+52] Magnus R Hestenes, Eduard Stiefel, et al. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.

[Lan52] Cornelius Lanczos. Solution of systems of linear equations by minimized iterations. *J. Res. Nat. Bur. Standards*, 49(1):33–53, 1952.

[Nes83] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983.

[NY83] A Nemirovski and D Yudin. Information-based complexity of mathematical programming. *Izvestia AN SSSR, Ser. Tekhnicheskaya Kibernetika (the journal is translated to English as Engineering Cybernetics. Soviet J. Computer & Systems Sci.)*, 1, 1983.