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.

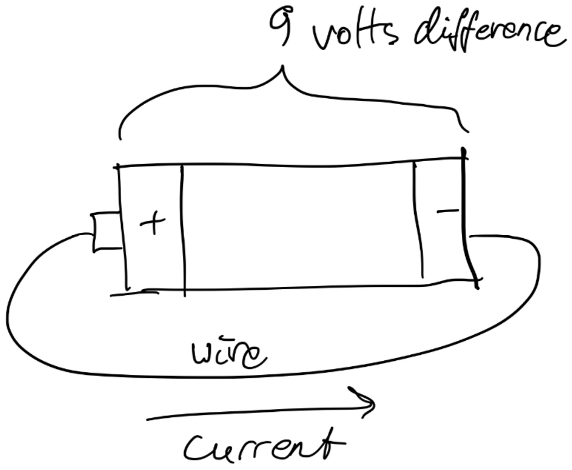


Figure 1.1: A 9 volts battery with a wire attached.

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.

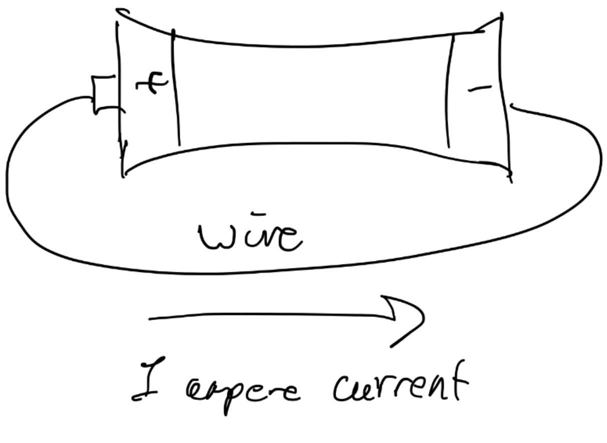


Figure 1.2: A 1 ampere battery with a wire attached.

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.

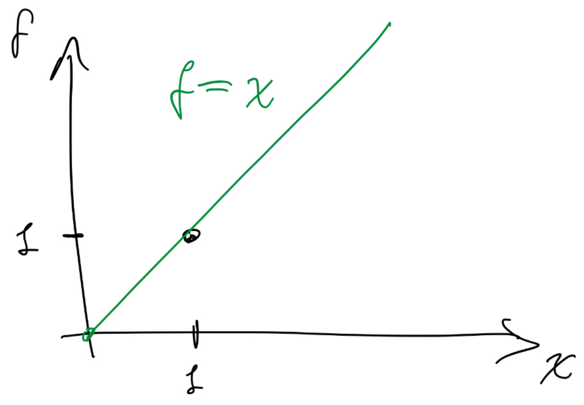


Figure 1.3: Ohm's Law for a resistor with resistance $r = 1$.

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

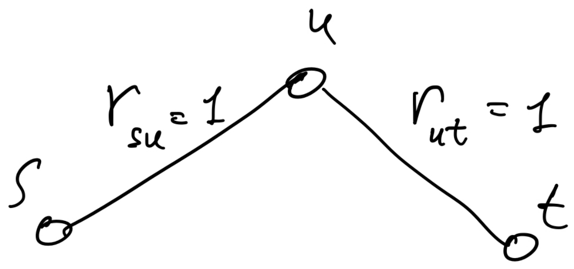


Figure 1.4: A path of two resistors.

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 $x = 0$.

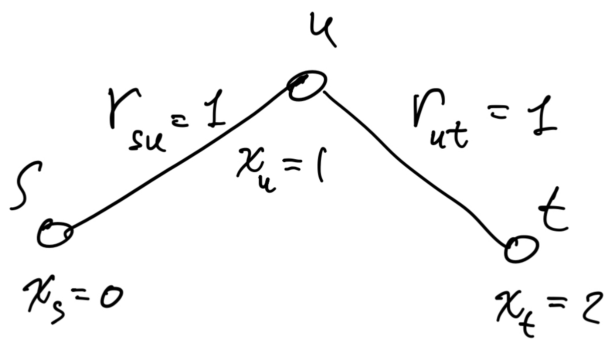


Figure 1.5: A path of two resistors.

Let us try one more example:

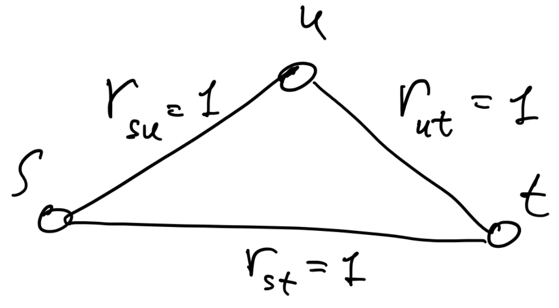


Figure 1.6: A network with three resistors.

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

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

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 $\boldsymbol{R} = \operatorname{diag}_{e \in E} \boldsymbol{r}(e)$ 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".

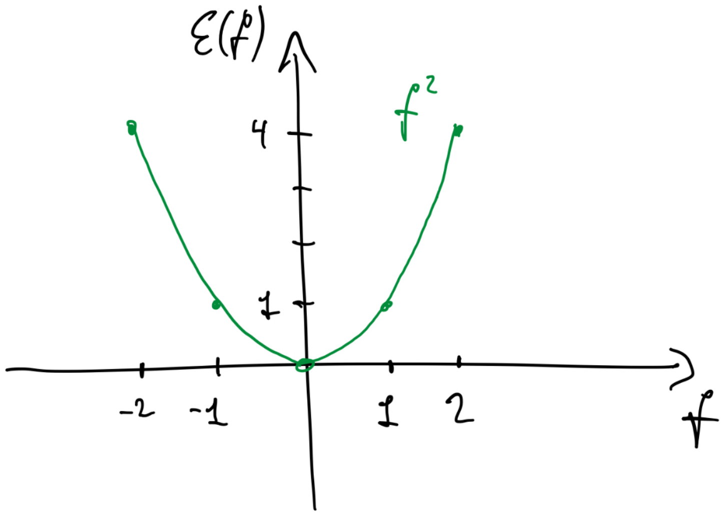


Figure 1.7: Energy as a function of flow in a resistor with resistance $r = 1$.

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:

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

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. $\boldsymbol{x}_0 = \boldsymbol{0}$, 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 $c(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Lx} - \boldsymbol{x}^\top \boldsymbol{d}$, 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 $\boldsymbol{Lx}^* = \boldsymbol{d}$. Now, we also need to make sure that $c(\boldsymbol{x}_t) \approx c(\boldsymbol{x}^*)$ implies that $\boldsymbol{Lx}_t \approx \boldsymbol{d}$ 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. $\boldsymbol{Lx} = \boldsymbol{d}$ 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.

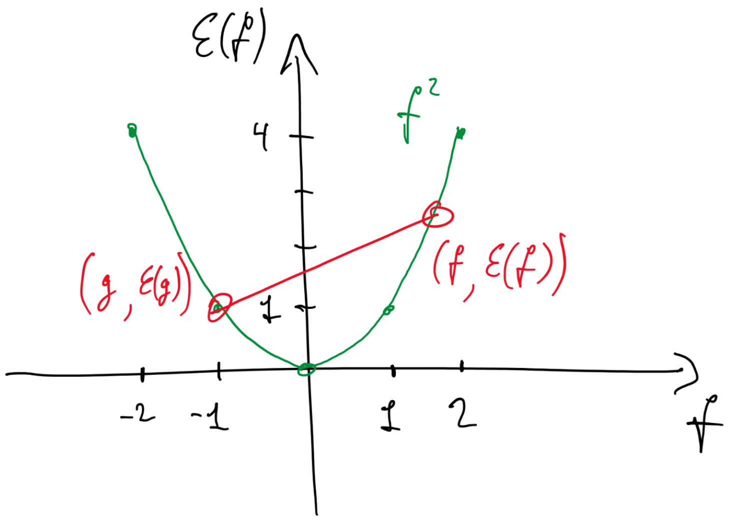


Figure 1.8: Energy as a function of flow in a resistor with resistance $r = 1$. The function is convex.

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

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

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.

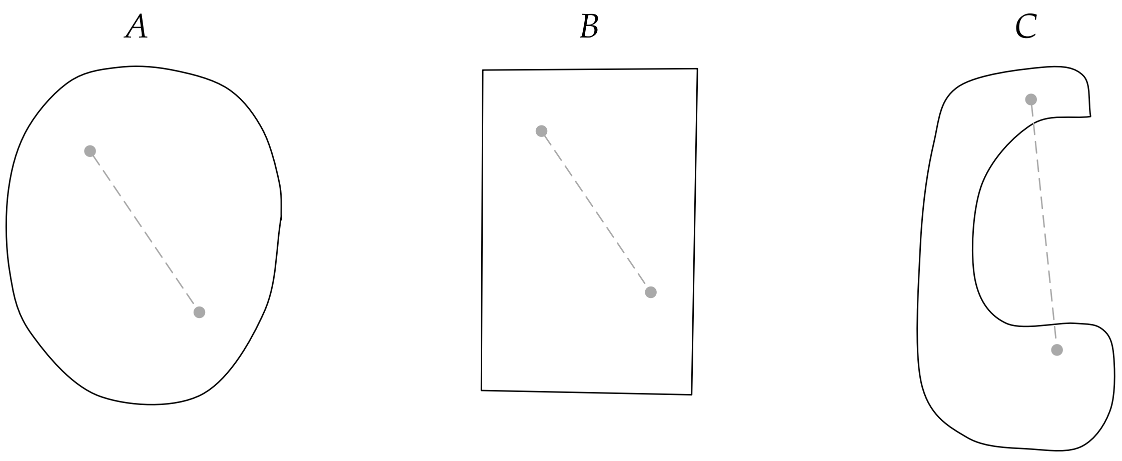


Figure 1.9: A depiction of convex and non-convex sets. The sets A and B are convex since the straight line between any two points inside them is also in the set. The set C is not convex.

$\boldsymbol{d}$ 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 $\boldsymbol{c} \in \mathbb{R}^E \geq \boldsymbol{0}$ and try to minimize $\left\lVert \boldsymbol{C}^{-1}\boldsymbol{f} \right\rVert_\infty$, where $\boldsymbol{C} = \operatorname{diag}_{e \in E} \boldsymbol{c}(e)$. Then our problem becomes

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

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

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

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 $\boldsymbol{c} \leq n^{100}\boldsymbol{1}$.

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

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

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**.

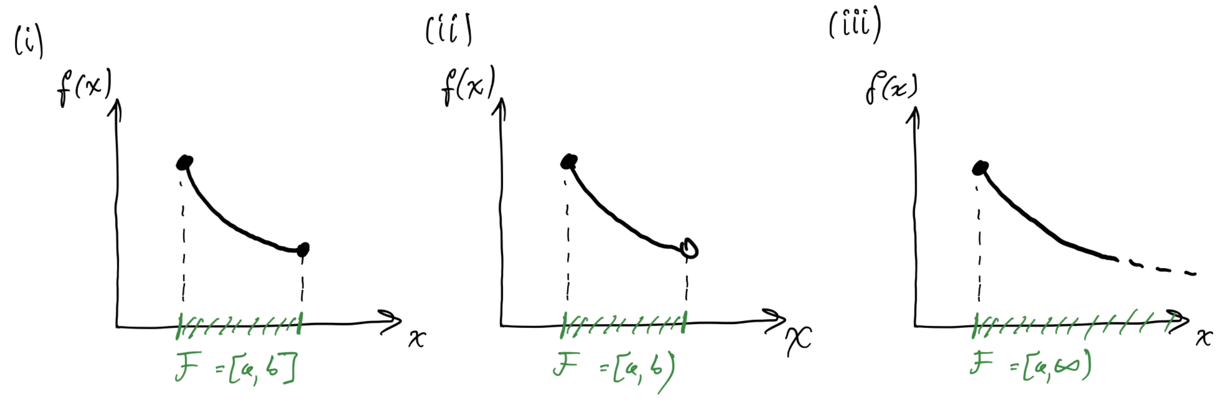


Figure 2.1

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.

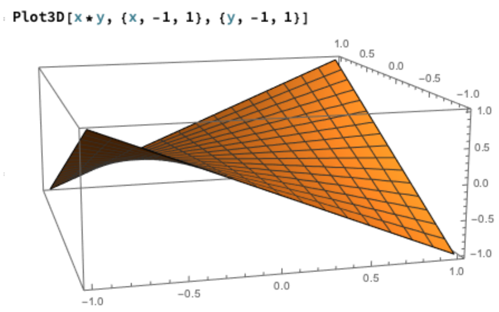
2.3 A Characterization of Convex Functions

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:

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



```
Plot3D[x*y, {x, -1, 1}, {y, -1, 1}]
```

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

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

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.

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

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].

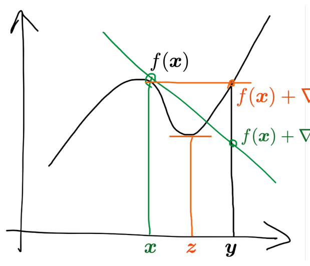
Remark. In this course, we will generally err on the side of being informal about functional analysis when we can afford to, and we will not worry too much about the details of different notions of differentiability (e.g. Fréchet and Gateaux differentiability), except when it turns out to be important.

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.



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

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

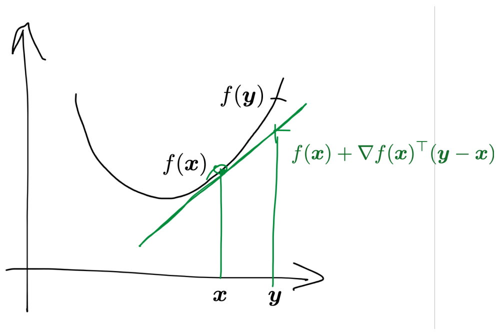

$f(\boldsymbol{y})$
$f(\boldsymbol{x})$
$f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y}-\boldsymbol{x})$
$\boldsymbol{x}$
$\boldsymbol{y}$

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. $\square$

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 $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$ 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 $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$.*

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 $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$.

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 $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$ 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 $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. 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 $\lambda_1(\boldsymbol{A}) \left\|\boldsymbol{x}\right\|_2^2$.

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 $\left\|\boldsymbol{x}\right\|_2 = 1$. Then $\boldsymbol{x} = \sum_{j=1}^{i} \boldsymbol{c}(j)\boldsymbol{v}_j$ for some vector $\boldsymbol{c} \in \mathbb{R}^i$ with $\left\|\boldsymbol{c}\right\|_2 = 1$.

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 $\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^\top\boldsymbol{V}^\top\boldsymbol{\Lambda}\boldsymbol{V}\boldsymbol{x} = (\boldsymbol{V}\boldsymbol{x})^\top\boldsymbol{\Lambda}(\boldsymbol{V}\boldsymbol{x}) = \sum_{j=1}^{i} \lambda_j \boldsymbol{c}(j)^2 \leq \lambda_i \left\|\boldsymbol{c}\right\|_2^2 = \lambda_i$. 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 $T = \operatorname{span}\{\boldsymbol{v}_i, \ldots, \boldsymbol{v}_n\}$. 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:

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

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

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

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

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 $\boldsymbol{x}$:

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

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

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

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

And because f is convex, we know that $\boldsymbol{\nabla} f(\boldsymbol{x}_0) \neq \boldsymbol{0}$ 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 $\boldsymbol{x}, \boldsymbol{y} \in S$

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

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 $\boldsymbol{0} = \boldsymbol{\nabla}_{\boldsymbol{\delta}} \left(f(\boldsymbol{x}_0) + \boldsymbol{\nabla} f(\boldsymbol{x}_0)^\top \boldsymbol{\delta} + \frac{\beta}{2} \|\boldsymbol{\delta}\|^2 \right) = \boldsymbol{\nabla} f(\boldsymbol{x}_0) + \beta \boldsymbol{\delta}$, 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

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

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 $\|\boldsymbol{\nabla} f(\boldsymbol{x}_i)\|_2$, 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*

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

ensures that the kth 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 $\boldsymbol{f}(\boldsymbol{y}_i) \leq f(\boldsymbol{x}_i) + \boldsymbol{\nabla} f(\boldsymbol{x}_i)^\top \boldsymbol{\delta}_i + \frac{\beta}{2} \left\lVert \boldsymbol{\delta}_i \right\rVert^2$, which gives us

$$\boldsymbol{y}_i = \boldsymbol{x}_i - \frac{1}{\beta} \boldsymbol{\nabla} f(\boldsymbol{x}_i) \text{ and } f(\boldsymbol{y}_i) \leq f(\boldsymbol{x}_i) - \frac{\left\lVert \boldsymbol{\nabla} f(\boldsymbol{x}_i) \right\rVert_2^2}{2\beta}.$$

We will introduce a notation for this upper bound

$$U_i = f(\boldsymbol{y}_i) \leq f(\boldsymbol{x}_i) - \frac{\left\lVert \boldsymbol{\nabla} f(\boldsymbol{x}_i) \right\rVert_2^2}{2\beta}. \tag{3.6}$$

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 $f(\boldsymbol{x}^*) \geq f(\boldsymbol{x}_i) - \left\|\boldsymbol{\nabla} f(\boldsymbol{x}_i)\right\|_2 \left\|\boldsymbol{x}_i - \boldsymbol{x}^*\right\|_2$. 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

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

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

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

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

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

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.

[Spi19] Daniel A Spielman. Spectral and Algebraic Graph Theory, 2019.

Part II

Spectral Graph Theory

Chapter 4

Introduction to Spectral Graph Theory

In this chapter, we will study graphs through linear algebra. This approach is known as Spectral Graph Theory and turns out to be surprisingly powerful. An in-depth treatment of many topics in this area can be found in [Spi19].

4.1 Recap: Incidence and Adjacency Matrices, the Laplacian Matrix and Electrical Energy

To exploit tools from linear algebra, we first have to introduce some interesting matrices that capture important properties of the graph G. In Chapter 1, we have already seen some of these matrices. Let us recall them and also revisit some additional material from the lectures before. For the rest of the lecture, we let $G = (V, E, \boldsymbol{w})$ be an undirected graph, with $n = |V|$ vertices and $m = |E|$ edges, where $\boldsymbol{w} \in \mathbb{R}_+^E$ assigns a positive weight for every edge. We also assume that G is connected!

Fundamental Matrices. We start by re-introducing the *edge-vertex incidence matrix* $\boldsymbol{B}$ of the graph G. Therefore, we first have to associate an arbitrary direction to every edge. We then let $\boldsymbol{B} \in \mathbb{R}^{V \times E}$.

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

The edge directions are only there to help us track the meaning of signs of quantities defined on edges: The math we do should not depend on the choice of sign.

We define the *weight matrix* $\boldsymbol{W} \in \mathbb{R}^{E \times E}$ be the diagonal matrix given by $\boldsymbol{W} = \operatorname{diag}(\boldsymbol{w})$, i.e $\boldsymbol{W}(e, e) = \boldsymbol{w}(e)$. The weighted degree of a vertex is defined as $\boldsymbol{d}(v) = \sum_{\{u,v\} \in E} \boldsymbol{w}(u, v)$.

Again we treat the edges as undirected. Let $\boldsymbol{D} = \operatorname{diag}(\boldsymbol{d})$ be the *degree matrix* that is a diagonal matrix in $\mathbb{R}^{V \times V}$ with weighted degrees on the diagonal.

We define the *weighted adjacency matrix* $\boldsymbol{A} \in \mathbb{R}^{V \times V}$ of a graph by

$$\boldsymbol{A}(u,v) = \begin{cases} \boldsymbol{w}(u,v) & \text{if } \{u,v\} \in E \\ 0 & \text{otherwise.} \end{cases}$$

Note that we treat the edges as undirected here, so $\boldsymbol{A}^\top = \boldsymbol{A}$.

Finally, we define the *Laplacian* of the graph as $\boldsymbol{L} = \boldsymbol{B}\boldsymbol{W}\boldsymbol{B}^\top$. Note that in the first chapter, we defined the Laplacian as $\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top$, where $\boldsymbol{R}$ is the diagonal matrix with edge resistances on the diagonal. We want to think of high *weight* on an edge as expressing that two vertices are highly connected, whereas we think of high resistance on an edge as expressing that the two vertices are poorly connected, so we let $\boldsymbol{w}(e) = 1/\boldsymbol{R}(e,e)$. In Problem Set 1, you showed that $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, and that for $\boldsymbol{x} \in \mathbb{R}^V$,

$$\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} = \sum_{\{a,b\} \in E} \boldsymbol{w}(a,b)(\boldsymbol{x}(a) - \boldsymbol{x}(b))^2.$$

Flows and Voltages. Let us next briefly review flows and voltages. Given the edge-vertex incidence matrix $\boldsymbol{B}$, we can now express the net flow constraint that $\boldsymbol{f}$ routes a demand $\boldsymbol{d}$ 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 $u \in V \setminus \{s,t\}$. Note that, unfortunately, the demand vector is typically denoted by $\boldsymbol{d}$, but so is the degree vector. Since it should always be clear from context which is which, we will also overload $\boldsymbol{d}$ in this way!

We then define for any flow $\boldsymbol{f} \in \mathbb{R}^E$ the electrical energy by

$$\mathcal{E}(\boldsymbol{f}) = \sum_e \boldsymbol{r}(e)\boldsymbol{f}(e)^2 = \boldsymbol{f}^\top \boldsymbol{R}\boldsymbol{f}$$

and for any electrical voltages $\boldsymbol{x} \in \mathbb{R}^E$

$$\mathcal{E}(\boldsymbol{f}) = \boldsymbol{f}^\top \boldsymbol{R}\boldsymbol{f} = \boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}.$$

While the former definition is standard, the second definition is motivated by the fact that by Ohm's law, where we use $\boldsymbol{R} = \operatorname{diag}_{e \in E} \boldsymbol{r}(e)$, we have that electrical voltages $\boldsymbol{x}$ will induce an electrical flow $\boldsymbol{f} = \boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{x}$. And for such a pair $\boldsymbol{f}$ and $\boldsymbol{x}$, we then have that the energy associated with $\boldsymbol{x}$ is equal to the energy consumed by the flow $\boldsymbol{f}$ induced by $\boldsymbol{x}$:

$$\mathcal{E}(\boldsymbol{f}) = \boldsymbol{f}^\top \boldsymbol{R}\boldsymbol{f} = \boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} = \mathcal{E}(\boldsymbol{x}).$$

The Courant-Fisher Theorem. Let us also recall the Courant-Fischer theorem, which we proved in Chapter 3 (Theorem 3.1.4). Here we add an additional characterization of λ_i. Note that the second equality for both versions to obtain λ_i can be extracted from our proof of the Courant-Fischer theorem in Chapter 3.

Theorem 4.1.1 (The Courant-Fischer Theorem, eigenbasis version)**.** *Let $\boldsymbol{A}$ be a symmetric matrix in $\mathbb{R}^{n\times n}$, with eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$, and corresponding eigenvectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ which form an othernormal basis. Then*

1.

$$\lambda_i = \min_{\substack{\text{subspace } W \subseteq \mathbb{R}^n \\ \dim(W)=i}} \max_{\boldsymbol{x}\in W, \boldsymbol{x}\neq \boldsymbol{0}} \frac{\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \min_{\substack{\boldsymbol{x}\perp \boldsymbol{x}_1,\ldots \boldsymbol{x}_{i-1} \\ \boldsymbol{x}\neq \boldsymbol{0}}} \frac{\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$$

2.

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

Of course, we also have $\lambda_i(\boldsymbol{A}) = \frac{\boldsymbol{x}_i^\top \boldsymbol{A}\boldsymbol{x}_i}{\boldsymbol{x}_i^\top \boldsymbol{x}_i}$.

4.2 Understanding Eigenvalues of the Laplacian

We would like to understand the eigenvalues of the Laplacian matrix of a graph.

But first, why should we care? It turns out that Laplacian eigenvalues can help us understand many properties of a graph. But we are going to start with a simple motivating observation: Electrical voltages $\boldsymbol{x} \in \mathbb{R}^V$ consume electrical energy $\mathcal{E}(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}$. This means that by the Courant-Fischer Theorem

$$\mathcal{E}(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} \leq \lambda_n(L)\boldsymbol{x}^\top \boldsymbol{x}$$

And, for any voltages $\boldsymbol{x} \perp \boldsymbol{1}$,

$$\mathcal{E}(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} \geq \lambda_2(L)\boldsymbol{x}^\top \boldsymbol{x}.$$

Thus, we can use the eigenvalues to give upper and lower bounds on how much electrical energy will be consumed by the flow induced by $\boldsymbol{x}$, in terms compared to $\boldsymbol{x}^\top \boldsymbol{x} = \|\boldsymbol{x}\|_2^2$.

In a couple of chapters, we will also prove the following claim, which shows that the Laplacian eigenvalues can directly tell us about the electrical energy that is required to route a given demand.

Claim 4.2.1. *Given a demand vector $\boldsymbol{d} \in \mathbb{R}^V$ such that $\boldsymbol{d} \perp \boldsymbol{1}$, the electrical voltages $\boldsymbol{x}$ that route $\boldsymbol{d}$ satisfy $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{d}$ and the electrical energy of these voltages satisfies*

$$\frac{\|\boldsymbol{d}\|_2^2}{\lambda_n} \leq \mathcal{E}(\boldsymbol{x}) \leq \frac{\|\boldsymbol{d}\|_2^2}{\lambda_2}$$

While the techniques in this section can be used to analyze various Laplacian matrices, we focus on understanding the eigenvalues of certain graph families that are particularly instructive. In this section, we obtain the following results.

Graph Family	λ_2 lower bound	λ_2 upper bound	λ_n lower bound	λ_n upper bound
Complete Graph K_n	n	n	n	n
Path Graph P_n	$\frac{1}{n^2}$	$\frac{12}{n^2}$	1	4
Binary Tree T_n	$\frac{1}{2n\log_2(n)}$	$\frac{2}{n-1}$	1	6

4.3 The Complete Graph

To get a sense of how Laplacian eigenvalues behave, let us start by considering the n-vertex complete graph with unit weights, which we denote by K_n. The adjacency matrix of K_n is $\boldsymbol{A} = \mathbf{1}\mathbf{1}^\top - \boldsymbol{I}$, since it has ones everywhere, except for the diagonal, where entries are zero. The degree matrix $\boldsymbol{D} = (n-1)\boldsymbol{I}$. Thus the Laplacian is $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} = n\boldsymbol{I} - \mathbf{1}\mathbf{1}^\top$.

Thus for any $\boldsymbol{y} \perp \mathbf{1}$, we have $\boldsymbol{y}^\top \boldsymbol{L}\boldsymbol{y} = n\boldsymbol{y}^\top\boldsymbol{y} - (\mathbf{1}^\top\boldsymbol{y})^2 = n\boldsymbol{y}^\top\boldsymbol{y}$.

From this, we can conclude that any $\boldsymbol{y} \perp \mathbf{1}$ is an eigenvector of eigenvalue n, and that all $\lambda_2 = \lambda_3 = \ldots = \lambda_n = n$.

4.4 The Path Graph

Next, let us try to understand λ_2 and λ_n for P_n, the n vertex path graph with unit weight edges. I.e. the graph has edges $E = \{\{i, i+1\} \text{ for } i = 1 \text{ to } (n-1)\}$. This is in a sense the least well-connected unit weight graph on n vertices, whereas K_n is the most well-connected.

4.4.1 Upper Bounding $\lambda_2(P_n)$ via Test Vectors

We can use the eigenbasis version of the Courant-Fisher theorem to observe that the second-smallest eigenvalue of the Laplacian is given by

$$\lambda_2(\boldsymbol{L}) = \min_{\substack{\boldsymbol{x}\neq\mathbf{0} \\ \boldsymbol{x}^\top\mathbf{1}=0}} \frac{\boldsymbol{x}^\top\boldsymbol{L}\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}. \tag{4.1}$$

We can get a better understanding of this particular case through a couple of simple observations. Suppose $\boldsymbol{x} = \boldsymbol{y} + \alpha\mathbf{1}$, where $\boldsymbol{y} \perp \mathbf{1}$. Then $\boldsymbol{x}^\top\boldsymbol{L}\boldsymbol{x} = \boldsymbol{y}^\top\boldsymbol{L}\boldsymbol{y}$, and $\|\boldsymbol{x}\|_2^2 = \|\boldsymbol{y}\|^2 + \alpha^2\|\mathbf{1}\|^2$.

So for any given vector, you can increase the value of $\frac{\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$, by replacing $\boldsymbol{x}$ with its component orthogonal to $\mathbf{1}$, which we denoted by $\boldsymbol{y}$.

We can conclude from Equation (4.1) that for *any* vector $\boldsymbol{y} \perp \mathbf{1}$,

$$\lambda_2 \leq \frac{\boldsymbol{y}^\top \boldsymbol{L}\boldsymbol{y}}{\boldsymbol{y}^\top \boldsymbol{y}}$$

When we use a vector $\boldsymbol{y}$ in this way to prove a bound on an eigenvalue, we call it a *test vector.*

Now, we'll use a test vector to give an upper bound on $\lambda_2(\boldsymbol{L}_{P_n})$. Let $\boldsymbol{x} \in \mathbb{R}^V$ be given by $\boldsymbol{x}(i) = (n+1) - 2i$, for $i \in [n]$. This vector satisfies $\boldsymbol{x} \perp \mathbf{1}$. We picked this because we wanted a sequence of values growing linearly along the path, while also making sure that the vector is orthogonal to $\mathbf{1}$. Now

$$\begin{aligned}
\lambda_2(\boldsymbol{L}_{P_n}) &\leq \frac{\sum_{i\in[n-1]} (\boldsymbol{x}(i) - \boldsymbol{x}(i+1))^2}{\sum_{i=1}^n \boldsymbol{x}(i)^2} \\
&= \frac{\sum_{i=1}^{n-1} 2^2}{\sum_{i=1}^n (n+1-2i)^2} \\
&= \frac{4(n-1)}{(n+1)n(n-1)/3} \\
&= \frac{12}{n(n+1)} \leq \frac{12}{n^2}.
\end{aligned}$$

Later, we will prove a lower bound that shows this value is right up to a constant factor. But the test vector approach based on the Courant-Fischer theorem doesn't immediately work when we want to prove lower bounds on $\lambda_2(\boldsymbol{L})$.

4.4.2 Lower Bounding $\lambda_n(P_n)$ via Test Vectors

We can see from either version of the Courant-Fischer theorem that

$$\lambda_n(\boldsymbol{L}) = \max_{v \neq \mathbf{0}} \frac{\boldsymbol{v}^\top \boldsymbol{L}\boldsymbol{v}}{\boldsymbol{v}^\top \boldsymbol{v}}. \tag{4.2}$$

Thus for *any* vector $\boldsymbol{y} \neq 0$,

$$\lambda_n \geq \frac{\boldsymbol{y}^\top \boldsymbol{L}\boldsymbol{y}}{\boldsymbol{y}^\top \boldsymbol{y}}.$$

This means we can get a test vector-based lower bound on λ_n. Let us apply this to the Laplacian of P_n. We'll try the vector $\boldsymbol{x} \in \mathbb{R}^V$ given by $\boldsymbol{x}(1) = -1$, and $\boldsymbol{x}(n) = 1$ and $\boldsymbol{x}(i) = 0$ for $i \neq 1, n$.

Here we get

$$\lambda_n(\boldsymbol{L}_{P_n}) \geq \frac{\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{2}{2} = 1. \tag{4.3}$$

Again, it's not clear how to use the Courant-Fischer theorem to prove an upper bound on $\lambda_n(\boldsymbol{L})$. But, later we'll see how to prove an upper that shows that for P_n, the lower bound we obtained is right up to constant factors.

4.4.3 Upper bounding $\lambda_n(P_n)$ via the Loewner Order

In the previous sections, we first saw a complete characterization of the eigenvalues and eigenvectors of the unit weight complete graph on n vertices, K_n. Namely, $\boldsymbol{L}_{K_n} = n\boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^\top$, and this means that *every* vector $\boldsymbol{y} \perp \boldsymbol{1}$ is an eigenvector of eigenvalue n. Ideally, we would like to prove an almost matching lower bound on $\lambda_2(P_n)$ and an almost matching upper bound on $\lambda_n(P_n)$, but it is not clear how to get that from the Courant-Fischer theorem.

To get there, we need to introduce some more tools. We now introduce an ordering on symmetric matrices called the *Loewner order*, which I also like to just call the positive semi-definite order. As we will see in a moment, it is a partial order on symmetric matrices, we denote it by "$\preceq$". For convenience, we allow ourselves to both write $\boldsymbol{A} \preceq \boldsymbol{B}$ and equivalently $\boldsymbol{B} \succeq \boldsymbol{A}$.

The Loewner Order, aka. the Positive Semi-Definite Order. For a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ we define that

$$\boldsymbol{A} \succeq \boldsymbol{0}$$

if and only if $\boldsymbol{A}$ is positive semi-definite.

More generally, when we have two symmetric matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$, we will write

$$\boldsymbol{A} \preceq \boldsymbol{B} \text{ if and only if for all } \boldsymbol{x} \in \mathbb{R}^n \text{ we have } \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x} \tag{4.4}$$

This is a partial order, because it satisfies the three requirements of

1. Reflexivity: $\boldsymbol{A} \preceq \boldsymbol{A}$.
2. Anti-symmetry: $\boldsymbol{A} \preceq \boldsymbol{B}$ and $\boldsymbol{B} \preceq \boldsymbol{A}$ implies $\boldsymbol{A} = \boldsymbol{B}$
3. Transitivity: $\boldsymbol{A} \preceq \boldsymbol{B}$ and $\boldsymbol{B} \preceq \boldsymbol{C}$ implies $\boldsymbol{A} \preceq \boldsymbol{C}$

Check for yourself that these properties hold!

The PSD order has other very useful properties: $\boldsymbol{A} \preceq \boldsymbol{B}$ implies $\boldsymbol{A} + \boldsymbol{C} \preceq \boldsymbol{B} + \boldsymbol{C}$ for any symmetric matrix $\boldsymbol{C}$. Convince yourself of this too!

And, combining this observation with transitivity, we can see that $\boldsymbol{A} \preceq \boldsymbol{B}$ and $\boldsymbol{C} \preceq \boldsymbol{D}$ implies $\boldsymbol{A} + \boldsymbol{C} \preceq \boldsymbol{B} + \boldsymbol{D}$.

Here is another useful property: If $\boldsymbol{0} \preceq \boldsymbol{A}$ then for all $\alpha \geq 1$

$$\frac{1}{\alpha}\boldsymbol{A} \preceq \boldsymbol{A} \preceq \alpha\boldsymbol{A}.$$

Here is another one:

Claim 4.4.1. *If $\boldsymbol{A} \preceq \boldsymbol{B}$, then for all i*

$$\lambda_i(\boldsymbol{A}) \leq \lambda_i(\boldsymbol{B}).$$

Proof. We can prove this Claim by applying the subspace version of the Courant-Fischer theorem.

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

□

Note that the converse of Clam 4.4.1 is very much false, for example the matrices $\boldsymbol{A} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ and $\boldsymbol{B} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ have equal eigenvalues, but both $\boldsymbol{A} \not\preceq \boldsymbol{B}$ and $\boldsymbol{B} \not\preceq \boldsymbol{A}$.

Remark 4.4.2. It's useful to get used to and remember some of the properties of the Loewner order, but all the things we have established so far are almost immediate from the basic characterization in Equation (4.4). So, ideally, don't memorize all these facts, instead, try to see that they are simple consequences of the definition.

Upper bounding $\lambda_n(L_G)$ Using Degrees. In an earlier chapter, we observed that for any graph $G = (V, E, \boldsymbol{w})$, $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \succeq \boldsymbol{0}$. We can see this from $\boldsymbol{x}^\top(\boldsymbol{D} - \boldsymbol{A})\boldsymbol{x} = \sum_{(u,v)\in E} \boldsymbol{w}(u,v)(\boldsymbol{x}(u) - \boldsymbol{x}(v))^2 \geq 0$. Similarly $\boldsymbol{D} + \boldsymbol{A} \succeq \boldsymbol{0}$. because $\boldsymbol{x}^\top(\boldsymbol{D} + \boldsymbol{A})\boldsymbol{x} = \sum_{(u,v)\in E} \boldsymbol{w}(u,v)(\boldsymbol{x}(u) + \boldsymbol{x}(v))^2 \geq 0$. But this means that $-\boldsymbol{A} \preceq \boldsymbol{D}$ and hence $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \preceq 2\boldsymbol{D}$.

So, for the path graph P_n, we have $\boldsymbol{L}_{P_n} \preceq \boldsymbol{D} - \boldsymbol{A} \preceq 2\boldsymbol{D} \preceq 4\boldsymbol{I}$. So by Claim 4.4.1

$$\lambda_n(\boldsymbol{L}_{P_n}) \leq 4. \tag{4.5}$$

We can see that our test vector-based lower bound on $\lambda_n(\boldsymbol{L}_{P_n})$ from Equation (4.3) is tight up to a factor 4.

Since this type of argument works for any unit weight graph, it proves the following claim.

Claim 4.4.3. *For any unit weight graph G, $\lambda_n(\boldsymbol{L}_G) \leq 2\max_{v\in V} \boldsymbol{d}(v)$.*

This is tight on a graph consisting of a single edge.

4.4.4 Lower bounding $\lambda_2(P_n)$ via the Loewner Order

We will now conclude our analysis of the eigenvalues λ_2 and λ_n of the path graph P_n by finally deriving a lower bound on $\lambda_2(P_n)$. Again, this requires us to use new tools. As in the last section, we first introduce the new tool for general graphs and then exploit the tool to derive the lower bound.

The Loewner Order for Graphs. It's sometimes convenient to overload the PSD order to also apply to graphs. We will write

$$G \preceq H$$

if $\boldsymbol{L}_G \preceq \boldsymbol{L}_H$.

For example, given two unit weight graphs $G = (V, E)$ and $H = (V, F)$, if H is a subgraph of G, then

$$\boldsymbol{L}_H \preceq \boldsymbol{L}_G.$$

We can see this from the Laplacian quadratic form:

$$\boldsymbol{x}^\top \boldsymbol{L}_G \boldsymbol{x} = \sum_{(u,v)\in E} \boldsymbol{w}(u,v)(\boldsymbol{x}(u) - \boldsymbol{x}(v))^2.$$

Dropping edges will only decrease the value of the quadratic form. The same is for decreasing the weights of edges. The graph order notation is especially useful when we allow for scaling a graph by a constant, say $c > 0$,

$$c \cdot H \preceq G$$

What is $c \cdot H$? It is the same graph as H, but the weight of every edge is multiplied by c. Now we can make statements like $\frac{1}{2}H \preceq G \preceq 2H$, which turn out to be useful notion of the two graphs approximating each other.

The Path Inequality. Now, we'll see a general tool for comparing two graphs G and H to prove inequalities like $cH \preceq G$ for some constant c. Our tools won't necessarily work well for all cases, but we'll see some examples where they do.

In the rest of the chapter, we will often need to compare two graphs defined on the same vertex set $V = \{1, \dots, n\} = [n]$.

We use $G_{i,j}$ to denote the unit weight graph on vertex set $[n]$ consisting of a single edge between vertices i and j.

Lemma 4.4.4 (The Path Inequality)**.**

$$(n-1) \cdot P_n \succeq G_{1,n},$$

Proof. We want to show that for every $\boldsymbol{x} \in \mathbb{R}^n$,

$$(n-1) \cdot \sum_{i=1}^{n-1} (\boldsymbol{x}(i+1) - \boldsymbol{x}(i))^2 \geq (\boldsymbol{x}(n) - \boldsymbol{x}(1))^2.$$

For $i \in [n-1]$, set

$$\boldsymbol{\Delta}(i) = \boldsymbol{x}(i+1) - \boldsymbol{x}(i).$$

The inequality we want to prove then becomes

$$(n-1) \sum_{i=1}^{n-1} (\boldsymbol{\Delta}(i))^2 \geq \left(\sum_{i=1}^{n-1} \boldsymbol{\Delta}(i) \right)^2 .$$

But, this is immediate from the Cauchy-Schwarz inequality $\boldsymbol{a}^\top \boldsymbol{b} \leq \|\boldsymbol{a}\|_2 \|\boldsymbol{b}\|_2$:

$$\begin{aligned}
(n-1) \sum_{i=1}^{n-1} (\boldsymbol{\Delta}(i))^2 &= \|\boldsymbol{1}_{n-1}\|^2 \cdot \|\boldsymbol{\Delta}\|^2 \\
&= (\|\boldsymbol{1}_{n-1}\| \cdot \|\boldsymbol{\Delta}\|)^2 \\
&\geq (\boldsymbol{1}_{n-1}^\top \boldsymbol{\Delta})^2 \\
&= (\sum_{i=1}^{n-1} \boldsymbol{\Delta}(i))^2
\end{aligned}$$

□

Lower Bounding λ_2 of a Path Graph. We will now use Lemma 4.4.4 to prove a lower bound on $\lambda_2(\boldsymbol{L}_{P_n})$. Our strategy will be to prove that the path P_n is at least some multiple of the complete graph K_n, measured by the Loewner order, i.e. $K_n \preceq f(n)P_n$ for some function $f : \mathbb{N} \to \mathbb{R}$. We can combine this with our observation earlier that $\lambda_2(\boldsymbol{L}_{K_n}) = n$ to show that

$$f(n)\lambda_2(\boldsymbol{L}_{P_n}) \geq \lambda_2(\boldsymbol{L}_{K_n}) = n, \tag{4.6}$$

and this will give our lower bound on $\lambda_2(\boldsymbol{L}_{P_n})$. When establishing the inequality between P_n and K_n, we can treat each edge of the complete graph separately, by first noting that

$$\boldsymbol{L}_{K_n} = \sum_{i<j} \boldsymbol{L}_{G_{i,j}}$$

For every edge (i,j) in the complete graph, we apply the Path Inequality, Lemma 4.4.4:

$$\begin{aligned}
G_{i,j} &\preceq (j-i) \sum_{k=i}^{j-1} G_{k,k+1} \\
&\preceq (j-i) P_n
\end{aligned}$$

This inequality says that $G_{i,j}$ is at most $(j-i)$ times the part of the path connecting i to j, and that this part of the path is less than the whole.

Summing inequality (4.3) over all edges $(i,j) \in K_n$ gives

$$K_n = \sum_{i<j} G_{i,j} \preceq \sum_{i<j} (j-i) P_n.$$

To finish the proof, we compute

$$\sum_{i<j} (j-i) \leq \sum_{i<j} n \leq n^3$$

So

$$L_{K_n} \preceq n^3 \cdot L_{P_n}.$$

Plugging this into Equation (4.6) we obtain

$$\frac{1}{n^2} \leq \lambda_2(P_n).$$

This only differs from our test vector-based upper bound in Equation (4.3) by a factor 12.

We could make this considerably tighter by being more careful about the sums.

4.5 The Complete Binary Tree

To consolidate our new expertise, let us finally consider the complete binary tree T_n with unit weight edges on n vertices. Note that in order to have T_n be a complete binary tree, we require that n is such that there is an integer d for which $n = 2^{d+1} - 1$.

T_n is the balanced binary tree on this many vertices, i.e. it consists of a root node, which has two children, each of those children have two children and so on until we reach a depth of d from the root, at which point the child vertices have no more children. A simple induction shows that indeed $n = 2^{d+1} - 1$.

We can also describe the edge set by saying that each node i has edges to its children $2i$ and $2i+1$ whenever the node labels do not exceed n. We emphasize that we still think of the graph as undirected.

4.5.1 Deriving Bounds on $\lambda_n(T_n)$

We'll start by above bounding $\lambda_n(\boldsymbol{L}_{T_n})$ using a test vector.

We let $\boldsymbol{x}(i) = 0$ for all nodes that have a child node, and $\boldsymbol{x}(i) = -1$ for even-numbered leaf nodes and $\boldsymbol{x}(i) = +1$ for odd-numbered leaf nodes. Note that there are $(n+1)/2$ leaf nodes, and every leaf node has a single edge, connecting it to a parent with value 0. Thus

$$\lambda_n(\boldsymbol{L}) = \max_{\boldsymbol{v} \neq \boldsymbol{0}} \frac{\boldsymbol{v}^\top \boldsymbol{L} \boldsymbol{v}}{\boldsymbol{v}^\top \boldsymbol{v}} \geq \frac{\boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{(n+1)/2}{(n+1)/2} = 1. \tag{4.7}$$

Meanwhile, every vertex has degree at most 3, so by Claim 4.4.3, $\lambda_n(\boldsymbol{L}) \leq 6$. So we can bound the largest eigenvalue above and below by a constant.

4.5.2 Upper Bounding λ_2 via Test Vectors

Let us give an upper bound on λ_2 of the tree using a test vector. Let $\boldsymbol{x} \in \mathbb{R}^v$ have $\boldsymbol{x}(1) = 0$ and $\boldsymbol{x}(i) = -1$ for i in the left subtree and $\boldsymbol{x}(i) = +1$ in the right subtree. Then

$$\lambda_2(\boldsymbol{L}_{T_n}) = \min_{\substack{\boldsymbol{v} \neq \boldsymbol{0} \\ \boldsymbol{v}^\top \boldsymbol{1} = 0}} \frac{\boldsymbol{v}^\top \boldsymbol{L} \boldsymbol{v}}{\boldsymbol{v}^\top \boldsymbol{v}} \leq \frac{\boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{2}{n-1}.$$

So, we have shown $\frac{1}{2n \log_2(n)} \leq \lambda_2(\boldsymbol{L}_{T_n}) \leq \frac{2}{n-1}$, and unlike the previous examples, the gap is more than a constant.

4.5.3 Lower Bounding λ_2

The following lemma gives a simple lower bound on λ_2 for any graph.

Lemma 4.5.1. *For any unit weight graph G with diameter D,*

$$\lambda_2(\boldsymbol{L}_G) \geq \frac{1}{nD}.$$

Proof. We will again prove a lower bound comparing G to the complete graph. For each edge $(i,j) \in K_n$, let $G^{i,j}$ denote a shortest path in G from i to j. This path will have length at most D. So, we have

$$\begin{aligned} K_n &= \sum_{i<j} G_{i,j} \\ &\preceq \sum_{i<j} D G^{i,j} \\ &\preceq \sum_{i<j} D G \\ &\preceq n^2 D G. \end{aligned}$$

So, we obtain the bound

$$n^2 D \lambda_2(G) \geq n,$$

which implies our desired statement. □

Since the complete binary tree T_n has diameter $2d \leq 2\log_2(n)$, by Lemma 4.5.1, $\lambda_2(\boldsymbol{L}_{T_n}) \geq \frac{1}{2n\log_2(n)}$.

In the exercises for Week 4, I will ask you to improve the lower bound to $1/(cn)$ for some constant c.

Chapter 5

Pseudo-inverses and Effective Resistance

5.1 What is a (Moore-Penrose) Pseudoinverse?

Recall that for a connected graph G with Laplacian $\boldsymbol{L}$, we have $\ker(\boldsymbol{L}) = \operatorname{span}\{\boldsymbol{1}\}$, which means $\boldsymbol{L}$ is not invertible. However, we still want some matrix which behaves like a real inverse. To be more specific, given a Laplacian $\boldsymbol{L} \in \mathbb{R}^{V \times V}$, we want some matrix $\boldsymbol{L}^+ \in \mathbb{R}^{V \times V}$ s.t.

1) $(\boldsymbol{L}^+)^\top = \boldsymbol{L}^+$ (symmetric)
2) $\boldsymbol{L}^+\boldsymbol{1} = \boldsymbol{0}$, or more generally, $\boldsymbol{L}^+\boldsymbol{v} = \boldsymbol{0}$ for $\boldsymbol{v} \in \ker(\boldsymbol{L})$
3) $\boldsymbol{L}^+\boldsymbol{L}\boldsymbol{v} = \boldsymbol{L}\boldsymbol{L}^+\boldsymbol{v} = \boldsymbol{v}$ for $\boldsymbol{v} \perp \boldsymbol{1}$, or more generally, for $\boldsymbol{v} \in \ker(\boldsymbol{L})^\perp$

Under the above conditions, $\boldsymbol{L}^+$ is uniquely defined and we call it the pseudoinverse of $\boldsymbol{L}$. Note that there are many other equivalent definitions of the pseudoinverse of some matrix $\boldsymbol{A}$, and we can also generalize the concept to matrices that aren't symmetric or even square.

Let $\lambda_i, \boldsymbol{v}_i$ be the i-th pair of eigenvalue and eigenvector of $\boldsymbol{L}$, with $\{\boldsymbol{v}_i\}_{i=1}^n$ forming a orthogonal basis. Then by the spectral theorem,

$$\boldsymbol{L} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top = \sum_i \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^\top,$$

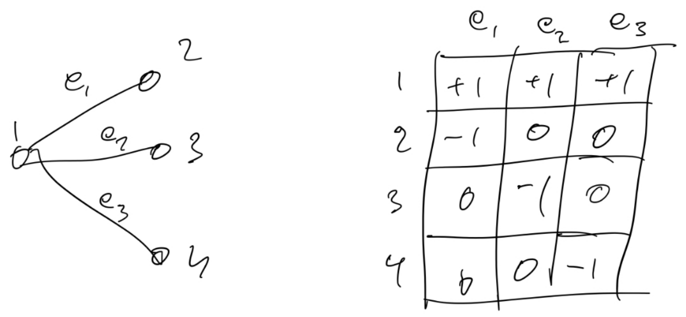
where $\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_n \end{bmatrix}$ and $\boldsymbol{\Lambda} = \operatorname{diag}\{\lambda_1, ..., \lambda_n\}$. And we can show that its pseudoinverse is exactly

$$\boldsymbol{L}^+ = \sum_{i, \lambda_i \neq 0} \lambda_i^{-1} \boldsymbol{v}_i \boldsymbol{v}_i^\top.$$

Checking conditions 1), 2), 3) is immediate. We can also prove uniqueness, but this takes slightly more work.

5.2 Electrical Flows Again

Recall the incidence matrix $\boldsymbol{B} \in \mathbb{R}^{V \times E}$ of a graph $G = (V, E)$.



1
2
3
4
e_1
e_2
e_3
$$\begin{array}{c|ccc} & e_1 & e_2 & e_3 \\ \hline 1 & +1 & +1 & +1 \\ 2 & -1 & 0 & 0 \\ 3 & 0 & -1 & 0 \\ 4 & 0 & 0 & -1 \end{array}$$

Figure 5.1: An example of a graph and its incidence matrix B.

In Chapter 1, we introduced the electrical flow routing demand $\boldsymbol{d} \in \mathbb{R}^V$. Let's call the electrical flow $\tilde{\boldsymbol{f}} \in \mathbb{R}^E$. The net flow constraint requires $\boldsymbol{B}\tilde{\boldsymbol{f}} = \boldsymbol{d}$. By Ohm's Law, $\tilde{\boldsymbol{f}} = \boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{x}$ for some voltage $x \in \mathbb{R}^V$ where $\boldsymbol{R} = \operatorname{diag}(\boldsymbol{r})$ and $\boldsymbol{r}(e) =$ resistance of edge e. We showed (in the exercises) that when $\boldsymbol{d} \perp \boldsymbol{1}$, there exists an voltage $\tilde{\boldsymbol{x}} \perp \boldsymbol{1}$ s.t. $\tilde{\boldsymbol{f}} = \boldsymbol{R}^{-1}\boldsymbol{B}^\top \tilde{\boldsymbol{x}}$ and $\boldsymbol{B}\tilde{\boldsymbol{f}} = \boldsymbol{d}$. This $\tilde{\boldsymbol{x}}$ solves $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{d}$ where $\boldsymbol{L} = \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top$.

And we also made the following claim.

Claim 5.2.1.

$$\tilde{\boldsymbol{f}} = \arg\min_{\boldsymbol{B}\boldsymbol{f}=\boldsymbol{d}} \boldsymbol{f}^\top \boldsymbol{R}\boldsymbol{f} \;\; \textit{where} \;\; \boldsymbol{f}^\top \boldsymbol{R}\boldsymbol{f} = \sum_e \boldsymbol{r}(e)\boldsymbol{f}(e)^2, \tag{5.1}$$

You proved this in the exercises for Week 1. Let's recap the proof briefly, just to get back into thinking about electrical flows.

Proof. Consider any $\boldsymbol{f} \in \mathbb{R}^E$ s.t. $\boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}$. For any $\boldsymbol{x} \in \mathbb{R}^V$, we have

$$\begin{aligned}
\frac{1}{2}\boldsymbol{f}^\top \boldsymbol{R}\boldsymbol{f} &= \frac{1}{2}\boldsymbol{f}^\top \boldsymbol{R}\boldsymbol{f} - \boldsymbol{x}^\top(\underbrace{\boldsymbol{B}\boldsymbol{f} - \boldsymbol{d}}_{\boldsymbol{0}}) \\
&\geq \min_{\boldsymbol{f} \in \mathbb{R}^E} \underbrace{\frac{1}{2}\boldsymbol{f}^\top \boldsymbol{R}\boldsymbol{f} - \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{f} + \boldsymbol{d}^\top \boldsymbol{x}}_{g(\boldsymbol{f})} \\
&= \boldsymbol{d}^\top \boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}
\end{aligned}$$

since $\nabla_{\boldsymbol{f}} g(\boldsymbol{f}) = \boldsymbol{0}$ gives us $\boldsymbol{f} = \boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{x}$. Thus, for all $\boldsymbol{f} \in \mathbb{R}^E$ s.t. $\boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}$ and all $\boldsymbol{x} \in \mathbb{R}^V$,

$$\frac{1}{2}\boldsymbol{f}^\top \boldsymbol{R}\boldsymbol{f} \geq \boldsymbol{d}^\top \boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}. \tag{5.2}$$

But for the electrical flow $\tilde{\boldsymbol{f}}$ and electrical voltage $\tilde{\boldsymbol{x}}$, we have $\tilde{\boldsymbol{f}} = \boldsymbol{R}^{-1}\boldsymbol{B}^\top\tilde{\boldsymbol{x}}$ and $\boldsymbol{L}\tilde{\boldsymbol{x}} = \boldsymbol{d}$. So

$$\tilde{\boldsymbol{f}}^\top \boldsymbol{R}\tilde{\boldsymbol{f}} = \left(\boldsymbol{R}^{-1}\boldsymbol{B}^\top\tilde{\boldsymbol{x}}\right)^\top \boldsymbol{R}\left(\boldsymbol{R}^{-1}\boldsymbol{B}^\top\tilde{\boldsymbol{x}}\right) = \tilde{\boldsymbol{x}}^\top\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}^\top\boldsymbol{L}\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}^\top\boldsymbol{d}.$$

Therefore,

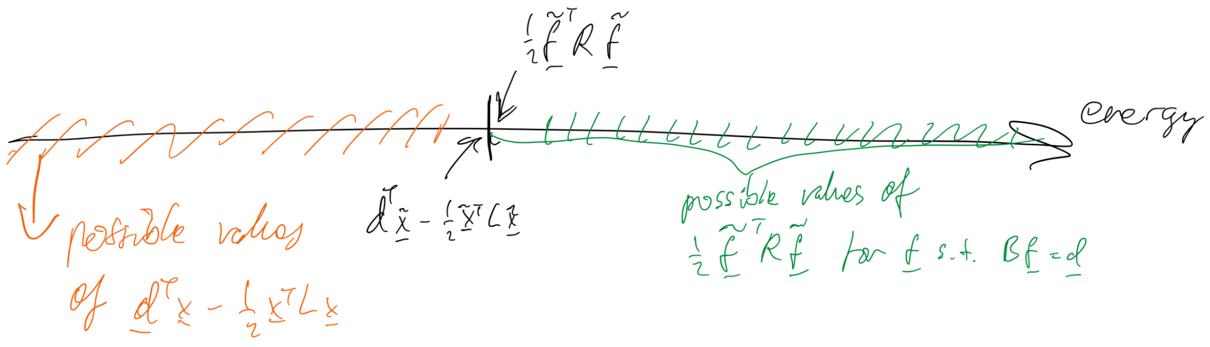
$$\frac{1}{2}\tilde{\boldsymbol{f}}^\top\boldsymbol{R}\tilde{\boldsymbol{f}} = \boldsymbol{d}^\top\tilde{\boldsymbol{x}} - \frac{1}{2}\tilde{\boldsymbol{x}}^\top\boldsymbol{L}\tilde{\boldsymbol{x}}. \tag{5.3}$$

By combining Equation (5.2) and Equation (5.3), we see that for all $\boldsymbol{f}$ s.t. $\boldsymbol{B}\boldsymbol{f} = \boldsymbol{d}$,

$$\frac{1}{2}\boldsymbol{f}^\top\boldsymbol{R}\boldsymbol{f} \geq \boldsymbol{d}^\top\tilde{\boldsymbol{x}} - \frac{1}{2}\tilde{\boldsymbol{x}}^\top\boldsymbol{L}\tilde{\boldsymbol{x}} = \frac{1}{2}\tilde{\boldsymbol{f}}^\top\boldsymbol{R}\tilde{\boldsymbol{f}}.$$

Thus $\tilde{\boldsymbol{f}}$ is the minimum electrical energy flow among all flows that route demand $\boldsymbol{d}$, proving Equation (5.1) holds.

The drawing below shows how the quantities line up:



$\frac{1}{2}\tilde{f}^\top R\tilde{f}$
energy
$d^\top\tilde{x} - \frac{1}{2}\tilde{x}^\top L\tilde{x}$
possible values of $d^\top x - \frac{1}{2}x^\top L x$
possible values of $\frac{1}{2}\tilde{f}^\top R\tilde{f}$ for f s.t. $Bf = d$

□

5.3 Effective Resistance

Given a graph $G = (V, E)$, for any pair of vertices $(a, b) \in V$, we want to compute the cost (or energy) of routing 1 unit of current from a to b. We call such cost the effective resistance between a and b, denoted by $R_{\text{eff}}(a, b)$. Recall for a single resistor $r(a, b)$,

$$\text{energy} = r(a, b)f^2(a, b) = r(a, b).$$

So when we have a graph consisting of just one edge (a, b), the effective resistance is just $R_{\text{eff}}(a, b) = r(a, b)$.

In a general graph, we can also consider the energy required to route one unit of current between two vertices. For any pair $a, b \in V$, we have

$$R_{\text{eff}}(a, b) = \min_{\boldsymbol{B}\boldsymbol{f} = \boldsymbol{e}_b - \boldsymbol{e}_a} \boldsymbol{f}^\top\boldsymbol{R}\boldsymbol{f},$$

where $\boldsymbol{e}_v \in \mathbb{R}^V$ is the indicator vector of v. Note that the cost of routing F units of flow from a to b will be $R_{\text{eff}}(a,b) \cdot F^2$.

Since $(\boldsymbol{e}_b - \boldsymbol{e}_a)^\top \boldsymbol{1} = 0$, we know from the previous section that $R_{\text{eff}}(a,b) = \tilde{\boldsymbol{f}}^\top \boldsymbol{R}\tilde{\boldsymbol{f}}$ where $\tilde{\boldsymbol{f}}$ is the electrical flow. Now we can write $\boldsymbol{L}\tilde{\boldsymbol{x}} = \boldsymbol{e}_b - \boldsymbol{e}_a$ and $\tilde{\boldsymbol{x}} = \boldsymbol{L}^+(\boldsymbol{e}_b - \boldsymbol{e}_a)$ for the electrical voltages routing 1 unit of current from a to b. Now the energy of routing 1 unit of current from a to b is

$$R_{\text{eff}}(a,b) = \tilde{\boldsymbol{f}}^\top \boldsymbol{R}\tilde{\boldsymbol{f}} = \tilde{\boldsymbol{x}}^\top \boldsymbol{L}\tilde{\boldsymbol{x}} = (\boldsymbol{e}_b - \boldsymbol{e}_a)^\top \boldsymbol{L}^+\boldsymbol{L}\boldsymbol{L}^+(\boldsymbol{e}_b - \boldsymbol{e}_a) = (\boldsymbol{e}_b - \boldsymbol{e}_a)^\top \boldsymbol{L}^+(\boldsymbol{e}_b - \boldsymbol{e}_a),$$

where the last equality is due to $\boldsymbol{L}^+\boldsymbol{L}\boldsymbol{L}^+ = \boldsymbol{L}^+$.

Remark 5.3.1. We have now seen several different expressions that all take on the same value: the energy of the electrical flow. It's useful to remind yourself what these are. Consider an electrical flow $\tilde{\boldsymbol{f}}$ routes demand $\boldsymbol{d}$, and associated electrical voltages $\tilde{\boldsymbol{x}}$. We know that $\boldsymbol{B}\tilde{\boldsymbol{f}} = \boldsymbol{d}$, and $\boldsymbol{f} = \boldsymbol{R}^{-1}\boldsymbol{B}^\top\tilde{\boldsymbol{x}}$, and $\boldsymbol{L}\tilde{\boldsymbol{x}} = \boldsymbol{d}$, where $\boldsymbol{L} = \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top$. And we have seen how to express the electrical energy using many different quantities:

$$\tilde{\boldsymbol{f}}^\top \boldsymbol{R}\tilde{\boldsymbol{f}} = \tilde{\boldsymbol{x}}^\top \boldsymbol{L}\tilde{\boldsymbol{x}} = \boldsymbol{d}^\top \boldsymbol{L}^+\boldsymbol{d} = \boldsymbol{d}^\top\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{f}}^\top \boldsymbol{B}^\top\tilde{\boldsymbol{x}}$$

Claim 5.3.2. *Any PSD matrix* $\boldsymbol{A}$ *has a PSD square root* $\boldsymbol{A}^{1/2}$ *s.t.* $\boldsymbol{A}^{1/2}\boldsymbol{A}^{1/2} = \boldsymbol{A}$.

Proof. By the spectral theorem, $\boldsymbol{A} = \sum_i \lambda_i \boldsymbol{v}_i\boldsymbol{v}_i^\top$ where $\{\boldsymbol{v}_i\}$ are orthonormal. Let $\boldsymbol{A}^{1/2} = \sum_i \lambda_i^{1/2}\boldsymbol{v}_i\boldsymbol{v}_i^\top$. Then

$$\begin{aligned}\boldsymbol{A}^{1/2}\boldsymbol{A}^{1/2} &= \left(\sum_i \lambda_i^{1/2}\boldsymbol{v}_i\boldsymbol{v}_i^\top\right)^2 \\ &= \sum_i \lambda_i \boldsymbol{v}_i\boldsymbol{v}_i^\top\boldsymbol{v}_i\boldsymbol{v}_i^\top + \sum_{i\neq j} \lambda_i \boldsymbol{v}_i\boldsymbol{v}_i^\top\boldsymbol{v}_j\boldsymbol{v}_j^\top \\ &= \sum_i \lambda_i \boldsymbol{v}_i\boldsymbol{v}_i^\top\end{aligned}$$

where the last equality is due to $\boldsymbol{v}_i^\top\boldsymbol{v}_j = \delta_{ij}$. It's easy to see that $\boldsymbol{A}^{1/2}$ is also PSD. □

Let $\boldsymbol{L}^{+/2}$ be the square root of $\boldsymbol{L}^+$. So

$$R_{\text{eff}}(a,b) = (\boldsymbol{e}_b - \boldsymbol{e}_a)^\top \boldsymbol{L}^+(\boldsymbol{e}_b - \boldsymbol{e}_a) = \|\boldsymbol{L}^{+/2}(\boldsymbol{e}_b - \boldsymbol{e}_a)\|^2.$$

Example: Effective resistance in a path. Consider a path graph on vertices $V = \{1, 2, 3, \ldots, k+1\}$, with with resistances $\boldsymbol{r}(1), \boldsymbol{r}(2), \ldots, \boldsymbol{r}(k)$ on the edges of the path.

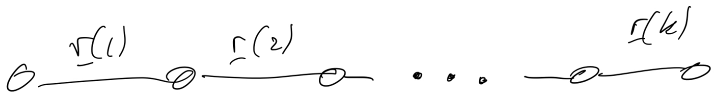


Figure 5.2: A path graph with k edges.

The effective resistance between the endpoints is

$$R_{\text{eff}}(1, k+1) = \sum_{i=1}^{k} \boldsymbol{r}(i)$$

To see this, observe that to have 1 unit of flow going from vertex 1 to vertex $k+1$, we must have one unit flowing across each edge i. Let $\boldsymbol{\Delta}(i)$ be the voltage difference across edge i, and $\boldsymbol{f}(i)$ the flow on the edge. Then $1 = \boldsymbol{f}(i) = \frac{\boldsymbol{\Delta}(i)}{\boldsymbol{r}(i)}$, so that $\boldsymbol{\Delta}(i) = \boldsymbol{r}(i)$. The electrical voltages are then $\tilde{\boldsymbol{x}} \in \mathbb{R}^V$ where $\tilde{\boldsymbol{x}}(i) = \tilde{\boldsymbol{x}}(1) + \sum_{j<i} \boldsymbol{\Delta}(j)$. Hence the effective resistance is

$$R_{\text{eff}}(1, k+1) = \boldsymbol{d}^\top \tilde{\boldsymbol{x}} = (\boldsymbol{e}_{k+1} - \boldsymbol{e}_1)^\top \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}(k+1) - \tilde{\boldsymbol{x}}(1) = \sum_{i=1}^{k} \boldsymbol{r}(i).$$

This behavior is sometimes known as the fact that the resistance of resistors adds up when they are connected in series.

Example: Effective resistance of parallel edges. So far, we have only considered graphs with at most one edge between any two vertices. But that math also works if we allow a pair of vertices to have multiple distinct edges connecting them. We refer to this as *multi-edges*. Suppose we have a graph on just two vertices, $V = \{1, 2\}$, and these are connected by k parallel multi-edges with resistances $\boldsymbol{r}(1), \boldsymbol{r}(2), \ldots, \boldsymbol{r}(k)$.

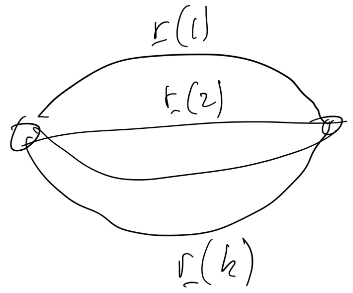


Figure 5.3: A graph on just two vertices with k parallel multiedges.

The effective resistance between the endpoints is

$$R_{\text{eff}}(1, 2) = \frac{1}{\sum_{i=1}^{k} 1/\boldsymbol{r}(i)}.$$

Let's see why. Our electrical voltages $\tilde{\boldsymbol{x}} \in \mathbb{R}^V$ can be described by just the voltage difference $\Delta \in \mathbb{R}$ between vertex 1 and vertex 2, i.e. $\tilde{\boldsymbol{x}}(2) - \tilde{\boldsymbol{x}}(1) = \Delta$. which creates a flow on edge i of $\tilde{\boldsymbol{f}}(i) = \Delta/\boldsymbol{r}(i)$. Thus the total flow from vertex 1 to vertex 2 is $1 = \sum_i \Delta/\boldsymbol{r}(i)$, so that $\Delta = \frac{1}{\sum_{i=1}^k 1/\boldsymbol{r}(i)}$. Meanwhile, the effective resistance is also

$$R_{\text{eff}}(1,2) = (\boldsymbol{e}_2 - \boldsymbol{e}_1)^\top \tilde{\boldsymbol{x}} = \Delta = \frac{1}{\sum_{i=1}^k 1/\boldsymbol{r}(i)}$$

5.3.1 Effective Resistance is a Distance

Definition 5.3.3. Consider a weighted undirected graph G with vertex set V. We say function $d : V \times V \to \mathbb{R}$, which takes a pair of vertices and returns a real number, is a *distance* if it satisfies

1. $d(a,a) = 0$ for all $a \in V$
2. $d(a,b) \geq 0$ for all $a,b \in V$.
3. $d(a,b) = d(b,a)$ for all $a,b \in V$.
4. $d(a,b) \leq d(a,c) + d(c,b)$ for all $a,b,c \in V$.

Lemma 5.3.4. *R_{eff} is a distance.*

Before proving this lemma, let's see a claim that will help us finish the proof.

Claim 5.3.5. *Let $\boldsymbol{L}\tilde{\boldsymbol{x}} = \boldsymbol{e}_b - \boldsymbol{e}_a$. Then for all $\boldsymbol{c} \in V$, we have $\tilde{\boldsymbol{x}}(b) \geq \tilde{\boldsymbol{x}}(c) \geq \tilde{\boldsymbol{x}}(a)$.*

We only sketch a proof of this claim:

Proof sketch. Consider any $c \in V$, where $c \neq a,b$. Now $(\boldsymbol{L}\tilde{\boldsymbol{x}})(c) = 0$, i.e.

$$\left(\sum_{(u,c)} \boldsymbol{w}(u,c) \right) \tilde{\boldsymbol{x}}(c) - \left(\sum_{(u,c)} \boldsymbol{w}(u,c)\tilde{\boldsymbol{x}}(u) \right) = 0$$

Rearranging $\tilde{\boldsymbol{x}}(c) = \frac{\sum_{(u,c)} \boldsymbol{w}(u,c)\tilde{\boldsymbol{x}}(u)}{\sum_{(u,c)} \boldsymbol{w}(u,c)}$. This tells us that $\tilde{\boldsymbol{x}}(c)$ is a weighted average of the voltages of its neighbors. From this, we can show that $\tilde{\boldsymbol{x}}(a)$ and $\tilde{\boldsymbol{x}}(b)$ are the extreme values. □

Proof. It is easy to check that conditions 1, 2, and 3 of Definition 5.3.3 are satisfied by R_{eff}. Let us confirm condition 4.

For any u, v, let $\tilde{\boldsymbol{x}}_{u,v} = \boldsymbol{L}^+(-\boldsymbol{e}_u + \boldsymbol{e}_v)$. Then

$$\tilde{\boldsymbol{x}}_{a,b} = \boldsymbol{L}^+(-\boldsymbol{e}_a + \boldsymbol{e}_b) = \boldsymbol{L}^+(-\boldsymbol{e}_a + \boldsymbol{e}_c - \boldsymbol{e}_c + \boldsymbol{e}_b) = \tilde{\boldsymbol{x}}_{a,c} + \tilde{\boldsymbol{x}}_{c,b}.$$

Thus,

$$\begin{aligned} R_{\text{eff}}(a,b) = (-\boldsymbol{e}_a + \boldsymbol{e}_b)^\top \tilde{\boldsymbol{x}}_{a,b} &= (-\boldsymbol{e}_a + \boldsymbol{e}_b)^\top (\tilde{\boldsymbol{x}}_{a,c} + \tilde{\boldsymbol{x}}_{c,b}) \\ &= -\tilde{\boldsymbol{x}}_{a,c}(a) + \tilde{\boldsymbol{x}}_{a,c}(b) - \tilde{\boldsymbol{x}}_{c,b}(a) + \tilde{\boldsymbol{x}}_{c,b}(b) \\ &\leq -\tilde{\boldsymbol{x}}_{a,c}(a) + \tilde{\boldsymbol{x}}_{a,c}(c) - \tilde{\boldsymbol{x}}_{c,b}(c) + \tilde{\boldsymbol{x}}_{c,b}(b). \end{aligned}$$

where in the last line we applied Claim 5.3.5 to show that $\tilde{\boldsymbol{x}}_{a,c}(b) \leq \tilde{\boldsymbol{x}}_{a,c}(c)$ and $-\tilde{\boldsymbol{x}}_{c,b}(a) \leq -\tilde{\boldsymbol{x}}_{c,b}(c)$. $\square$

Chapter 6

Different Perspectives on Gaussian Elimination

6.1 An Optimization View of Gaussian Elimination for Laplacians

In this section, we will explore how to exactly minimize a Laplacian quadratic form by minimizing over one variable at a time. It turns out that this is in fact Gaussian Elimination in disguise – or, more precisely, the variant of Gaussian elimination that we tend to use on symmetric matrices, which is called Cholesky factorization.

Consider a Laplacian $\boldsymbol{L}$ of a connected graph $G = (V, E, \boldsymbol{w})$, where $\boldsymbol{w} \in \mathbb{R}^E$ is a vector of positive edge weights. Let $\boldsymbol{W} \in \mathbb{R}^{E\times E}$ be the diagonal matrix with the edge weights on the diagonal, i.e. $\boldsymbol{W} = \operatorname{diag}(\boldsymbol{w})$ and $\boldsymbol{L} = \boldsymbol{B}\boldsymbol{W}\boldsymbol{B}^\top$. Let $\boldsymbol{d} \in \mathbb{R}^V$ be a demand vector s.t. $\boldsymbol{d} \perp \boldsymbol{1}$.

Let us define an energy

$$\mathcal{E}(\boldsymbol{x}) = -\boldsymbol{d}^\top \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}$$

Note that this function is convex and is minimized at $\boldsymbol{x}$ s.t. $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{d}$.

We will now explore an approach to solving the minimization problem

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

Let $\boldsymbol{x} = \begin{pmatrix} y \\ \boldsymbol{z} \end{pmatrix}$ where $y \in \mathbb{R}$ and $\boldsymbol{z} \in \mathbb{R}^{V\setminus\{1\}}$.

We will now explore how to minimize over y, given any $\boldsymbol{z}$. Once we find an expression for y in terms of $\boldsymbol{z}$, we will be able to reduce it to a new quadratic minimization problem in $\boldsymbol{z}$,

$$\mathcal{E}'(\boldsymbol{z}) = -\boldsymbol{d}'^\top \boldsymbol{z} + \frac{1}{2}\boldsymbol{z}^\top \boldsymbol{L}'\boldsymbol{z}$$

where $\boldsymbol{d}'$ is a demand vector on the remaining vertices, with $\boldsymbol{d} \perp \boldsymbol{1}$ and $\boldsymbol{L}'$ is a Laplacian of a graph on the remaining vertices $V' = V \setminus \{1\}$. We can then repeat the procedure to eliminate another variable and so on. Eventually, we can then find all the solution to our original minimization problem.

To help us understand how to minimize over the first variable, we introduce some notation for the first row and column of the Laplacian:

$$\boldsymbol{L} = \begin{pmatrix} W & -\boldsymbol{a}^\top \\ -\boldsymbol{a} & \operatorname{diag}(\boldsymbol{a}) + \boldsymbol{L}_{-1} \end{pmatrix}. \tag{6.1}$$

Note that W is the weighted degree of vertex 1, and that

$$\begin{pmatrix} W & -\boldsymbol{a}^\top \\ -\boldsymbol{a} & \operatorname{diag}(\boldsymbol{a}) \end{pmatrix} \tag{6.2}$$

is the Laplacian of the subgraph of G containing only the edges incident on vertex 1, while $\boldsymbol{L}_{-1}$ is the Laplacian of the subgraph of G containing all edges *not* incident on vertex 1.

Let us also write $\boldsymbol{d} = \begin{pmatrix} b \\ \boldsymbol{c} \end{pmatrix}$ where $b \in \mathbb{R}$ and $\boldsymbol{c} \in \mathbb{R}^{V \setminus \{1\}}$.

Now,

$$\begin{aligned} \mathcal{E}(\boldsymbol{x}) = -\boldsymbol{d}^\top \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} &= -\begin{pmatrix} b \\ \boldsymbol{c} \end{pmatrix}^\top \begin{pmatrix} y \\ \boldsymbol{z} \end{pmatrix} + \frac{1}{2}\begin{pmatrix} y \\ \boldsymbol{z} \end{pmatrix}^\top \begin{pmatrix} W & -\boldsymbol{a}^\top \\ -\boldsymbol{a} & \operatorname{diag}(\boldsymbol{a}) + \boldsymbol{L}_{-1} \end{pmatrix} \begin{pmatrix} y \\ \boldsymbol{z} \end{pmatrix} \\ &= -by - \boldsymbol{c}^\top \boldsymbol{z} + \frac{1}{2}\left(y^2 W - 2y\boldsymbol{a}^\top \boldsymbol{z} + \boldsymbol{z}^\top \operatorname{diag}(\boldsymbol{a})\boldsymbol{z} + \boldsymbol{z}^\top \boldsymbol{L}_{-1}\boldsymbol{z}\right). \end{aligned}$$

Now, to minimize over y, we set $\frac{\partial}{\partial y}\mathcal{E}(\boldsymbol{x}) = 0$ and get

$$-b + yW - \boldsymbol{a}^\top \boldsymbol{z} = 0.$$

Solving for y, we get that the minimizing y is

$$y = \frac{1}{W}(b + \boldsymbol{a}^\top \boldsymbol{z}). \tag{6.3}$$

Observe that

$$\begin{aligned} \mathcal{E}(\boldsymbol{x}) &= -by - \boldsymbol{c}^\top \boldsymbol{z} + \frac{1}{2}\left(y^2 W - 2y\boldsymbol{a}^\top \boldsymbol{z} + \boldsymbol{z}^\top \operatorname{diag}(\boldsymbol{a})\boldsymbol{z} + \boldsymbol{z}^\top \boldsymbol{L}_{-1}\boldsymbol{z}\right) \\ &= -by - \boldsymbol{c}^\top \boldsymbol{z} + \frac{1}{2}\left(\frac{1}{W}(yW - \boldsymbol{a}^\top \boldsymbol{z})^2 \underbrace{- \frac{1}{W}\boldsymbol{z}^\top \boldsymbol{a}\boldsymbol{a}^\top \boldsymbol{z} + \boldsymbol{z}^\top \operatorname{diag}(\boldsymbol{a})\boldsymbol{z} + \boldsymbol{z}^\top \boldsymbol{L}_{-1}\boldsymbol{z}}_{\text{Let } \boldsymbol{S} = \operatorname{diag}(\boldsymbol{a}) - \frac{1}{W}\boldsymbol{a}\boldsymbol{a}^\top + \boldsymbol{L}_{-1}}\right) \\ &= -by - \boldsymbol{c}^\top \boldsymbol{z} + \frac{1}{2}\left(\frac{1}{W}(yW - \boldsymbol{a}^\top \boldsymbol{z})^2 + \boldsymbol{z}^\top \boldsymbol{S}\boldsymbol{z}\right), \end{aligned}$$

where we simplified the expression by defining $\boldsymbol{S} = \operatorname{diag}(\boldsymbol{a}) - \frac{1}{W}\boldsymbol{a}\boldsymbol{a}^\top + \boldsymbol{L}_{-1}$. Plugging in $y = \frac{1}{W}(b + \boldsymbol{a}^\top \boldsymbol{z})$, we get

$$\min_y \mathcal{E}\begin{pmatrix} y \\ \boldsymbol{z} \end{pmatrix} = -\left(\boldsymbol{c} + b\frac{1}{W}\boldsymbol{a}\right)^\top \boldsymbol{z} - \frac{b^2}{2W} + \frac{1}{2}\boldsymbol{z}^\top \boldsymbol{S}\boldsymbol{z}.$$

Now, we define $\boldsymbol{d}' = \boldsymbol{c} + b\frac{1}{W}\boldsymbol{a}$ and $\mathcal{E}'(\boldsymbol{z}) = -\boldsymbol{d}'^\top \boldsymbol{z} + \frac{1}{2}\boldsymbol{z}^\top \boldsymbol{S}\boldsymbol{z}$. And, we can see that

$$\arg\min_{\boldsymbol{z}} \min_y \mathcal{E}\begin{pmatrix} y \\ \boldsymbol{z} \end{pmatrix} = \arg\min_{\boldsymbol{z}} \mathcal{E}'(\boldsymbol{z}),$$

since dropping the constant term $-\frac{b^2}{2W}$ does not change what the minimizing $\boldsymbol{z}$ values are.

Claim 6.1.1.

1. $\boldsymbol{d}' \perp \boldsymbol{1}$
2. $\boldsymbol{S} = \operatorname{diag}(\boldsymbol{a}) - \frac{1}{W}\boldsymbol{a}\boldsymbol{a}^\top + \boldsymbol{L}_{-1}$ *is a Laplacian of a graph on the vertex set* $V \setminus \{1\}$.

We will prove Claim 6.1.1 in a moment. From the Claim, we see that the problem of finding $\arg\min_{\boldsymbol{z}} \mathcal{E}'(\boldsymbol{z})$, is exactly of the same form as finding $\arg\min_{\boldsymbol{x}} \mathcal{E}(\boldsymbol{x})$, but with one fewer variables.

We can get a minimizing $\boldsymbol{x}$ that solves $\arg\min_{\boldsymbol{x}} \mathcal{E}(\boldsymbol{x})$ by repeating the variable elimination procedure until we get down to a single variable and finding its value. We then have to work back up to getting a solution for $\boldsymbol{z}$, and then substitute that into Equation (6.3) to get the value for y.

Remark 6.1.2. In fact, this perspective on Gaussian elimination also makes sense for any positive definite matrix. In this setting, minimizing over one variable will leave us with another positive definite quadratic minimization problem.

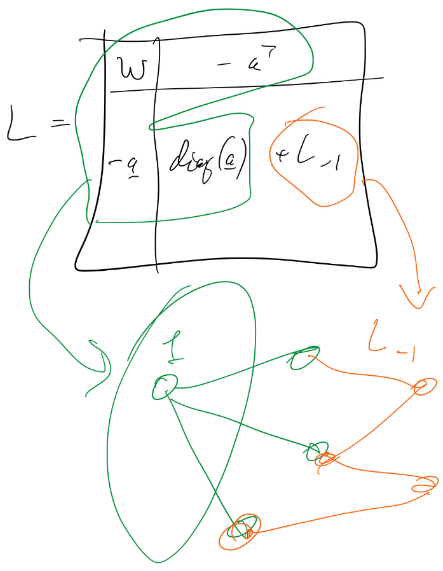
Proof of Claim 6.1.1. To establish the first part, we note that $\boldsymbol{1}^\top \boldsymbol{d}' = \boldsymbol{1}^\top \boldsymbol{c} + b\frac{\boldsymbol{1}^\top \boldsymbol{a}}{W} = \boldsymbol{1}^\top \boldsymbol{c} + b = \boldsymbol{1}^\top \boldsymbol{d} = 0$. To establish the second part, we notice that $\boldsymbol{L}_{-1}$ is a graph Laplacian by definition. Since the sum of two graph Laplacians is another graph Laplacian, it now suffices to show that $\boldsymbol{S}$ is a graph Laplacian.

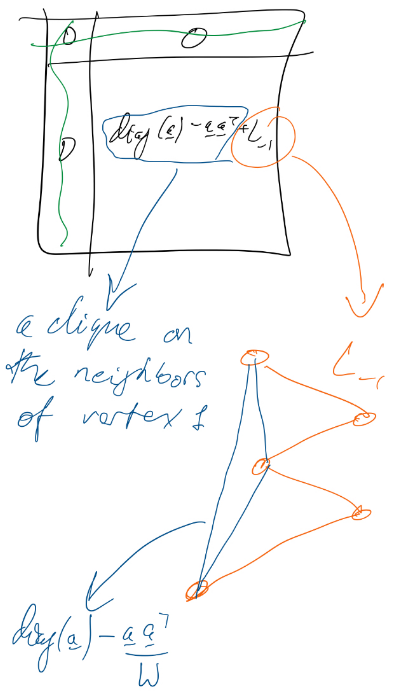
Claim 6.1.3. *A matrix* $\boldsymbol{M}$ *is a graph Laplacian if and only it satisfies the following conditions:*

- $\boldsymbol{M}^\top = \boldsymbol{M}$.
- *The diagonal entries of* $\boldsymbol{M}$ *are non-negative, and the off-diagonal entries of* $\boldsymbol{M}$ *are non-positive.*
- $\boldsymbol{M}\boldsymbol{1} = \boldsymbol{0}$.

Let's see that Claim 6.1.3 is true. Firstly, when the conditions hold we can write $\boldsymbol{M} = \boldsymbol{D} - \boldsymbol{A}$ where $\boldsymbol{D}$ is diagonal and non-negative, and $\boldsymbol{A}$ is non-negative, symmetric, and zero on the diagonal, and from the last condition $\boldsymbol{D}(i,i) = \sum_{j \neq i} \boldsymbol{A}(i,j)$. Thus we can view $\boldsymbol{A}$ as a graph adjacency matrix and $\boldsymbol{D}$ as the corresponding diagonal matrix of weighted degrees. Secondly, it is easy to check that the conditions hold for any graph Laplacian, so the conditions indeed hold if and only if. Now we have to check that the claim applies to $\boldsymbol{S}$. We leave this as an exercise for the reader.

Finally, we want to argue that the graph corresponding to $\boldsymbol{S}$ is connected. Consider any $i, j \in V \setminus \{1\}$. Since G, the graph of $\boldsymbol{L}$, is connected, there exists a simple path in G connecting i and j. If this path does not use vertex 1, it is a path in the graph of $\boldsymbol{L}_{-1}$ and hence in the graph of $\boldsymbol{S}$. If the path does use vertex 1, it must do so by reaching the vertex on some edge $(v, 1)$ and leaving on a different edge $(1, u)$. Replace this pair of edges with edge (u, v), which appears in the graph of $\boldsymbol{S}$ because $\boldsymbol{S}(u, v) < 0$. Now we have a path in the graph of $\boldsymbol{S}$. □





0
0
$\deg(a) - aa^T + L_{-1}$
a clique on the neighbors of vertex 1
L_{-1}
$\deg(a) - \frac{aa^T}{W}$

6.2 An Additive View of Gaussian Elimination

Cholesky decomposition basics. Again we consider a graph Laplacian $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ of a connected graph $G = (V, E, \boldsymbol{w})$, where as usual $|V| = n$ and $|E| = m$.

In this Section, we'll study how to decompose a graph Laplacian as $\boldsymbol{L} = \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top$, where $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{n\times n}$ is a lower triangular matrix, i.e. $\boldsymbol{\mathcal{L}}(i,j) = 0$ for $i < j$. Such a factorization is called a Cholesky decomposition. It is essentially the result of Gaussian elimination with a slight twist to ensure the matrices maintained at intermediate steps of the algorithm remain symmetric.

We use $\mathrm{nnz}(\boldsymbol{A})$ to denote the number of non-zero entries of matrix $\boldsymbol{A}$.

Lemma 6.2.1. *Given an* invertible *square lower triangular matrix $\boldsymbol{\mathcal{L}}$, we can solve the linear equation $\boldsymbol{\mathcal{L}}\boldsymbol{y} = \boldsymbol{b}$ in time $O(\mathrm{nnz}(\boldsymbol{\mathcal{L}}))$. Similarly, given an upper triangular matrix $\boldsymbol{\mathcal{U}}$, we can solve linear equations $\boldsymbol{\mathcal{U}}\boldsymbol{z} = \boldsymbol{c}$ in time $O(\mathrm{nnz}(\boldsymbol{\mathcal{U}}))$.*

We omit the proof, which is a straight-forward exercise. The algorithms for solving linear equations in upper and lower triangular matrices are known as forward and back substitution respectively.

Remark 6.2.2. Strictly speaking, the lemma requires us to have access an adjacency list representation of $\boldsymbol{\mathcal{L}}$ so that we can quickly tell where the non-zero entries are.

Using forward and back substitution, if we have a decomposition of an invertible matrix $\boldsymbol{M}$ into $\boldsymbol{M} = \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top$, we can now solve linear equations in $\boldsymbol{M}$ in time $O(\mathrm{nnz}(\boldsymbol{\mathcal{L}}))$.

Remark 6.2.3. We have learned about decompositions using a lower triangular matrix, and later we will see an algorithm for computing these. In fact, we can have more flexibility than that. From an algorithmic perspective, it is sufficient that there exists a permutation matrix $\boldsymbol{P}$ s.t. $\boldsymbol{P}\boldsymbol{\mathcal{L}}\boldsymbol{P}^\top$ is lower triangular. If we know the ordering under which the matrix becomes lower triangular, we can perform substitution according to that order to solve linear equations in the matrix without having to explicitly apply a permutation to the matrix.

Dealing with pseudoinverses. But how can we solve a linear equation in $\boldsymbol{L} = \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top$, where $\boldsymbol{L}$ is not invertible? For graph Laplacians we have a simple characterization of the kernel, and because of this, dealing with the lack of invertibility turns out to be fairly easy.

We can use the following lemma which you will prove in an exercise next week.

Lemma 6.2.4. *Consider a real symmetric matrix $\boldsymbol{M} = \boldsymbol{X}\boldsymbol{Y}\boldsymbol{X}^\top$, where $\boldsymbol{X}$ is real and invertible and $\boldsymbol{Y}$ is real symmetric. Let $\boldsymbol{\Pi}_{\boldsymbol{M}}$ denote the orthogonal projection to the image of $\boldsymbol{M}$. Then $\boldsymbol{M}^+ = \boldsymbol{\Pi}_{\boldsymbol{M}}(\boldsymbol{X}^\top)^{-1}\boldsymbol{Y}^+\boldsymbol{X}^{-1}\boldsymbol{\Pi}_{\boldsymbol{M}}$.*

The factorizations $\boldsymbol{L} = \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top$ that we produce will have the property that all diagonal entries of $\boldsymbol{\mathcal{L}}$ are strictly non-zero, except that $\boldsymbol{\mathcal{L}}(n,n) = 0$. Let $\widehat{\boldsymbol{\mathcal{L}}}$ be the matrix whose entries agree with $\boldsymbol{\mathcal{L}}$, except that $\widehat{\boldsymbol{\mathcal{L}}}(n,n) = 1$. Let $\boldsymbol{\mathcal{D}}$ be the diagonal matrix with $\boldsymbol{\mathcal{D}}(i,i) = 1$ for $i < n$ and $\boldsymbol{\mathcal{D}}(n,n) = 0$. Then $\boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top = \widehat{\boldsymbol{\mathcal{L}}}\boldsymbol{\mathcal{D}}\widehat{\boldsymbol{\mathcal{L}}}^\top$, and $\widehat{\boldsymbol{\mathcal{L}}}$ is invertible, and $\boldsymbol{\mathcal{D}}^+ = \boldsymbol{\mathcal{D}}$. Finally, $\boldsymbol{\Pi}_{\boldsymbol{L}} = \boldsymbol{I} - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^\top$, because this matrix is acts like identity on vectors orthogonal to $\boldsymbol{1}$ and ensures $\boldsymbol{\Pi}_{\boldsymbol{L}}\boldsymbol{1} = \boldsymbol{0}$, and this matrix can be applied to a vector in $O(n)$ time. Thus $\boldsymbol{L}^+ = \boldsymbol{\Pi}_{\boldsymbol{L}}(\widehat{\boldsymbol{\mathcal{L}}}^\top)^{-1}\boldsymbol{\mathcal{D}}\widehat{\boldsymbol{\mathcal{L}}}^{-1}\boldsymbol{\Pi}_{\boldsymbol{L}}$, and this matrix can be applied in time $O(\mathrm{nnz}(\boldsymbol{\mathcal{L}}))$.

An additive view of Gaussian Elimination. The following theorem describes Gaussian Elimination / Cholesky decompostion of a graph Laplacian.

Theorem 6.2.5 (Cholesky Decomposition on graph Laplacians)**.** *Let $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ be a graph Laplacian of a connected graph $G = (V, E, \boldsymbol{w})$, where $|V| = n$. Using Gaussian Elimination, we can compute in $O(n^3)$ time a factorization $\boldsymbol{L} = \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top$ where $\boldsymbol{\mathcal{L}}$ is lower triangular, and has positive diagonal entries except $\boldsymbol{\mathcal{L}}(n, n) = 0$.*

Proof. Let $\boldsymbol{L}^{(0)} = \boldsymbol{L}$. We will use $\boldsymbol{A}(:, i)$ to denote the the ith column of a matrix $\boldsymbol{A}$. Now, for $i = 1$ to $i = n - 1$ we define

$$\boldsymbol{l}_i = \frac{1}{\sqrt{\boldsymbol{L}^{(i-1)}(i,i)}}\boldsymbol{L}^{(i-1)}(:, i) \text{ and } \boldsymbol{L}^{(i)} = \boldsymbol{L}^{(i-1)} - \boldsymbol{l}_i\boldsymbol{l}_i^\top$$

Finally, we let $\boldsymbol{l}_n = \boldsymbol{0}_{n\times 1}$. We will show later that

$$\boldsymbol{L}^{(n-1)} = \boldsymbol{0}_{n\times n}. \tag{6.4}$$

It follows that $\boldsymbol{L} = \sum_i \boldsymbol{l}_i\boldsymbol{l}_i^\top$, provided this procedure is well-defined, i.e. $\boldsymbol{L}^{(i-1)}(i, i) \neq 0$ for all $i < n$. We will sketch a proof of this later, while also establishing several other properties of the procedure.

Given a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $U \subseteq [n]$, we will use $\boldsymbol{A}(U, U)$ to denote the principal submatrix of $\boldsymbol{A}$ obtained by restricting to the rows and columns with index in U, i.e. all entries $\boldsymbol{A}(i, j)$ where $i, j \in U$.

Claim 6.2.6. *Fix some $i < n$. Let $U = \{i + 1, \ldots, n\}$. Then $\boldsymbol{L}^{(i)}(i, j) = 0$ if $i \notin U$ or $j \notin U$. And $\boldsymbol{L}^{(i)}(U, U)$ is a graph Laplacian of a connected graph on the vertex set U.*

From this claim, it follows that $\boldsymbol{L}^{(i-1)}(i, i) \neq 0$ for $i < n$, since a connected graph Laplacian on a graph with $|U| > 1$ vertices cannot have a zero on the diagonal, and it follows that $\boldsymbol{L}^{(n-1)}(i, i) = 0$, because the only graph we allow on one vertex is the empty graph. This shows Equation (6.4) holds. □

Sketch of proof of Claim 6.2.6. We will focus on the first elimination, as the remaining are similar. Adopting the same notation as in Equation (6.1), we write

$$\boldsymbol{L}^{(0)} = \boldsymbol{L} = \begin{pmatrix} W & -\boldsymbol{a}^\top \\ -\boldsymbol{a} & \operatorname{diag}(\boldsymbol{a}) + \boldsymbol{L}_{-1} \end{pmatrix}$$

and, noting that

$$\boldsymbol{l}_1\boldsymbol{l}_1^\top = \begin{pmatrix} W & -\boldsymbol{a}^\top \\ -\boldsymbol{a} & \frac{1}{W}\boldsymbol{a}\boldsymbol{a}^\top \end{pmatrix}$$

we see that

$$\boldsymbol{L}^{(1)} = \boldsymbol{L}^{(0)} - \boldsymbol{l}_1\boldsymbol{l}_1^\top = \begin{pmatrix} 0 & \boldsymbol{0} \\ \boldsymbol{0} & \operatorname{diag}(\boldsymbol{a}) - \frac{1}{W}\boldsymbol{a}\boldsymbol{a}^\top + \boldsymbol{L}_{-1} \end{pmatrix}.$$

Thus the first row and column of $\boldsymbol{L}^{(1)}$ are zero claimed. It also follows by Claim 6.1.1 that $\boldsymbol{L}^{(1)}(\{2, \ldots, n\}, \{2, \ldots, n\})$ is the Laplacian of a connected graph. This proves Claim 6.2.6 for the case $i = 1$. An induction following the same pattern can be used to prove the claim for all $i < n$. $\square$