MATH-329 – Continuous optimization

Instructor: Nicolas Boumal
TAs: Antoine Gonon, Guifré Sánchez i Serra

Institute of mathematics, EPFL

compiled December 17, 2025

Contents

Preface

Continuous optimization (also called nonlinear optimization or nonlinear programming) is a branch of applied mathematics concerned with *modeling* computational problems in the form $\min_{x \in S} f(x)$ where S is a continuous space, and with *designing*, *analyzing* and *implementing* efficient algorithms to solve such problems.

These notes contain the material covered in *MATH-329 – Continuous optimization* taught at EPFL during the Spring semesters of 2021 and 2022, and the Fall semesters of 2022–2025. They draw on the reference book of the course, namely,

- J. Nocedal and S. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer Science & Business Media, 2 edition, 2006, https://link.springer.com/book/10.1007/978-0-387-40065-5.

The aim is to give a concise summary of all the concepts you should become comfortable with to succeed in the course. The textbook provides more context, examples, counter-examples, illustrations etc., which we also explore during lectures and exercise sessions.

Other sources used quite often in preparing these lecture notes include:

- A.P. Ruszczyński. *Nonlinear optimization*. Princeton University Press, Princeton, NJ, 2006,
- D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.

Special thanks to Quentin Rebjock and Axel Séguin who were my TAs for the first year teaching this course.

These notes are always work in progress. Please do let me know about errors, typos, suggestions for improvements, ... (however small.) Your feedback is welcome, always.

Nicolas Boumal

Chapter 1

Introduction

This course is about solving problems of the form

$$\min_{x \in S} f(x)$$

for **a set** S and **a function** $f\colon S \to \mathbb{R}$. That is, we aim to find $x \in S$ such that $f(x)$ is as small as possible.

The following notations are often encountered to describe such a problem:

$$\min_{x \in S} f(x), \qquad \min_{x} f(x) \text{ subject to } x \in S, \qquad \arg\min_{x \in S} f(x).$$

The first two also represent the **optimal value** (that is, the smallest possible value $f(x)$ can take with $x \in S$). The third one also represents the **optimal set** (that is, the set of $x \in S$ such that $f(x)$ is minimal.)

The next section highlights the point that almost every computational task can be written as an optimization problem. But of course, many computational tasks are really, really hard to solve. Therefore, it must be that optimization in general is hard. The key then is (a) to require mathematical structure we could exploit (that is, make some assumptions about S and f), and (b) to adjust our targets (unfortunately, the sky is not the limit in general, but sometimes good things happen).

In this course, we assume that S is a continuous (as opposed to discrete) subset of a Euclidean space, and that f is differentiable. This provides ample mathematical structure for a first dive into continuous optimization.

Exercise 1.1. *What if what we want is to* maximize *a function $g\colon S \to \mathbb{R}$? Check that the problems $\max_{x \in S} g(x)$ and $\min_{x \in S} f(x)$ have the same set of solutions if we let $f = -g$, and check how their optimal values are related. For this reason, we almost exclusively discuss minimization problems. This convention will be particularly convenient when we get to convexity and to second-order optimality conditions later on.*

Remark 1.2. *We should bear in mind that f might not actually attain a minimal value on S, in which case the* 'arg min' *is empty and the* 'min' *should really be an* 'inf' *(infimum). However, because optimization problems make the most sense when they have a solution, it is habitual to just write* 'min' *while keeping the possibility of an empty solution set in mind at all times.*

1.1 Optimization is a powerful modeling tool

Optimization problems in general come in mostly two flavors:

1. **Discrete** problems: the set S is discrete. It contains a finite or countably infinite number of elements. In principle, we could evaluate f for each element of S in turn, but typically that is impossibly expensive computationally. The game then is to devise clever algorithms to explore more promising parts of S first. This is the terrain of combinatorial mathematics.

2. **Continuous** problems: the set S is continuous. It is fundamentally impossible to enumerate all possible solutions for the same reason that the real numbers cannot be enumerated. What complicates matters further is that S is typically a subet of a high dimensional space (say, $\mathbb{R}^n$ with large n). The game then is to devise iterative algorithms that hopefully converge to interesting points in S. This is the terrain of numerical analysis.

Discrete optimization comes up when we must make discrete choices (on/off, yes/no, select a particular subset of objects in a bigger set, order objects in a certain way, ...) and we have a clear way to assign a value to each possible choice. Typical examples include:

- Facility location selection (e.g., warehouses, drop-off points),
- Scheduling (e.g., trains, deliveries, traffic lights, exams),
- Path finding on graphs (e.g., google maps, NPCs in video games),
- ...

We do not discuss these at all in this course.

Continuous optimization comes up when the variables are best described using real numbers (signals, images, distances, orientations, intensities, flows, voltages, statistical features, hyper-parameters, fractions, ...) and we have a clear way to assign a value to each possible choice. Here are typical examples:

- **Regression** is the task of finding a function g which approximately interpolates pairs (x_i, y_i) so that $g(x_i) \approx y_i$. We want this to hold both on given examples used to choose g (training) and on unseen, new examples (generalization). This describes much of machine learning, but is much older than that. The unknown here is a function g: that is uncomfortable. Instead, we choose a class of functions that can be described using a finite number of real parameters. For example, we can use polynomials. Alternatively, we can fix a certain architecture for a neural network, then what remains is to choose the weights of the neurons: these are real numbers. Finding the best function g in the chosen class amounts to finding real parameters that correspond to the best function: there is our optimization problem. How do we choose the class of functions? That is application-specific. In machine learning, engineers use deep neural networks of all kinds for different purposes; outside of machine learning, people also use different function classes.

- **Classification** is closely related to regression. A classical approach to this problem is support vector machines, which admits a clean formulation as a continuous optimization problem where the unknown is a plane separating two classes of objects (possibly nonlinearly mapped to a feature space first, to improve the chances that the two classes can be separated by a plane). A plane can be represented by a normal vector, and that vector has real components: those are our variables.

- In **inverse problems**, we are interested in a signal x^* (it could be an image in medical imaging, a 3-D shape in microscopy for structural biology, a message emitted through a distorted radio channel), and we obtain some measurements $y = \Phi(x^*)$, where Φ captures a physical model. Recovering x^* from y looks like a standard problem in numerical analysis, namely, solving (possibly nonlinear) equations. However, such tasks are often ill-conditioned, in the sense that there could exist x_1 and x_2 very different yet such that $\Phi(x_1) \approx \Phi(x_2) \approx y$. If that is so, it looks like both x_1 and x_2 might be reasonable estimators for x^*, but they are so different from each other... which one should we trust? In such situations, it is more fruitful to incorporate some prior knowledge we may have about x^* (for example, that it has sharp edges, or that it is elongated, or that it is low-pass) by finding x that simultaneously agrees with the measurements as much as possible ($\Phi(x) \approx y$) and is compatible with our prior. This is typically done through regularization; the problem might look something like this: $\min_{x \in S} \|\Phi(x) - y\|^2 + \lambda R(x)$, with a well-chosen R.

- **Statistical estimation problems**: these are similar to inverse problems, only with a statistical model on top, often leading to optimization via maximum likelihood estimation. There, a likelihood function L assesses how likely a particular x is given the measurements we have, the statistical model that relates the true unknown and those measurements, and any prior beliefs we have about the latter (think Bayes).

- **Resource allocation**: assuming your resources are continuous or nearly so—as may be the case for money or computational power—allocate a fraction of it to each of a number of options to optimize something (returns, balance between returns and risk, probability of success of a certain project, ...).

- **Optimal control**: choose which commands to send to motors, actuators, fans, heating devices, etc. so as to get a machine to do something optimally (e.g., drive something autonomously, produce glass panels with an industrial oven, air-condition a building). This is often done in a real-time loop with input from various sensors, and it involves a great deal of modeling to relate those inputs to the overall state of the system we aim to control. A prime example of optimization-based control is embodied by model predictive control.

- ...

We focus on continuous optimization. You will encounter simple applications for homework.

It should be noted that complex optimization problems as they come up in industry can easily involve a mix of continuous and discrete optimization variables (e.g., one may have to optimize the flow of gas through a distribution network of pipes, with the ability to switch particular compressors on or off to boost pressure where most needed). The data and models used to phrase these problems are also often affected by noise and uncertainty. And it is rather common to have to re-solve them repeatedly over time as the data evolve. Accordingly, consider this course an invitation; an introduction to a much bigger story.

1.2 With great generality comes great difficulty

Our hope is always to find a *global* minimizer.

Definition 1.3. *A* global minimizer *of* $f\colon S \to \mathbb{R}$ *is a point* $x^* \in S$ *such that* $f(x) \geq f(x^*)$ *for all* $x \in S$.

This, however, turns out to be too much to ask in most situations.

A less ambitious hope could be to find a *local* minimizer. To define this concept, we need a notion of what "local" means, that is, we need a *topology* on S.[1] It will always be clear from context which topology we use. For example, if S is a vector space such as $\mathbb{R}^n$, we use the usual topology. If S is a subset of a vector space, we use the subspace topology: a subset of S is open if it is the intersection of S with some open set of the vector space.

Definition 1.4. *A* neighborhood *of $x \in S$ is an open subset of S which contains x.*[2]

Definition 1.5. *A* local minimizer *of $f\colon S \to \mathbb{R}$ is a point $x^* \in S$ such that $f(x) \geq f(x^*)$ for all x in a neighborhood of x^* in S.*

In many practical situations, it is perfectly reasonable to hope that we might be able to find a local minimizer. However, even this is computationally hard in general.[3] One reason is that f might have wide, flat areas, where there are no local minimizers, yet f almost doesn't vary. Another reason is that it can be difficult to check (computationally) whether a given point x is or is not a local minimizer. In subsequent chapters, we introduce an even weaker class of points, using the concept of *necessary optimality conditions*.

We close this section with two additional definitions: they are subtly different restrictions on the concept of local minimizer. During precept, you will explore how they differ.

Definition 1.6. *A* strict local minimizer *of $f\colon S \to \mathbb{R}$ is a local minimizer x^* of f such that $f(x) > f(x^*)$ for all $x \neq x^*$ in a neighborhood of x^* in S.*

Definition 1.7. *An* isolated local minimizer *of $f\colon S \to \mathbb{R}$ is a local minimizer x^* of f such that, in some neighborhood of x^*, there are no other local minimizers of f.*

Based on the above definitions, we similarly define the notions of *global maximizer*, *local maximizer*, *strict local maximizer* and *isolated local maximizer*.

---

[1] We always assume that S is equipped with a Hausdorff, second-countable topology. If these words mean nothing to you, it's safe to ignore this footnote.

[2] Many authors define a neighborhood of x as a set which contains an open set which itself contains x; then, they would call our neighborhoods "open neighborhoods." We always work with open neighborhoods, so we prefer the concise naming convention.

[3] See for example the following wide-audience article in Quanta Magazine: https://www.quantamagazine.org/surprising-limits-discovered-in-quest-for-optimal-solutions-20211101/.

Part I

Unconstrained optimization

Chapter 2

The setup, the rules, the goal

In this first part of the course, we focus on solving *unconstrained* optimization problems, that is, we wish to compute a solution of

$$\min_{x \in \mathcal{E}} f(x), \tag{P}$$

where $\mathcal{E}$ is a Euclidean space and $f\colon \mathcal{E} \to \mathbb{R}$ is our cost function. If f is k-times continuously differentiable on $\mathcal{E}$, we say f is of class C^k and we write $f \in C^k(\mathcal{E})$. Unless otherwise stated, **we always assume f is continuously differentiable**, that is, $f \in C^1(\mathcal{E})$.

As we already discussed in the previous chapter, it is in general too difficult to find a global minimizer of (P). A notable exception to that sad state of affairs is the case where (P) is convex: more on that later. For now, we must content ourselves with a more modest goal.

Our revised goal is merely to find a point $x \in \mathcal{E}$ which satisfies certain necessary optimality conditions, defined below. The conditions are designed such that all minimizers (local and global) satisfy them.

Our hope is that, in doing so, we will actually often find a local or even a global minimizer—but only under special circumstances will we be able to *guarantee* that this hope is realized.

The rules of the game are that we may evaluate f at any point $x \in \mathcal{E}$ that we like. We are also allowed to evaluate the derivatives of f at any point: the gradient, but also the Hessian if it exists. However, we very much would like to "query" as few points as possible. The reason is simple really: evaluating f and its derivatives requires computational power and time; we have a limited amount of both.[1]

[1] In particular, even if f is a function of just one or two variables, we would not simply "plot the function and look at it." This is because to plot f we need to evaluate f at very many point, which is inefficient.

2.1 Reminders about Euclidean space

Let $\mathcal{E}$ be a real, finite dimensional vector space (also called a linear space). We can equip $\mathcal{E}$ with an inner product, as defined below. The pair $\mathcal{E}$ together with an inner product is what we call a Euclidean space.

Definition 2.1. *An* inner product *on $\mathcal{E}$ (also called a Euclidean metric) is a map $\langle \cdot, \cdot \rangle : \mathcal{E} \times \mathcal{E} \to \mathbb{R} \colon (u, v) \mapsto \langle u, v \rangle$ which satisfies the following properties:*

1. *Symmetry: $\langle u, v \rangle = \langle v, u \rangle$ for all $u, v \in \mathcal{E}$,*
2. *Bi-linearity: $\langle \alpha u + \beta v, w \rangle = \alpha \langle u, w \rangle + \beta \langle v, w \rangle$ for all $u, v, w \in \mathcal{E}$ and $\alpha, \beta \in \mathbb{R}$,*
3. *Positive definiteness: $\langle u, u \rangle \geq 0$ for all $u \in \mathcal{E}$ and $\langle u, u \rangle = 0$ if and only if $u = 0$.*

Definition 2.2. *A* Euclidean space *is a real, finite dimensional vector space $\mathcal{E}$ equipped with an inner product $\langle \cdot, \cdot \rangle$. The associated* Euclidean norm *is defined by $\|u\| = \sqrt{\langle u, u \rangle}$ for all $u \in \mathcal{E}$.*

Example 2.3. *The canonical Euclidean space is $\mathcal{E} = \mathbb{R}^n$ with*

$$\langle u, v \rangle = u^\top v = u_1 v_1 + \cdots + u_n v_n.$$

The associated Euclidean norm is the 2-norm: $\|u\| = \sqrt{u_1^2 + \cdots + u_n^2}$.

Example 2.4. *The space of matrices $\mathcal{E} = \mathbb{R}^{m \times n}$ is also a Euclidean space when equipped with the trace (or Frobenius) inner product*

$$\langle U, V \rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} U_{ij} V_{ij} = \mathrm{Tr}(U^\top V).$$

The associated Euclidean norm is the Frobenius norm: $\|U\| = \sqrt{\sum_{i,j} U_{ij}^2}$.

Example 2.5. *Any linear subspace of $\mathbb{R}^n$ or $\mathbb{R}^{m \times n}$ can be turned into a Euclidean space with the same inner products as above, only restricted to the subspace in question. For example, the space of symmetric matrices*

$$\mathcal{E} = \mathrm{Sym}(n) = \{X \in \mathbb{R}^{n \times n} : X = X^\top\}$$

is a Euclidean space when equipped with the trace inner product.

Theorem 2.6 (Cauchy–Schwarz)**.** *We have $|\langle u, v \rangle| \leq \|u\| \|v\|$ for all u, v in $\mathcal{E}$. Moreover, equality is attained exactly when u, v are colinear.*

A bit of linear algebra

Definition 2.7. *The* norm of a linear map $L\colon \mathcal{E} \to \mathcal{F}$ *is*

$$\|L\| = \max_{u \in \mathcal{E}, u \neq 0} \frac{\|L(u)\|_{\mathcal{F}}}{\|u\|_{\mathcal{E}}}$$

where $\|\cdot\|_{\mathcal{E}}$ and $\|\cdot\|_{\mathcal{F}}$ denote the norms on the Euclidean spaces $\mathcal{E}$ and $\mathcal{F}$, respectively. In other words, $\|L\|$ is the smallest real number such that

$$\forall u \in \mathcal{E}, \qquad \|L(u)\|_{\mathcal{F}} \leq \|L\|\|u\|_{\mathcal{E}}.$$

Definition 2.8. *The* adjoint *of a linear map $L\colon \mathcal{E} \to \mathcal{F}$ is the linear map $L^*\colon \mathcal{F} \to \mathcal{E}$ defined by the following property:*

$$\forall u \in \mathcal{E}, w \in \mathcal{F}, \qquad \langle L(u), w \rangle_{\mathcal{F}} = \langle u, L^*(w) \rangle_{\mathcal{E}},$$

where $\langle \cdot, \cdot \rangle_{\mathcal{E}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ denote the inner products on the Euclidean spaces $\mathcal{E}$ and $\mathcal{F}$, respectively.

Definition 2.9. *A linear map $A\colon \mathcal{E} \to \mathcal{E}$ on a Euclidean space $\mathcal{E}$ is called* symmetric *or* self-adjoint *if $A = A^*$, that is,*

$$\forall u, v \in \mathcal{E}, \qquad \langle A(u), v \rangle = \langle u, A(v) \rangle.$$

Definition 2.10. *A basis $u_1, \ldots, u_n$ is* orthonormal *for $\mathcal{E}$ if*

$$\forall\, 1 \leq i, j \leq n, \qquad \langle u_i, u_j \rangle = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Exercise 2.11 (Adjoint and transpose)**.** *Let $u_1, \ldots, u_n$ form an orthonormal basis of $\mathcal{E}$. Likewise, let $v_1, \ldots, v_m$ form an orthonormal basis of $\mathcal{F}$. Consider a linear operator $L\colon \mathcal{E} \to \mathcal{F}$. For each $1 \leq i \leq n$, the vector $L(u_i)$ is an element of $\mathcal{F}$; therefore, we may expand it in the basis v as follows:*

$$L(u_i) = \sum_{j=1}^{m} M_{ji} v_j,$$

where we collect the coefficients into a matrix $M \in \mathbb{R}^{m \times n}$. This matrix represents *L with respect to the chosen bases. Show that the matrix which represents L^* with respect to those same bases is $M^\top$: the transpose of M. In particular, a linear map $A\colon \mathcal{E} \to \mathcal{E}$ is symmetric if the matrix associated to it with respect to the basis $u_1, \ldots, u_n$ is symmetric.*

Theorem 2.12 (Spectral theorem)**.** *Let A be a symmetric linear map on a Euclidean space $\mathcal{E}$. Then, A admits an orthonormal basis of eigenvectors $u_1, \ldots, u_n \in \mathcal{E}$ associated to real eigenvalues $\lambda_1, \ldots, \lambda_n$, that is, $A(u_i) = \lambda_i u_i$ for each i.*

Exercise 2.13. *Show that for a symmetric linear map A on $\mathcal{E}$ with orthonormal basis of eigenvectors $u_1, \ldots, u_n$ and associated eigenvalues $\lambda_1, \ldots, \lambda_n$ we have:*

$$\forall v \in \mathcal{E}, \qquad A(v) = \sum_{i=1}^{n} \lambda_i \langle v, u_i \rangle u_i.$$

Exercise 2.14. *Let A be a symmetric linear map on $\mathcal{E}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$. Show that, for all $u \in \mathcal{E}$, we have:*

$$\lambda_{\min} \|u\|^2 \leq \langle u, A(u) \rangle \leq \lambda_{\max} \|u\|^2 \tag{2.1}$$

where $\lambda_{\min} = \min_{1 \leq k \leq n} \lambda_k$ and $\lambda_{\max} = \max_{1 \leq k \leq n} \lambda_k$. Further check the following expression for the operator norm of A:

$$\|A\| = \max_{u \in \mathcal{E}, \|u\|=1} |\langle u, A(u) \rangle| = \max_{1 \leq k \leq n} |\lambda_k|. \tag{2.2}$$

Definition 2.15. *Let $A \colon \mathcal{E} \to \mathcal{E}$ be a symmetric linear map. We say:*

1. *A is* positive semidefinite *if $\langle u, A(u) \rangle \geq 0$ for all $u \in \mathcal{E}$; we write $A \succeq 0$.*

2. *A is* positive definite *if $\langle u, A(u) \rangle > 0$ for all $u \in \mathcal{E}, u \neq 0$; we write $A \succ 0$.*

Exercise 2.16. *Show that $A \succ 0$ if and only if all eigenvalues of A are positive. Show that $A \succeq 0$ if and only if all eigenvalues of A are nonnegative.*

Exercise 2.17. *Let $L \colon \mathcal{E} \to \mathcal{F}$ be a linear map between two Euclidean spaces. Check that the map $A = L^* \circ L \colon \mathcal{E} \to \mathcal{E}$ is symmetric and positive semidefinite. Let $n = \dim \mathcal{E}$, so that A has n real eigenvalues: denote them $\lambda_1 \geq \cdots \geq \lambda_n$. The* singular values *of L are defined as $\sigma_i = \sqrt{\lambda_i}$ for $i = 1, \ldots, n$. Show that*

$$\sigma_{\min} \|u\|_{\mathcal{E}} \leq \|L(u)\|_{\mathcal{F}} \leq \sigma_{\max} \|u\|_{\mathcal{E}} \tag{2.3}$$

for all $u \in \mathcal{E}$, where $\sigma_{\max} = \sigma_1$ and $\sigma_{\min} = \sigma_n$. Further check that the operator norm of L is $\|L\| = \sigma_{\max}$.

A bit of multivariate calculus

Let $\mathcal{E}, \mathcal{F}$ be two Euclidean spaces. Consider a map $F\colon \mathcal{E} \to \mathcal{F}$. We say F is *differentiable* at $x \in \mathcal{E}$ if there exists a linear map $\mathrm{D}F(x)\colon \mathcal{E} \to \mathcal{F}$ (called the *differential* of F at x) such that

$$\lim_{v \to 0} \frac{\|F(x+v) - F(x) - \mathrm{D}F(x)[v]\|_{\mathcal{F}}}{\|v\|_{\mathcal{E}}} = 0, \tag{2.4}$$

where we write $\|\cdot\|_{\mathcal{E}}$ and $\|\cdot\|_{\mathcal{F}}$ to distinguish between the Euclidean norms on $\mathcal{E}$ and $\mathcal{F}$ respectively. We say F is differentiable if it is so at all $x \in \mathcal{E}$.

When F is differentiable at x, all the *directional derivatives* of F are defined at x, and we can write:

$$\mathrm{D}F(x)[v] = \lim_{t \to 0} \frac{F(x+tv) - F(x)}{t}. \tag{2.5}$$

If $F\colon \mathcal{E} \to \mathcal{F}$ is differentiable at $x \in \mathcal{E}$ and $G\colon \mathcal{F} \to \mathcal{H}$ is differentiable at $F(x) \in \mathcal{F}$, then $G \circ F$ is differentiable at x, and:

$$\mathrm{D}(G \circ F)(x) = \mathrm{D}G(F(x)) \circ \mathrm{D}F(x). \tag{2.6}$$

Equivalently, we can also write this as:

$$\forall v \in \mathcal{E}, \qquad \mathrm{D}(G \circ F)(x)[v] = \mathrm{D}G(F(x))[\mathrm{D}F(x)[v]]. \tag{2.7}$$

This is the *chain rule*. We use it all the time.

In particular, consider a differentiable function $f\colon \mathcal{E} \to \mathbb{R}$. Its differential at x is a linear map $\mathrm{D}f(x)$ from $\mathcal{E}$ to $\mathbb{R}$. Through the inner product, this map can be represented by a unique vector of $\mathcal{E}$, which we call the gradient of f at x.

Definition 2.18. *The* gradient *of a differentiable function $f\colon \mathcal{E} \to \mathbb{R}$ on a Euclidean space $\mathcal{E}$ is the map $\nabla f\colon \mathcal{E} \to \mathcal{E}$ defined by the following property:*

$$\forall x, v \in \mathcal{E}, \qquad \langle \nabla f(x), v \rangle = \mathrm{D}f(x)[v].$$

Exercise 2.19. *Show that Definition 2.18 is valid, that is, show that $\nabla f(x)$ exists under the stated conditions, and is unique.*

Exercise 2.20. *Let $\mathcal{E} = \mathbb{R}^n$ with the usual inner product. Show that (in this important particular case) $\nabla f(x) \in \mathbb{R}^n$ is the vector whose entries are the n partial derivatives of f with respect to $x_1, \ldots, x_n$: $\nabla f(x) = [\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n}]^\top$.*

Exercise 2.21. *Consider the following function* $f\colon \mathrm{Sym}(n) \to \mathbb{R}$*, where* $\mathrm{Sym}(n)$ *is equipped with the trace inner product:*

$$f(X) = \frac{1}{2}\|X - M\|^2,$$

where $M \in \mathbb{R}^{n\times n}$ *is not necessarily symmetric. Give an expression for* $\nabla f(X)$*. Be mindful that, by Definition 2.18,* $\nabla f(X)$ *must belong to* $\mathrm{Sym}(n)$*.*

We say $f\colon \mathcal{E} \to \mathbb{R}$ is twice differentiable if $\nabla f\colon \mathcal{E} \to \mathcal{E}$ is differentiable. When such is the case, the differential of ∇f at $x \in \mathcal{E}$, namely, $\mathrm{D}(\nabla f)(x)$ is a linear operator from $\mathcal{E}$ to $\mathcal{E}$, called the *Hessian* of f at x. For convenience, we denote it by $\nabla^2 f(x)$.

Definition 2.22. *The* Hessian *at* x *of a twice differentiable function* $f\colon \mathcal{E} \to \mathbb{R}$ *on a Euclidean space* $\mathcal{E}$ *is the linear map* $\nabla^2 f(x)\colon \mathcal{E} \to \mathcal{E}$ *defined by the following property:*

$$\forall v \in \mathcal{E}, \qquad \nabla^2 f(x)[v] = \mathrm{D}(\nabla f)(x)[v] = \lim_{t\to 0} \frac{\nabla f(x+tv) - \nabla f(x)}{t}.$$

Exercise 2.23. *Let* $\mathcal{E} = \mathbb{R}^n$ *with the usual inner product. Show that (in this important particular case)* $\nabla^2 f(x)$ *can be represented as a matrix of size* $n \times n$ *whose entry* (i, j) *is the second-order partial derivative of* f *with respect to* x_i *and* x_j*, often written* $\frac{\partial^2 f(x)}{\partial x_i \partial x_j}$*. We know that this matrix is symmetric.*

It is well known that the Hessian is a symmetric linear map (in the sense of Definition 2.9); this is directly related to the well-known fact that partial derivatives in $\mathbb{R}^n$ commute.

Theorem 2.24. *The Hessian* $\nabla^2 f(x)\colon \mathcal{E} \to \mathcal{E}$ *is a symmetric linear map.*

A function $f\colon \mathcal{E} \to \mathbb{R}$ is *twice continuously differentiable* if $x \mapsto \nabla^2 f(x)$ is continuous over $\mathcal{E}$. This is equivalent to the property that the map $(x, v) \mapsto \langle v, \nabla^2 f(x)[v]\rangle$ is continuous over $\mathcal{E} \times \mathcal{E}$.

We close with some multivariate corollaries of the usual Taylor theorem.

Theorem 2.25 (Taylor's Theorem)**.** *Let* $f\colon \mathcal{E} \to \mathbb{R}$ *be continuously differentiable. Given* $x, u \in \mathcal{E}$*, there exists* $t \in (0, 1)$ *such that*

$$f(x+u) = f(x) + \langle \nabla f(x+tu), u\rangle\,. \tag{2.8}$$

Moreover, if f *is twice continuously differentiable, we have*

$$\nabla f(x+u) = \nabla f(x) + \int_0^1 \nabla^2 f(x+tu)[u]\mathrm{d}t, \tag{2.9}$$

and also that there exists $t \in (0,1)$ *such that*

$$f(x+u) = f(x) + \langle \nabla f(x), u \rangle + \frac{1}{2} \left\langle u, \nabla^2 f(x+tu)[u] \right\rangle . \tag{2.10}$$

Proof. These expressions can be recovered from the "1-D" Taylor theorem you surely know. Indeed, to establish the first expression, let $g(t) = f(x+tu)$. By the chain rule, we have:

$$g'(t) = \mathrm{D}f(x+tu)[u] = \langle \nabla f(x+tu), u \rangle .$$

Taylor's theorem provides $g(1) = g(0) + g'(t)$ for some $t \in (0,1)$ (this is nothing but the mean value theorem). Plug in the expressions for $g(1)$, $g(0)$ and $g'(t)$ and this becomes (2.8).

Likewise, we have $g''(t) = \langle \nabla^2 f(x+tu)[u], u \rangle$. Taylor's theorem provides $g(1) = g(0) + g'(0) + \frac{1}{2}g''(t)$ for some $t \in (0,1)$, which becomes (2.10).

To recover (2.9), pick an orthonormal basis $v_1, \ldots, v_n$ of $\mathcal{E}$ and define $g_i(t) = \langle \nabla f(x+tu), v_i \rangle$ for all i. The fundamental theorem of calculus says:

$$g_i(1) = g_i(0) + \int_0^1 g_i'(t) \mathrm{d}t. \tag{2.11}$$

Equivalently, this means:

$$\langle \nabla f(x+u), v_i \rangle = \langle \nabla f(x), v_i \rangle + \int_0^1 \left\langle \nabla^2 f(x+tu)[u], v_i \right\rangle \mathrm{d}t. \tag{2.12}$$

Multiply this identity by v_i and sum over i to conclude. □

2.2 Optimality conditions

Recall that a point $x^* \in \mathcal{E}$ is a local minimizer for (P) exactly if there exists a neighborhood $\mathcal{N}$ of x^* in $\mathcal{E}$ such that $f(x) \geq f(x^*)$ for all $x \in \mathcal{N}$.

When f is differentiable, we can identify certain properties that all local (and global) minimizers have. We call them *necessary* optimality conditions. Points which satisfy those conditions are called *critical* (or *stationary*) points, of first- or second-order.

Theorem 2.26. *Let* $f : \mathcal{E} \to \mathbb{R}$ *be continuously differentiable. If* x^* *is a local minimizer for* f*, then* $\nabla f(x^*) = 0$.

Proof. For contradiction, assume $\nabla f(x^*) \neq 0$. Let $u = -\nabla f(x^*)$ and define $g(t) = f(x^* + tu)$. By the chain rule we have

$$g'(t) = \mathrm{D}f(x^*+tu)[u] = \langle \nabla f(x^*+tu), u \rangle .$$

The fundamental theorem of calculus provides:

$$f(x^* + tu) = g(t) = g(0) + \int_0^t g'(\tau)\mathrm{d}\tau = f(x^*) + \int_0^t \langle \nabla f(x^* + \tau u), u\rangle \, \mathrm{d}\tau.$$

Notice that $\langle \nabla f(x^*), u\rangle = -\|\nabla f(x^*)\|^2 < 0$. Since ∇f is continuous, there exists a scalar $T > 0$ such that $\langle \nabla f(x^* + tu), u\rangle < 0$ for all $t \in [0, T]$. Thus,

$$f(x^* + \tau u) < f(x^*) \text{ for all } \tau \in (0, T].$$

This is a contradiction because x^* is a local minimizer.

Let us be more formal about this last step: By definition, since x^* is a local minimizer there exists a neighborhood $\mathcal{N}$ of x^* (that is, an open set which contains x^*) such that $f(x) \geq f(x^*)$ for all $x \in \mathcal{N}$. The line $c(t) = x^* + tu$ intersects $\mathcal{N}$. More precisely, since c is continuous, it holds that $c^{-1}(\mathcal{N})$ is open; and since $c(0) = x^*$ is in $\mathcal{N}$, it holds that $c^{-1}(\mathcal{N})$ contains 0. Thus, there exists $0 < \varepsilon \leq T$ such that $c(\varepsilon)$ is in $\mathcal{N}$. This implies $f(x^* + \varepsilon u) = f(c(\varepsilon)) \geq f(x^*)$. On the other hand, $\varepsilon \leq T$ implies $f(x^* + \varepsilon u) < f(x^*)$: a contradiction indeed. □

Theorem 2.27. *Let $f : \mathcal{E} \to \mathbb{R}$ be twice continuously differentiable. If x^* is a local minimizer for f, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succeq 0$.*

Proof. By Theorem 2.26, we already know that $\nabla f(x^*) = 0$. For contradiction, assume $\nabla^2 f(x^*)$ is not positive semidefinite. Then, there exists $u \in \mathcal{E}$ such that $\langle u, \nabla^2 f(x^*)[u]\rangle < 0$. Since $\nabla^2 f$ is continuous, there exists a scalar $T > 0$ such that $\langle u, \nabla^2 f(x^* + tu)[u]\rangle < 0$ for all $t \in [0, T]$. Moreover, Taylor's theorem (2.10) tells us that, for all $\tau > 0$, there exists $t \in (0, \tau)$ such that

$$f(x^* + \tau u) = f(x^*) + \frac{\tau^2}{2} \left\langle u, \nabla^2 f(x^* + tu)[u] \right\rangle.$$

We deduce that $f(x^* + \tau u) < f(x^*)$ for all $\tau \in (0, T]$. This is a contradiction because x^* is a local minizer. □

Definition 2.28. *A point $x \in \mathcal{E}$ such that $\nabla f(x) = 0$ is called a* (first-order) critical point. *A point $x \in \mathcal{E}$ such that $\nabla f(x) = 0$ and $\nabla^2 f(x) \succeq 0$ is called a* second-order critical point.

Definition 2.29. *A* saddle point *is a critical point which is neither a local minimizer nor a local maximizer.*

Exercise 2.30. *Consider $f : \mathbb{R}^2 \to \mathbb{R}$ defined by $f(x, y) = x^2 - y^2$. Show that $(0, 0)$ is a saddle point but not a second-order critical point. Now consider $f : \mathbb{R}^2 \to \mathbb{R}$ defined by $f(x, y) = x^2 + y^3$. Show that $(0, 0)$ is a saddle point and also a second-order critical point.*

The other way around, we have a sufficient condition for local optimality if we require a stronger condition on the Hessian.

Theorem 2.31. *Let $f\colon \mathcal{E} \to \mathbb{R}$ be twice continuously differentiable. If $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succ 0$, then x^* is an isolated local minimizer.*

Proof. By continuity of $\nabla^2 f$, we can select $r > 0$ such that $\nabla^2 f(z) \succ 0$ for all z in the Euclidean ball $\mathcal{N} = \{x \in \mathcal{E} : \|x - x^*\| < r\}$ of radius r and centered around x^*. Then, by Taylor's theorem (2.10) we deduce that

$$\begin{aligned} \forall x \in \mathcal{N}\backslash\{x^*\}, \quad f(x) &= f(x^* + (x - x^*)) \\ &= f(x^*) + \frac{1}{2}\left\langle x - x^*, \nabla^2 f(z)[x - x^*]\right\rangle \text{ for some } z \in \mathcal{N} \\ &> f(x^*). \end{aligned}$$

This confirms that x^* is a strict local minimizer of f. It is in fact also an isolated local minimizer for f. Indeed, if there was another local minimizer in $\mathcal{N}$ (call it y), then in particular we would have $\nabla f(y) = 0$. Yet, by Taylor's theorem (2.9) we have

$$\nabla f(y) = \nabla f(x^*) + \int_0^1 \nabla^2 f(x^* + t(y - x^*))[y - x^*]\mathrm{d}t.$$

Since $x^* + t(y - x^*)$ is in $\mathcal{N}$ for all $t \in [0, 1]$, and since $\nabla f(y) = \nabla f(x^*) = 0$, upon taking an inner product of the above with $y - x^* \neq 0$, it follows that

$$0 = \int_0^1 \left\langle y - x^*, \nabla^2 f(x^* + t(y - x^*))[y - x^*]\right\rangle \mathrm{d}t > 0,$$

a contradiction. □

Exercise 2.32. *Here are functions with surprising properties:*

1. *One might expect that if a function has two local minima, then somewhere in between them there ought to be a local maximum or a saddle point. That's not true.*[2] *Check it with $f(x, y) = (x^2y - x - 1)^2 + (x^2 - 1)^2$.*

2. *Check that $f(x, y) = x^2 + y^2(1 + x)^3$ has a single critical point, that this critical point is a strict local minimum, and yet that it is not a global minimum.*[3]

[2] https://www.johndcook.com/blog/2017/10/04/no-critical-point-between-two-peaks/ and https://arxiv.org/abs/1302.0759.

[3] https://www.math.tamu.edu/~tom.vogel/gallery/node16.html

3. *Plot the function* $f(x, y) = x^3 - 3xy^2$ *around the origin. Notice one can descend away from that saddle point along three directions, not just two. This is called a monkey saddle.*[4]

4. *Consider the function* $f(x, y) = (y^2 - x^3)(y^2 - 4x^3)$. *Check that* $(0, 0)$ *is not a local minimizer for* f. *And yet, check that if* $c\colon \mathbb{R} \to \mathbb{R}^2$ *is any curve which is twice continuously differentiable and which satisfies* $c(0) = (0, 0)$ *and* $c'(0) \neq (0, 0)$ *then* $t = 0$ *is a local minimizer for* $f \circ c\colon \mathbb{R} \to \mathbb{R}$. *Hint: Taylor expand* c. *(This is not true if we only require one continuous derivative for* c.*)*[5]

5. *One might expect that if* x *is not a local minimum then it is possible to move away from it in such a way that the cost function value only goes down (not strictly). That's not true. Indeed, let* $f(x) = \sin(1/x)x^2$. *Check that* f *is continuously differentiable and that* $x = 0$ *is not a local minimum. Yet, show that there does not exist a continuous curve* $c\colon [0, 1] \to \mathbb{R}$ *such that* $c(0) = 0$ *and* $f(c(t)) \leq f(0)$ *for all* $t \in [0, 1]$.

[4] https://en.wikipedia.org/wiki/Monkey_saddle

[5] See Udriste's 1994 book, Thm. 8.8 + remarks, https://tinyurl.com/2tm59a6d. Already in the first textbook on optimization (Harris, 1917), a similar observation is made about the fact that it is insufficient to study f along straight lines. Harris gives historical notes pointing to the fact that Lagrange himself made that mistake, and many after him as a result: https://archive.org/details/theoryofmaximami00hancuoft/page/32/mode/2up, page 33. According to Harris, it is Peano who set the record straight.

Chapter 3

Gradient descent

Optimization algorithms are iterative: they require an initial point (often called an initial guess) x_0, and they use this initial point to generate a sequence of points $x_0, x_1, x_2, \ldots$ Our goal is to design the algorithm in such a way that the sequence $(x_k)_{k\geq 0}$ converges to—or at least admits accumulation points which are—points of interest for our purpose. The absolute best scenario is if we can find a global minimizer. More commonly we should be content to find a local minimizer. And for theoretical purposes, we will already be happy if we can guarantee that we find critical points.

Gradient descent (GD) is probably the most famous and most versatile optimization algorithm to (try to) solve problem (P). Given an initial point $x_0 \in \mathcal{E}$, GD iterates the following:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \tag{GD}$$

where $\alpha_k > 0$ is called the *step-length* or *step-size*. There are several possible strategies to choose the step-length.

GD is also called *steepest descent* because, locally, following the negative gradient induces the steepest decrease in the cost function f (up to first order approximation). Indeed, on the one hand Cauchy–Schwarz tells us that

$$\mathrm{D}f(x)[v] = \langle \nabla f(x), v\rangle \geq -\|\nabla f(x)\|$$

for all $v \in \mathcal{E}$ with $\|v\| = 1$, and on the other hand we can verify that this bound is attained when we set $v = -\frac{1}{\|\nabla f(x)\|}\nabla f(x)$.

3.1 Lipschitz continuous gradients

GD bases its decisions on the gradient of f. Specifically, if the current iterate is $x_k \in \mathcal{E}$, then x_{k+1} is constructed using $\nabla f(x_k)$. If ∇f is discontinuous,

Algorithm 3.1 Gradient descent

Input: $x_0 \in \mathcal{E}$
for k in $0, 1, 2 \ldots$ **do**
    Pick a step-length α_k (we discuss several options)
    $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
end for

or even if it is continuous but varies in a wild manner, then these decisions are unstable, in the sense that if x_k had been a little bit different, then x_{k+1} might end up being very different. For the purpose of analyzing GD, it would be uncomfortable to allow such behavior. The following definition captures a condition under which the gradient of f does not vary too fast.

Definition 3.1. *The gradient of f is* L-Lipschitz continuous *if*[1]

$$\forall x, y \in \mathcal{E}, \qquad \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \tag{3.1}$$

where $L \geq 0$ is a constant.

Many cost functions f encountered in practice satisfy (3.1), though perhaps not for *all* $x, y \in \mathcal{E}$. This is fine: we really only need (3.1) to hold in the regions that our algorithms explore. Let us keep that in mind, and proceed with (3.1) as stated to keep things simple.

When f has Lipschitz continuous gradient, we get excellent control over its local behavior. The following theorem is tremendously useful: it gives us uniform control over the error term in first-order Taylor expansions of f.

Theorem 3.2. *Let $f \colon \mathcal{E} \to \mathbb{R}$ be continuously differentiable. If ∇f is L-Lipschitz continuous, then*

$$\forall x, u \in \mathcal{E}, \qquad f(x + u) \leq f(x) + \langle \nabla f(x), u \rangle + \frac{L}{2}\|u\|^2.$$

Proof. With $c(t) = x + tu$, consider the function $g = f \circ c \colon \mathbb{R} \to \mathbb{R}$. This function satisfies $g(0) = f(x)$ and $g(1) = f(x + u)$. Moreover, g is continuously differentiable. This allows us to invoke the fundamental theorem of calculus to claim:

$$g(1) - g(0) = \int_0^1 g'(t)\mathrm{d}t = \int_0^1 \mathrm{D}f(c(t))[c'(t)]\mathrm{d}t = \int_0^1 \langle \nabla f(c(t)), c'(t) \rangle \, \mathrm{d}t.$$

[1] In the research literature, especially in computer science, this is often referred to as L-*smoothness*, and some authors define smoothness as meaning "Lipschitz continuous gradient"; we reserve the word "smooth" for functions that are infinitely many times differentiable (C^∞).

Plugging in our expressions for $g(0)$, $g(1)$, $c(t)$ and $c'(t) = u$, it follows that:

$$\begin{aligned} f(x+u) - f(x) &= \int_0^1 \langle \nabla f(x+tu), u \rangle \, \mathrm{d}t \\ &= \langle \nabla f(x), u \rangle + \int_0^1 \langle \nabla f(x+tu) - \nabla f(x), u \rangle \, \mathrm{d}t. \end{aligned}$$

The integrand is easily bounded using Cauchy–Schwarz and Lipschitzness:

$$\langle \nabla f(x+tu) - \nabla f(x), u \rangle \leq \|\nabla f(x+tu) - \nabla f(x)\| \|u\| \leq L\|tu\|\|u\| = L|t|\|u\|^2.$$

It follows that

$$f(x+u) - f(x) - \langle \nabla f(x), u \rangle \leq L\|u\|^2 \int_0^1 t \mathrm{d}t = \frac{L}{2}\|u\|^2.$$

Since x, u are arbitrary, this completes the proof. □

The following result shows that the inequalities in Theorem 3.2 essentially characterize the Lipschitz gradient property. Moreover, we get a convenient criterion to establish that property: it is sufficient to check that the Hessian of f has bounded norm.

Theorem 3.3. *Let $f \colon \mathcal{E} \to \mathbb{R}$ be twice continuously differentiable. Then, the following properties are equivalent:*

(a) $\|\nabla^2 f(x)\| \leq L$ *for all* $x \in \mathcal{E}$,

(b) ∇f *is* L*-Lipschitz continuous,*

(c) $|f(x+u) - f(x) - \langle \nabla f(x), u \rangle| \leq \frac{L}{2}\|u\|^2$ *for all* $x, u \in \mathcal{E}$.

Proof. To see that (b) implies (c), reconsider the proof of Theorem 3.2: only minor tweaks are necessary.

To see that (c) implies (a), consider the following Taylor expansion:

$$f(x+tu) = f(x) + t\langle \nabla f(x), u \rangle + \frac{t^2}{2}\left\langle u, \nabla^2 f(x)[u] \right\rangle + o(t^2).$$

(The notation $o(t^2)$ stands for a quantity such that $o(t^2)/t^2 \to 0$ when $t \to 0$, i.e., it tends to zero faster than t^2.) Rearrange the above then take absolute values to see that:

$$\frac{1}{2}|\langle u, \nabla^2 f(x)[u] \rangle| = \frac{|f(x+tu) - f(x) - \langle \nabla f(x), tu \rangle|}{t^2} + o(1).$$

Property (c) tells us that the numerator on the right-hand side is bounded by $\frac{L}{2}\|tu\|^2$. Therefore,

$$\frac{1}{2}|\langle u, \nabla^2 f(x)[u]\rangle| \leq \frac{L}{2}\|u\|^2 + o(1).$$

This holds for all t; thus, we may take $t \to 0$, at which point $o(1)$ vanishes (by definition). Since $\nabla^2 f(x)$ is a symmetric linear map, the property $|\langle u, \nabla^2 f(x)[u]\rangle| \leq L\|u\|^2$ for all u is equivalent to the property $\|\nabla^2 f(x)\| \leq L$ (see (2.2)).

It remains to show that (a) implies (b). To this end, we use Taylor's theorem, specifically, eq. (2.9), to see that

$$\begin{aligned}
\|\nabla f(x+u) - \nabla f(x)\| &= \left\| \int_0^1 \nabla^2 f(x+tu)[u] \mathrm{d}t \right\| \\
&\leq \int_0^1 \left\| \nabla^2 f(x+tu)[u] \right\| \mathrm{d}t \\
&\leq \int_0^1 \left\| \nabla^2 f(x+tu) \right\| \|u\| \mathrm{d}t \\
&\leq L\|u\|.
\end{aligned}$$

It is to reach the last inequality that we used property (a). □

3.2 GD with constant step-length: global behavior

We are now ready to analyze a first version of GD. Let us make the two following assumptions about $f\colon \mathcal{E} \to \mathbb{R}$.

A1. *There exists $f_{\text{low}} \in \mathbb{R}$ such that $f(x) \geq f_{\text{low}}$ for all $x \in \mathcal{E}$.*

A2. *The gradient of f is L-Lipschitz continuous.*

Using the constant L from A2, we define *GD with constant step-length* $1/L$ as follows: given an arbitrary $x_0 \in \mathcal{E}$, iterate

$$x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k). \tag{3.2}$$

Theorem 3.4. *Under assumptions A1 and A2, GD with constant step-length $1/L$ generates a sequence $(x_k)_{k\geq 0} \subset \mathcal{E}$ with the following properties:*

1. *Descent: $f(x_{k+1}) \leq f(x_k)$ for all k,*

2. *Rate: for all K,* $\min_{0\le k\le K-1}\|\nabla f(x_k)\| \le \sqrt{2L(f(x_0)-f_{\text{low}})}\frac{1}{\sqrt{K}}$,

3. *Accumulation points (if any) are critical:* $\lim_{k\to\infty}\|\nabla f(x_k)\| = 0$.

Proof. Let $u_k = \frac{1}{L}\nabla f(x_k)$. Through Theorem 3.2, the Lipschitz gradient assumption A2 provides us with the following inequalities for all k:

$$f(x_{k+1}) = f(x_k - u_k) \le f(x_k) - \langle \nabla f(x_k), u_k\rangle + \frac{L}{2}\|u_k\|^2.$$

Plugging in the expression for u_k and reorganizing, we get

$$f(x_k) - f(x_{k+1}) \ge \frac{1}{L}\|\nabla f(x_k)\|^2 - \frac{L}{2}\frac{1}{L^2}\|\nabla f(x_k)\|^2 = \frac{1}{2L}\|\nabla f(x_k)\|^2. \quad (3.3)$$

Not only does this show that the value of f is decreasing along the sequence (x_k), but also it shows that the value of f decreases by some substantial amount at each iteration where the gradient is not small.

Now using assumption A1 together with (3.3), a classic telescoping sum argument gives, for all K,

$$\begin{aligned} f(x_0) - f_{\text{low}} &\ge f(x_0) - f(x_K) \\ &= \sum_{k=0}^{K-1} f(x_k) - f(x_{k+1}) \\ &\ge \sum_{k=0}^{K-1} \frac{1}{2L}\|\nabla f(x_k)\|^2 \qquad (3.4) \\ &\ge \frac{K}{2L}\min_{0\le k\le K-1}\|\nabla f(x_k)\|^2. \end{aligned}$$

Reorganize to get the rate claim.

For the final claim, consider (3.4) again, namely:

$$2L(f(x_0) - f_{\text{low}}) \ge \sum_{k=0}^{K-1}\|\nabla f(x_k)\|^2.$$

This inequality holds for all K. The left-hand side is a constant (independent of K). Thus, taking $K\to\infty$, the series on the right-hand side must converge. In particular, the summands must converge to zero, i.e., $\|\nabla f(x_k)\|^2 \to 0$. Assume x is an accumulation point of (x_k), that is, there exists a subsequence (x_{k_ℓ}) of (x_k) which converges to x. Then, by continuity of the function $x\mapsto\|\nabla f(x)\|$ we see that

$$\|\nabla f(x)\| = \|\nabla f(\lim_{\ell\to\infty} x_{k_\ell})\| = \lim_{\ell\to\infty}\|\nabla f(x_{k_\ell})\| = 0.$$

In other words: accumulation points of (x_k) are critical points for f. □

A few remarks are in order to make sure we understand precisely what the above theorem does and does not say.

Remark 3.5. *The rate result can be restated equivalently as follows: for all $\varepsilon > 0$, there exists $k \leq \lceil 2L(f(x_0) - f_{\text{low}})\frac{1}{\varepsilon^2} \rceil$ such that $\|\nabla f(x_k)\| \leq \varepsilon$. This can be very slow! And yet, one can show that it is unimprovable unless we make more assumptions about f. We should remember that this statement is about the worst-case behavior: in practice, GD can perform quite a bit better.*

Remark 3.6. *Notice how the rate $\sqrt{2L(f(x_0) - f_{\text{low}})}/\sqrt{K}$ is* independent *of the dimension of $\mathcal{E}$. That is, insofar as the worst-case is concerned, it does not seem to matter whether we are minimizing a function of one, two or one billion variables! This is rather remarkable, and indeed provides some explanation as to why we can routinely solve high-dimensional optimization problems in applications (think machine learning models with billions of parameters). However, we should remember that computing f and its gradient is likely to incur a cost which grows with the dimension of $\mathcal{E}$. Moreover, it is not uncommon for L (the Lipschitz constant of the gradient) to grow with dimension too.*

Remark 3.7. *Let us stress this: there are* no *assumptions on x_0. On the other hand, the theorem only guarantees that all accumulation points of the sequence of iterates are critical points. It does* ***not*** *guarantee that the sequence converges: it might have more than one accumulation point, and it might even not have any accumulation point at all!*

Remark 3.8. *Even when GD converges to a single point, Theorem 3.4 most certainly does* not *guarantee that the limit points are local minimizers, let alone global minimizers. In principle, GD could converge to saddle points. However, such situations are unstable. While it is true that GD can slow down significantly in the vicinity of saddle points, it* usually *manages to escape them "eventually." This is why in casual conversation people often say that GD converges to local minima: it is not mathematically correct in full generality, but it is "true in spirit."*

Remark 3.9. *Theorem 3.4 is called a* global convergence *result for GD. This can be deceptive, and it is somewhat abusive:*

1. *In numerical analysis, we say an iterative method enjoys "global convergence" if the sequences it generates converge regardless of x_0, that is: however we initialize the algorithm, it will converge. This has nothing to do with global optimality. Let us stress this again: when we say that an optimization algorithm enjoys global convergence, we are* ***not***

saying that it converges to global optima. We are only saying that it converges regardless of how it is initialized. And indeed, Theorem 3.4 does not involve any assumptions on x_0. That being said:

2. *Theorem 3.4 does* ***not*** *prove that sequences generated by GD converge. However, it takes such a weird function f and initialization x_0 for GD not to converge to a single point that it is common in oral discussions and (to a lesser extent) in the literature to say that GD enjoys global convergence. A convenient turn of phrase is to say: "GD converges to critical points" (note the plural)—this way, if GD admits more than one accumulation point, we are in the clear. It is often easy to ensure that GD has at least one accumulation point, hence that turn of phrase is satisfactory for most purposes.*

Exercise 3.10. *Consider GD with constant step-length α, as: $x_{k+1} = x_k - \alpha\nabla f(x_k)$. Working through the proof of Theorem 3.4, verify that $\alpha = 1/L$ leads to the best rate.*

Exercise 3.11. *Consider $f : \mathbb{R} \to \mathbb{R}$ with $f(x) = \frac{1}{1+e^{-x}}$ (plot it). Show that f satisfies assumptions A1 and A2, and specify constants f_{low} and L. Check that GD with constant step-length $1/L$ generates a sequence which doesn't have any accumulation points, regardless of x_0.*

Exercise 3.12. *Let $f : \mathcal{E} \to \mathbb{R}$ be continuously differentiable. Assume $\{x \in \mathcal{E} : f(x) \leq f(x_0)\}$ is bounded and ∇f is L-Lipschitz continuous. Show that GD with step-length $1/L$ generates a sequence which has at least one accumulation point (and that all accumulation points are critical points).*

3.3 GD with constant step-length: local behavior

When is it the case that GD actually converges to a point? Assuming it does so, how fast is the convergence? Let us first review the notion of *local convergence rate*.

Definition 3.13. *Let $\theta_0, \theta_1, \theta_2, \ldots$ be a sequence converging to θ in $\mathbb{R}$. We say (θ_k) converges to θ* at least linearly *if there exists $r \in (0, 1)$ and there exist $\varepsilon_0, \varepsilon_1, \varepsilon_2, \ldots > 0$ such that*

$$\varepsilon_k \to 0, \qquad |\theta_k - \theta| \leq \varepsilon_k, \qquad \text{and} \qquad \lim_{k\to\infty} \frac{\varepsilon_{k+1}}{\varepsilon_k} = r.$$

The above is called (at least) *linear* convergence for the following reason: if we plot $\log|\theta_k - \theta|$ against the iteration count k, then the resulting points

are (asymptotically) upper-bounded by a line. The slope of that line is given by

$$\lim_{k\to\infty} \frac{\log(\varepsilon_{k+1}) - \log(\varepsilon_k)}{(k+1)-k} = \lim_{k\to\infty} \log\left(\frac{\varepsilon_{k+1}}{\varepsilon_k}\right) = \log(r). \tag{3.5}$$

This type of convergence is sometimes called "geometric" or "exponential".

Exercise 3.14. *Show that if (θ_k) converges to θ at least linearly then there exist constants $C \geq 0$ and $\sigma \in (0,1)$ such that $|\theta_k - \theta| \leq C\sigma^k$ for all $k \geq 0$. In particular, deduce that $|\theta_k - \theta|$ drops below any desired tolerance $\varepsilon > 0$ for k larger than $O(\log(1/\varepsilon))$.*

Theorem 3.15. *Assume $f\colon \mathcal{E} \to \mathbb{R}$ is twice continuously differentiable and has L-Lipschitz continuous gradient (A2). Suppose x^* is a strict local minimizer satisfying $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succ 0$.*

Consider the sequence (x_k) generated by GD with constant step-length $1/L$. There exists a neighborhood U of x^* such that if x_k is in U for some k then all subsequent iterates $x_{k+1}, x_{k+2}, \ldots$ are also in U.

In that scenario, the sequence (x_k) converges to x^*. Moreover, the quantity $f(x_k)$ converges to $f(x^*)$ at least linearly, with rate $r = 1 - \frac{1}{\kappa}$ where κ is the condition number of $\nabla^2 f(x^*)$, that is, the ratio of its largest to smallest singular values. Additionally, $\|\nabla f(x_k)\|$ converges to 0 at least linearly, with rate $r = \sqrt{1 - 1/\kappa} \approx 1 - \frac{1}{2\kappa}$.

Remark 3.16. *The global convergence rate in Theorem 3.4 suggests that it might take as many as $O(1/\varepsilon^2)$ iterations to find a point x where $\|\nabla f(x)\| \leq \varepsilon$. That turns out to be true (i.e., there* exist *functions for which GD is indeed that slow). If GD were* typically *that slow, it would be almost useless. Fortunately, Theorem 3.15 reveals that, while GD can be slow initially, it can also speed up eventually: Once the linear convergence rate kicks in, we can find points which satisfy $\|\nabla f(x)\| \leq \varepsilon$ with merely $O(\log(1/\varepsilon))$ additional iterations. See also Remark 4.35.*

3.4 Practical versions of GD

In practice, we rarely (if ever) run GD with a constant step-length. We almost always want to use a form of line-search method to choose the step size, thus iteration

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \tag{3.6}$$

Algorithm 3.2 Backtracking line-search. Typical values: $\rho = \frac{1}{2}$, $c = 10^{-4}$.

Input: $x \in \mathcal{E}$
Parameters: $\bar{\alpha} > 0$, $\rho \in (0,1)$, $c \in (0,1)$
Set $\alpha \leftarrow \bar{\alpha}$ ▷ Initialize with some step
while $f(x) - f(x - \alpha \nabla f(x)) < c\alpha \|\nabla f(x)\|^2$ **do**
    Set $\alpha \leftarrow \rho\alpha$ ▷ Make α smaller
end while
Return α.

where α_k is determined by line-search, that is, by investigation of the 1-D function $\phi(\cdot; x_k) \colon \mathbb{R} \to \mathbb{R}$ defined by

$$\phi(t; x) = f(x - t\nabla f(x)). \tag{3.7}$$

One particularly simple method is called the *backtracking line-search method*, see Algorithm 3.2. The spirit is:

1. We try with some step-length $\bar{\alpha}$ (it may be bad: we have little information for now);

2. Then we check whether using that step-length makes the value of f decrease sufficiently. What is sufficient? That is decided by the so-called Armijo criterion (see the algorithm).

3. If yes, we return that step-length. If not, we make the tentative step-length smaller (multiplying our current guess by a constant $\rho \in (0,1)$), and we repeat until success.

Assuming f has L-Lipschitz continuous gradient, it's easy to see based on a drawing that there exists a whole interval of values for α that satisfy the Armijo condition. Therefore, either $\bar{\alpha}$ is already in that interval and we stop immediately, or the process of iteratively making α smaller will eventually lead us to that interval and we will stop.

You should read more about line-search methods in [NW06, Ch. 3] (Algorithm 3.1 in that book corresponds to Algorithm 3.2 here.) You will experiment with this in exercise sessions and homework. The essential guarantees we have obtained for GD with constant step-length (namely, global and local convergence results) still hold when we use a proper line-search method.

Another point of importance in practice when using any iterative algorithm is: when do we stop? That is: what do we use as a stopping criterion? One needs to be pragmatic about this. Popular options include combinations of any or all of the following (in no particular order):

- Maximum run time,
- Maximum number of iterations,
- Threshold on the gradient norm,
- Threshold on the cost function value,
- Threshold on the amount of change in cost function value or gradient norm amortized over the last few iterations (i.e.: stop if it seems we are no longer making substantial progress),
- ...

Tuning such stopping criteria is part of the algorithm design. It is a bit of an art.

Chapter 4

Convex functions

Much can go wrong when we try to minimize a function $f\colon \mathcal{E} \to \mathbb{R}$. Ultimately, that is because we only get *local* access to f: we can evaluate f and its derivatives at a point x, and that gives us valuable information about what f looks like at and around x, but it tells us little of value about the *global* behavior of f. This is the main reason why optimization algorithms typically face the risk of getting stuck at a local minimum of f: blind-spots.

To overcome such difficulties, we must know something more about f: we need additional assumptions.

One particularly fruitful assumption we can make is that of *convexity*. Graphically, the definition below expresses the requirement that the graph of f lies below its chords.

Definition 4.1. *Let $\mathcal{E}$ be a linear space. A function $f\colon \mathcal{E} \to \mathbb{R}$ is* convex *if*

$$f((1-t)x + ty) \leq (1-t)f(x) + tf(y)$$

for all $x, y \in \mathcal{E}$ and all $t \in [0,1]$.

There are several reasons why convexity is a delightful property to have in optimization. One of them is that convex functions do not have non-global local minima.

Theorem 4.2. *If f is convex and x^* is a local minimum for f, then x^* is a global minimum for f.*

Proof. For contradiction, assume x^* is not a global minimum of f. Then, there exists a point $y \in \mathcal{E}$ such that $f(y) < f(x^*)$. Convexity implies that the value of f is strictly decreasing along the line segment going from x^* to y. Indeed, for all $t \in (0,1]$:

$$f((1-t)x^* + ty) \leq (1-t)f(x^*) + tf(y) < (1-t)f(x^*) + tf(x^*) = f(x^*).$$

This contradicts the fact that x^* is a local minimum. □

Thus, we need not worry about optimization algorithms getting trapped in local minima of a convex f.

This is already great as such, but it doesn't do much for us unless (a) convex functions come up in applications, and (b) convex functions are easy to recognize. As it turns out, we can check both of these boxes.

In this chapter, we review some basic examples and properties of convex functions: this allows us to "spot" convexity in applications. Then, we study the behavior of algorithms on convex functions.

Convex analysis is a broad field which is of interest to mathematicians in both fundamental and applied research also beyond optimization. References for this chapter include [BV04, Roc70] and lecture notes by Stephen J. Wright.[1]

4.1 Basic definitions

In Definition 4.1, we stated what it means for a real-valued function f on a linear space $\mathcal{E}$ to be convex. Let us add a few additional related terms to our lexicon.

Definition 4.3. *A function $f : \mathcal{E} \to \mathbb{R}$ is* strictly convex *if*

$$f((1-t)x + ty) < (1-t)f(x) + tf(y)$$

for all $x, y \in \mathcal{E}$ distinct and all $t \in (0,1)$.

Exercise 4.4. *It is clear that if f is strictly convex then it is convex. On the other hand, give an example of a function which is convex yet is not strictly convex.*

Definition 4.5. *Assume $\mathcal{E}$ is a Euclidean space with norm $\|\cdot\|$. A function $f : \mathcal{E} \to \mathbb{R}$ is* μ-strongly convex *if $x \mapsto f(x) - \frac{\mu}{2}\|x\|^2$ is convex with $\mu > 0$.*

Exercise 4.6. *Show that if f is μ-strongly convex then f is strictly convex. On the other hand, check that $f(x) = x^4$ from $\mathbb{R}$ to $\mathbb{R}$ is strictly convex yet not strongly convex.*

Definition 4.7. *A function $f : \mathcal{E} \to \mathbb{R}$ is* concave *if $-f$ is convex. Likewise, f is* strictly *or* μ-strongly concave *if $-f$ is strictly or μ-strongly convex, respectively.*

Exercise 4.8. *Show that f is simultaneously convex and concave if and only if f is an affine function, that is, $f(x) = \langle w, x\rangle + b$ for some $w \in \mathcal{E}$ and $b \in \mathbb{R}$.*

[1] http://www.optimization-online.org/DB_FILE/2016/12/5748.pdf

Exercise 4.9. *Show that $f\colon \mathcal{E} \to \mathbb{R}$ is convex if and only if the restriction of f to lines is convex, that is, the univariate functions $t \mapsto f(x + t(y - x))$ are convex from $\mathbb{R}$ to $\mathbb{R}$ for all $x, y \in \mathcal{E}$. Argue the same for strict convexity. What can you say on this topic regarding strong convexity? What does that imply for line-search algorithms in optimization?*

Exercise 4.10. *Show that if f is convex then it satisfies what is called* Jensen's inequality*, namely:*

$$f(a_1x_1 + \ldots + a_nx_n) \leq a_1f(x_1) + \cdots + a_nf(x_n)$$

for all $x_1, \ldots, x_n \in \mathcal{E}$ and weights $a_1, \ldots, a_n \geq 0$ such that $a_1 + \cdots + a_n = 1$. Hint: proceed by induction on n.

Comment: We call $a_1x_1 + \cdots + a_nx_n$ a convex combination *of the points $x_1, \ldots, x_n$; notice that it is a kind of weighted average. Thus, the inequality says that the value of f at the weighted average of the base points is not more than the same weighted average of the values of f at the base points. This extends to continuous averages as well, into a powerful inequality of the form $f(\mathbf{E}X) \leq \mathbf{E}f(X)$, where X is a random variable and $\mathbf{E}$ denotes expectation.*

Exercise 4.11. *The arithmetic-mean–geometric-mean inequality states that*

$$\forall x_1, \ldots, x_n > 0, \qquad \frac{1}{n}(x_1 + \cdots + x_n) \geq \sqrt[n]{x_1 \cdots x_n}. \qquad \text{(AM-GM)}$$

Can you prove (AM-GM) using Jensen's inequality?

4.2 Recognizing convex functions

Beyond the absence of non-global local minima, part of the power of convexity for optimization is that it is often easy to spot. This is because many well-known functions are convex, and many common operations on such functions preserve their convexity. In this section, we consider such examples and operations.

For now, we only discuss convexity for functions whose domains are a whole linear space $\mathcal{E}$. Later in the course, we will discuss convex functions defined on subsets (which are themselves convex in a sense we shall define): that will further enhance our collection of convex functions.

Exercise 4.12. *Show that the following functions from $\mathbb{R}$ to $\mathbb{R}$ are convex.*

1. $f(x) = e^x$
2. $f(x) = x^2, f(x) = x^4, f(x) = x^6, \ldots$

3. $f(x) = |x|^a$ *with* $a \geq 1$

Show that the following functions from a Euclidean space $\mathcal{E}$ *to* $\mathbb{R}$ *are convex.*

4. $f(x) = \langle w, x \rangle + b$ *with* $w \in \mathcal{E}, b \in \mathbb{R}$

5. $f(x) = \frac{1}{2}\langle x, Ax \rangle + \langle b, x \rangle + c$ *with* $A \colon \mathcal{E} \to \mathcal{E}$ *a symmetric positive semidefinite linear map,* $b \in \mathcal{E}$, $c \in \mathbb{R}$.

Among all of these functions, which are strictly convex? Which are μ*-strongly convex and with which constant* μ*? (You might find this exercise easier to solve after reading the section about convexity and derivatives.)*

Exercise 4.13. *Show that if* f_1, f_2 *are two convex functions on* $\mathcal{E}$ *and* $a_1, a_2 \geq 0$ *are two nonnegative real numbers then* $f = a_1 f_1 + a_2 f_2$ *defined by*

$$f(x) = a_1 f_1(x) + a_2 f_2(x)$$

is a convex function on $\mathcal{E}$*. Extend the claim to* $f = a_1 f_1 + \cdots + a_k f_k$*. What can you say about strict or strong convexity of* f *depending on properties of* $f_1, \ldots, f_k$*?*

Exercise 4.14. *Show that if* f_1, f_2 *are two convex functions on* $\mathcal{E}$*, then the max of those two functions,* $f = \max(f_1, f_2)$ *defined by*

$$f(x) = \max(f_1(x), f_2(x)),$$

is convex. Extend your reasoning to $f = \max(f_1, \ldots, f_k)$*. Deduce that the function* $f(x) = |x|$ *is convex.*

Exercise 4.15. *Using some of the exercises above, show that the function* $f \colon \mathbb{R}^n \to \mathbb{R}$ *which to each vector* x *associates the sum of the* k *largest entries of* x *is convex.*

Exercise 4.16. *Based on some of the exercises above, show that the* log-sum-exp function *is convex (*$t > 0$ *is a fixed, real parameter):*

$$f(x) = t \log\left(\sum_{i=1}^{k} e^{x_i/t}\right). \tag{4.1}$$

(You might find this exercise easier to solve after reading the section about convexity and derivatives.)

This function is often used in applications because it is a smooth approximation of the maximum function. Indeed, with $\bar{x} = \max_i x_i$ *show that*

$$\bar{x} \leq f(x) = \bar{x} + t \log\left(\sum_{i=1}^{k} e^{\frac{x_i - \bar{x}}{t}}\right) \leq \bar{x} + t \log(k). \tag{4.2}$$

Thus, the smaller t *is, the better the approximation. However, from an optimization perspective (for example, if we plan to use gradient descent), can you see a reason why we should not take* t *too small?*

Note: on a computer, it is necessary to use expression (4.2) rather than expression (4.1) to compute f *(and its derivatives). Indeed, expression (4.1) can lead to overflow when* t *is small because it involves computing exponentials of possibly large numbers. In contrast, expression (4.2) only involves exponentials of nonpositive numbers.*

Exercise 4.17. *Let* $\mathcal{E}, \mathcal{F}$ *be two linear spaces. Let* $f\colon \mathcal{E} \to \mathbb{R}$ *be convex. Further let* $L\colon \mathcal{F} \to \mathcal{E}$ *be a linear map and let* $b \in \mathcal{E}$ *be arbitrary. Show that the function* $g\colon \mathcal{F} \to \mathbb{R}$ *defined by* $g(y) = f(Ly + b)$ *is convex. In other words: convexity is preserved under affine transformations.*

Recall that a norm on a linear space $\mathcal{E}$ is a map $\|\cdot\|_\square\colon \mathcal{E} \to \mathbb{R}$ satisfying the following properties:

1. $\|\alpha x\|_\square = |\alpha| \|x\|_\square$ for all $x \in \mathcal{E}, \alpha \in \mathbb{R}$,
2. $\|x + y\|_\square \leq \|x\|_\square + \|y\|_\square$,
3. $\|x\|_\square \geq 0$ for all x and $\|x\|_\square = 0 \iff x = 0$.

This include the Euclidean norm if $\mathcal{E}$ is equipped with a Eulidean structure of course, but also any other norm (e.g., the 1-norm and ∞-norm on $\mathbb{R}^n$.)

Exercise 4.18. *Show that any norm on a linear space* $\mathcal{E}$ *is convex.*

Exercise 4.19. *Let* $L \in \mathbb{R}^{m \times n}$ *and* $b \in \mathbb{R}^m$ *be arbitrary. Using some of the exercises above, show that the functions*

$$f_p(x) = \|Lx - b\|_p, \qquad p \in \{1, 2, \infty\}$$

are convex on $\mathbb{R}^n$*, where* $\|u\|_1 = |u_1| + \cdots + |u_m|$ *is the 1-norm on* $\mathbb{R}^m$*,* $\|u\|_2 = \sqrt{u_1^2 + \cdots + u_m^2}$ *is the 2-norm on* $\mathbb{R}^m$ *(which we usually simply denote by* $\|\cdot\|$ *as it is the Euclidean norm for* $\mathbb{R}^m$ *with the usual inner product) and* $\|u\|_\infty = \max(|u_1|, \ldots, |u_m|)$ *is the infinity-norm on* $\mathbb{R}^m$*.*

Exercise 4.20. *Consider functions $g\colon \mathcal{E} \to \mathbb{R}$ and $h\colon \mathbb{R} \to \mathbb{R}$. Let $f = h \circ g\colon \mathcal{E} \to \mathbb{R}$ be their composition. Show the following claims:*

1. *f is convex if: h is convex and nondecreasing and g is convex;*
2. *f is convex if: h is convex and nonincreasing and g is concave;*
3. *f is concave if: h is concave and nondecreasing and g is concave;*
4. *f is concave if: h is concave and nonincreasing and g is convex.*

(To develop some confidence in the claim, you may find it convenient to consider the case where both h and g are twice differentiable, and to use some of the results presented below regarding convexity and second derivatives.)

In particular, argue that if g is convex and nonnegative then $f(x) = g(x)^2$ is convex.

4.3 Convexity and derivatives

We can learn much about a function's derivatives if we know that it is convex. The other way around, we can learn much about a function's convexity by looking at its derivatives.

The first such result is a typical local-to-global property of convexity. Namely: the gradient of f at x (which is a *local* quantity) provides us with a *global* lower-bound on f. That global piece of information immediately translates into a striking result, namely, that the first-order necessary optimality condition is also a sufficient condition for global optimality when f is convex.

Theorem 4.21. *Assume $f\colon \mathcal{E} \to \mathbb{R}$ is differentiable on a Euclidean space $\mathcal{E}$. Then, f is convex if and only if*

$$\forall x, y \in \mathcal{E}, \qquad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle .$$

Moreover, f is strictly convex if and only if

$$\forall x, y \in \mathcal{E}, x \neq y, \qquad f(y) > f(x) + \langle \nabla f(x), y - x \rangle .$$

The proof strategy for Theorem 4.21 is to lean on Exercise 4.9. Indeed, we can use the latter to reduce the question to one about univariate functions: this is often useful when trying to establish properties of convexity. Accordingly, let us start with the following lemma, which is nothing but Theorem 4.21 but restricted to the univariate case.

Lemma 4.22. *Let $g\colon \mathbb{R} \to \mathbb{R}$ be differentiable. Then, g is convex if and only if*

$$\forall x, y \in \mathbb{R}, \qquad g(y) \geq g(x) + g'(x)(y - x).$$

Moreover, g is strictly convex if and only if

$$\forall x, y \in \mathbb{R}, x \neq y, \qquad g(y) > g(x) + g'(x)(y - x).$$

Proof of Lemma 4.22. Let us consider the case of convexity first (we will handle strict convexity afterwards.) Assume the inequalities hold. Then, for $x, y \in \mathbb{R}$ and $t \in [0, 1]$ arbitrary, define $z = (1 - t)x + ty$. Our assumption implies that both of the following inequalities hold:

$$g(x) \geq g(z) + g'(z)(x - z), \qquad g(y) \geq g(z) + g'(z)(y - z).$$

Add them up with weights $1 - t$ and t, respectively:

$$\begin{aligned}(1 - t)g(x) + tg(y) &\geq g(z) + g'(z)\big((1 - t)(x - z) + t(y - z)\big) \\ &= g(z) \\ &= g((1 - t)x + ty).\end{aligned}$$

This shows that g is convex. The other way around, if g is convex, then for all $x, y \in \mathbb{R}$ and $t \in (0, 1]$ we have

$$\begin{aligned}g(x + t(y - x)) &= g((1 - t)x + ty) \\ &\leq (1 - t)g(x) + tg(y) = g(x) + t(g(y) - g(x)).\end{aligned}$$

Move $g(x)$ to the left-hand side and divide by t to find:

$$g(y) \geq g(x) + \frac{g(x + t(y - x)) - g(x)}{t}.$$

Since this holds for all x, y, t as prescribed and since g is differentiable at x, we can take the limit for $t \to 0$ and conclude that the sought inequalities hold.

Let us now turn to the case of strict convex. If the strict inequalities in the lemma statement hold, then it is easy to show that g is strictly convex using the argument above with minimal changes. For the other direction, assume g is strictly convex. Then, it lies strictly below its chords, that is, for all x, y distinct in $\mathbb{R}$,

$$\forall t \in (0, 1), \qquad g((1 - t)x + ty) < (1 - t)g(x) + tg(y).$$

Since g is (in particular) convex, it also lies above its first-order approximations—that is what we proved above:

$$\forall t \in [0,1], \qquad g(x + t(y-x)) \geq g(x) + g'(x)t(y-x).$$

The left-hand sides of those inequalities coincide. Combine them to find:

$$\forall t \in (0,1), \qquad (1-t)g(x) + tg(y) > g(x) + g'(x)t(y-x).$$

Subtract $g(x)$ on both sides and divide by t to conclude. □

Proof of Theorem 4.21. Let $x, y \in \mathcal{E}$ be arbitrary, with $x \neq y$ (the case $x = y$ is easily treated separately). These two points define a line in $\mathcal{E}$ parameterized by

$$\gamma(t) = (1-t)x + ty.$$

Restricting f to that line by composition with γ, we get a univariate function:

$$g(t) = f(\gamma(t)), \qquad g'(t) = \mathrm{D}f(\gamma(t))[\gamma'(t)] = \langle \nabla f(\gamma(t)), y - x \rangle .$$

We have in particular that $g(0) = f(x)$, $g(1) = f(y)$ and $g'(0) = \langle \nabla f(x), y - x \rangle$. Thus, the following two inequalities are equivalent:

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle , \qquad g(1) \geq g(0) + g'(0).$$

This is the main observation en route to our conclusion, which we reach as follows:

1. If f is convex, then g is convex (Exercise 4.9) and Lemma 4.22 tells us that $g(1) \geq g(0) + g'(0)$, which in turn confirms that $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$. This holds for arbitrary x, y.

2. If $f(\tilde{y}) \geq f(\tilde{x}) + \langle \nabla f(\tilde{x}), \tilde{y} - \tilde{x} \rangle$ holds for all $\tilde{x}, \tilde{y}$, then it holds in particular for $\tilde{x} = \gamma(t_0)$ and $\tilde{y} = \gamma(t_1)$ with $t_0, t_1 \in \mathbb{R}$ arbitrary. Since

$$\begin{aligned} f(\tilde{x}) &= f(\gamma(t_0)) = g(t_0), & \tilde{y} - \tilde{x} &= (t_1 - t_0)(y - x), \\ f(\tilde{y}) &= f(\gamma(t_1)) = g(t_1), & \langle \nabla f(\tilde{x}), \tilde{y} - \tilde{x} \rangle &= g'(t_0)(t_1 - t_0), \end{aligned}$$

   we deduce that

$$\forall t_0, t_1 \in \mathbb{R}, \qquad g(t_1) \geq g(t_0) + g'(t_0)(t_1 - t_0).$$

   Lemma 4.22 tells us that g is convex. Since this holds for arbitrary lines, we conclude that f is convex (Exercise 4.9).

The argument for strict convexity follows as above with minimal changes. □

Corollary 4.23. *Let f be convex and differentiable on a Euclidean space. Then x is a global minimum for f if and only if $\nabla f(x) = 0$.*

Proof. We already know from Theorem 2.26 that if x is a global minimum then $\nabla f(x) = 0$. The other way around, if $\nabla f(x) = 0$, then Theorem 4.21 tells us that

$$\forall y \in \mathcal{E}, \qquad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle = f(x),$$

that is, x is a global minimum for f. □

Recall Theorem 3.2: that result gave us (convex) quadratic *upper* bounds on functions with Lipschitz continuous gradients. We have seen just now that convexity gives us linear *lower*-bounds. For strongly convex functions, we get (convex) quadratic lower-bounds as well.

Corollary 4.24. *Assume $f : \mathcal{E} \to \mathbb{R}$ is differentiable. Then f is μ-strongly convex if and only if*

$$\forall x, y \in \mathcal{E}, \qquad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2.$$

Proof. By definition, f is μ-strongly convex if and only if the function $g(x) = f(x) - \frac{\mu}{2}\|x\|^2$ is convex. Theorem 4.21 further tells us that g is convex if and only if

$$\forall x, y \in \mathcal{E}, \qquad g(y) \geq g(x) + \langle \nabla g(x), y - x \rangle .$$

Plugging in the expression for g and for $\nabla g(x) = \nabla f(x) - \mu x$, we see that this is equivalent to the following statement:

$$\forall x, y \in \mathcal{E}, \qquad f(y) - \frac{\mu}{2}\|y\|^2 \geq f(x) - \frac{\mu}{2}\|x\|^2 + \langle \nabla f(x) - \mu x, y - x \rangle .$$

We reach the claim by reorganizing the terms. □

The second class of results we state regarding convexity and differentiability relate the spectrum of the Hessian of f to its convexity properties.

Theorem 4.25. *Assume $f : \mathcal{E} \to \mathbb{R}$ is twice differentiable on a Euclidean space $\mathcal{E}$. Then,*

1. *f is convex if and only if $\nabla^2 f(x) \succeq 0$ for all $x \in \mathcal{E}$;*

2. *If $\nabla^2 f(x) \succ 0$ for all $x \in \mathcal{E}$ then f is strictly convex (but the converse need not hold);*

3. *f is μ-strongly convex if and only if $\nabla^2 f(x) \succeq \mu I$ for all $x \in \mathcal{E}$.*

Proof. For the second claim, notice that $f(x) = x^4$ from $\mathbb{R}$ to $\mathbb{R}$ is strictly convex yet $\nabla^2 f(0) = f''(0) = 0$ is not positive definite. The rest of the proof is left as an exercise. You may use Theorem 4.21, and you may want to proceed with a similar strategy (that is, exploiting Exercise 4.9 to reduce the theorem to a lemma about univariate functions first.) □

Exercise 4.26. *Consider $f(x) = x_1 x_2$ from $\mathbb{R}^2$ to $\mathbb{R}$. Show that f is not convex, even though it is a convex function of x_1 (with x_2 fixed) and it is also a convex function of x_2 (with x_1 fixed). This shows that* joint *convexity in $\mathbb{R}^n$ (being convex in all n variables simultaneously) is more demanding than simply being convex in the n individual variables separately.*

4.4 Global minima of convex functions

It is an exercise to check that the set of global minima of a convex function can be empty, that it can be a singleton, and that it can contain more than one point (we will have more to say about this in Section 9.2).

Strictly convex functions may or may not have a global minimum. However, they never have more than one.

Theorem 4.27. *If $f\colon \mathcal{E} \to \mathbb{R}$ is strictly convex, then there exists at most one global minimum $x^* \in \mathcal{E}$ for f.*

Proof. For contradiction, assume $x, y \in \mathcal{E}$ are distinct global minima of f, so that $f(x) = f(y) = f_{\min}$ and $f(z) \geq f_{\min}$ for all $z \in \mathcal{E}$. Then, for all $t \in (0, 1)$ we have

$$f_{\min} \leq f((1-t)x + ty) < (1-t)f(x) + tf(y) = f_{\min},$$

where the first inequality holds because $f_{\min}$ is the minimal value of f and the second inequality holds because f is strictly convex. This is a contradiction. □

Strongly convex functions admit a unique global minimum.

Theorem 4.28. *If $f\colon \mathcal{E} \to \mathbb{R}$ is strongly convex, then there exists exactly one global minimum $x^* \in \mathcal{E}$ for f.*

Proof. By Theorem 4.27, we know f admits at most one global minimum. It remains to show that it also admits at least one global minimum. Let us sketch the proof by assuming the following:[2]

1. f is continuous; and

2. There exists at least one point (call it x) where f is differentiable.

Using differentiability at x, Corollary 4.24 tells us that

$$\forall y \in \mathcal{E}, \qquad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2.$$

Define the ball $B = \{y \in \mathcal{E} : \|y - x\| \leq r\}$ with $r = \frac{4\|\nabla f(x)\|}{\mu}$. Then,

$$\begin{aligned} \forall y \in \mathcal{E}, y \notin B, \qquad f(y) &\geq f(x) - \|\nabla f(x)\|\|y - x\| + \frac{\mu}{2}\|y - x\|^2 \\ &= f(x) + \frac{\mu}{2}(\|y - x\|^2 - \frac{1}{2}r\|y - x\|) \\ &\geq f(x) + \frac{\mu r^2}{4}. \end{aligned}$$

(The first inequality is Cauchy–Schwarz; the equality is by substituting the definition of r to replace $\|\nabla f(x)\|$; and the last inequality holds because $\|y - x\| \geq r$.) Since B is compact and f is continuous, Weierstrass tells us that f attains its minimum in B, that is, there exists $x^* \in B$ such that $f(y) \geq f(x^*)$ for all $y \in B$. Moreover, x is in B. Thus, we can also say for all $y \notin B$ that $f(y) \geq f(x) \geq f(x^*)$. We summarize those statements as: there exists x^* such that $f(y) \geq f(x^*)$ for all $y \in \mathcal{E}$, that is, f admits a global minimum. □

Exercise 4.29. *Give an example of a smooth convex function which has more than one global minimum. Give an example of a smooth convex function which has exactly one global minimum.*

Exercise 4.30. *Give an example of a smooth, strictly convex function which is bounded below yet does not have any local minimum.*

[2] Though it would take some work to prove so, those assumptions are actually always satisfied: convexity for $f : \mathcal{E} \to \mathbb{R}$ implies that f is continuous, and that it is continuously differentiable on a dense subset of $\mathcal{E}$ [Roc70, Cor. 10.1.1,Thm. 25.5].

4.5 Gradient descent for strongly convex functions

If $f : \mathcal{E} \to \mathbb{R}$ is lower-bounded and differentiable with L-Lipschitz continuous gradient, we know that GD with constant step-size $1/L$ produces a sequence of iterates (x_k) in $\mathcal{E}$ whose accumulation points (if any) are critical points of f. Moreover, we know that if f is convex, then its critical points are its global minima. Thus, in this setting, all accumulation points of GD (if any) are global optima. This remains true if we implement a reasonable line-search algorithm to replace the fixed step-size $1/L$.

If f is also strongly convex, then we can make a much stronger claim. Indeed, we know that f has a unique minimum (Theorem 4.28), and moreover GD converges to that minimum at least linearly.

Theorem 4.31. *Let $f : \mathcal{E} \to \mathbb{R}$ be μ-strongly convex and have L-Lipschitz continuous gradient. Define $\kappa = \frac{L}{\mu}$. Let x^* be the (unique) global minimizer of f. Given an arbitrary $x_0 \in \mathcal{E}$, gradient descent with constant step-size $1/L$, that is, $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$ produces a sequence (x_k) which satisfies*

$$f(x_k) - f(x^*) \leq \left(1 - \frac{1}{\kappa}\right)^k (f(x_0) - f(x^*)),$$

that is, $f(x_k)$ converges at least linearly to the optimal value $f(x^*)$.

Proof. Through Theorem 3.2, L-Lipschitz continuous gradients give us the following upper-bounds on f:

$$\forall x, y \in \mathcal{E}, \qquad f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2.$$

In the general analysis of GD (Theorem 3.4), we used those upper-bounds to ensure that the value of f decreases by some non-trivial amount at each iteration, namely, we established (3.3):

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2L}\|\nabla f(x_k)\|^2. \tag{4.3}$$

Now equipped with the additional assumption of μ-strong convexity, we will show that $\|\nabla f(x_k)\|$ is quite large as long as we are far away from the global minimizer. We do so with the help of Corollary 4.24, which gives us the following lower-bounds on f:

$$\forall x, y \in \mathcal{E}, \qquad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2. \tag{4.4}$$

Think of x being fixed. Since the inequality holds for all y, it holds in particular if we minimize each side of the inequality with respect to y. The

left-hand side is minimized when we set $y = x^*$. The right-hand side (a quadratic) is minimized when we set $y = x - \frac{1}{\mu}\nabla f(x)$. When we plug these into the inequality, we learn the following:

$$\begin{aligned}\forall x \in \mathcal{E}, \qquad f(x^*) &\geq f(x) - \frac{1}{\mu}\langle \nabla f(x), \nabla f(x)\rangle + \frac{\mu}{2}\frac{1}{\mu^2}\|\nabla f(x)\|^2 \\ &= f(x) - \frac{1}{2\mu}\|\nabla f(x)\|^2.\end{aligned}$$

Stated differently,

$$\forall x \in \mathcal{E}, \qquad \|\nabla f(x)\|^2 \geq 2\mu(f(x) - f(x^*)). \tag{4.5}$$

Combine (4.3) and (4.5) with $x = x_k$ to reveal that

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2 \leq f(x_k) - \frac{\mu}{L}(f(x_k) - f(x^*)).$$

Subtract $f(x^*)$ on both sides to conclude that

$$f(x_{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{L}\right)(f(x_k) - f(x^*)),$$

as announced. (Conclude by induction on k.) □

Corollary 4.32. *In the setup of Theorem 4.31, we also have that (x_k) converges to x^* at least linearly, and that $\|\nabla f(x_k)\|$ converges to zero at least linearly. More precisely,*

$$\begin{aligned}\|x_k - x_*\| &\leq \sqrt{\frac{2}{\mu}(f(x_0) - f(x^*))}\sqrt{1 - \frac{1}{\kappa}}^k, \\ \|\nabla f(x_k)\| &\leq \sqrt{2L(f(x_0) - f(x^*))}\sqrt{1 - \frac{1}{\kappa}}^k.\end{aligned}$$

(For κ large, note that $\sqrt{1 - \frac{1}{\kappa}} \approx 1 - \frac{1}{2\kappa}$.)

Proof. From (4.3) we know that

$$\frac{1}{2L}\|\nabla f(x_k)\|^2 \leq f(x_k) - f(x_{k+1}) \leq f(x_k) - f(x^*). \tag{4.6}$$

From (4.4) with $x = x^*$ (noting that $\nabla f(x^*) = 0$) and $y = x_k$, we learn that:

$$\frac{\mu}{2}\|x_k - x^*\|^2 \leq f(x_k) - f(x^*). \tag{4.7}$$

Conclude using Theorem 4.31 to bound the right-hand sides.

Notice what happened here: for strongly convex functions with Lipschitz-continuous gradient, the optimality gap $f(x_k) - f(x^*)$ is indicative of both our distance to the optimum and of the norm of the gradient. □

Remark 4.33. *We call* $\kappa = \frac{L}{\mu}$ *the* condition number *of* f*. The reason for this is that if* f *is a quadratic, namely,*

$$f(x) = \frac{1}{2}\langle x, Ax\rangle - \langle b, x\rangle + c,$$

then f *is* μ*-strongly convex and has* L*-Lipschitz gradient if and only if* $\mu I \preceq A \preceq LI$ *(where* I *is the identity on* $\mathcal{E}$*). Therefore, we can set* $\mu = \lambda_{\min}(A)$ *and* $L = \lambda_{\max}(A)$*: the smallest and largest eigenvalues of* A*, respectively. Since* A *is positive definite, its eigenvalues coincide with its singular values. Then, the condition number of* A *(in the usual linear algebra sense of the word, i.e., the ratio of its largest to smallest singular values) is precisely* κ*. The gradient of* f *is* $\nabla f(x) = Ax - b$*. Therefore, the unique global minimizer of* f *is the solution of the linear system of equations* $Ax = b$*. We know from linear algebra that solving such a linear system is more complicated when the condition number of* A *is large. Here, we see that likewise minimizing* f *may be slower if the condition number of* A *is large.*

Remark 4.34. *Gradient descent is not the best gradient-based algorithm to minimize strongly convex functions with Lipschitz continuous gradients. There exist other methods, often generically called "accelerated gradient methods" which enjoy linear convergence to* x^* *at a rate controlled by* $1 - \frac{1}{\sqrt{\kappa}}$ *instead of* $1 - \frac{1}{\kappa}$*. In concrete terms, if* $\kappa \approx 10^6$*, we could expect accelerated methods to converge about a thousand times faster than standard GD. It is possible to show that no gradient-based algorithm can overcome a dependency on* $\sqrt{\kappa}$*.*

Remark 4.35. *Theorem 3.15 states (without proof) that GD enjoys at least linear convergence locally around critical points where the Hessian is positive definite. Theorem 4.31 sheds some light on that claim, as follows. Even if* f *is not convex globally, the following is true: if* x^* *is critical and* $\nabla^2 f(x^*)$ *is positive definite (say, all eigenvalues larger than* μ*), then there exists a ball around* x^* *in which the eigenvalues of the Hessian remain larger that* $.99\mu$*. Therefore,* f *is* 0.99μ*-strongly convex in that ball. It is possible to formalize the claim that once GD enters such a ball then it does not leave it. Thus, once it enters the ball, GD would only "see" a region where* f *is strongly convex: the fact that* f *lacks (strong) convexity outside of the ball no longer has any effect on the behavior of GD. In essence, this is why GD then enjoys at least linear convergence to* x^**.*

Chapter 5

Newton's method

Let $f\colon \mathcal{E} \to \mathbb{R}$ be differentiable with L-Lipschitz continuous gradient on a Euclidean space $\mathcal{E}$. We can think of gradient descent with constant step-size $1/L$ as doing the following: given x_0, iterate for $k = 0, 1, 2, \ldots$:

$$x_{k+1} = \arg\min_{x \in \mathcal{E}} \tilde{m}_k(x)$$

with

$$\tilde{m}_k(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2}\|x - x_k\|^2.$$

Indeed, the function $\tilde{m}_k\colon \mathcal{E} \to \mathbb{R}$ is a (strongly) convex quadratic whose unique minimizer is attained at $x_k - \frac{1}{L}\nabla f(x_k)$. We can think of $\tilde{m}_k$ as a quadratic *model* (an approximation) of f around x_k. (We also know from Theorem 3.2 that $\tilde{m}_k$ is an upper-bound on f.)

If f is twice differentiable, then Taylor's theorem suggests that we could consider a more accurate model for f around x_k. Namely, let us now consider

$$m_k(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2}\left\langle x - x_k, \nabla^2 f(x_k)[x - x_k] \right\rangle. \quad (5.1)$$

If $\nabla^2 f(x_k)$ is positive definite, then m_k is again a strongly convex quadratic. In that scenario, we know that the unique minimizer of m_k is attained at the point x where the gradient of m_k vanishes, that is:

$$0 = \nabla m_k(x) = \nabla f(x_k) + \nabla^2 f(x_k)[x - x_k].$$

The above defines a linear system of equations in $\mathcal{E}$ where $u_k = x - x_k$ is the unknown in $\mathcal{E}$:

$$\nabla^2 f(x_k)[u_k] = -\nabla f(x_k). \quad (5.2)$$

Algorithm 5.1 Newton's method

Input: $x_0 \in \mathcal{E}$
for k in $0, 1, 2 \ldots$ **do**
    Solve the Newton system $\nabla^2 f(x_k)[u_k] = -\nabla f(x_k)$ for $u_k \in \mathcal{E}$
    $x_{k+1} = x_k + u_k$
end for

Equation (5.2) is called the *Newton system* at x_k. *Newton's method* iterates the following given $x_0 \in \mathcal{E}$:

$$\text{For } k = 0, 1, 2, \ldots, \qquad x_{k+1} = x_k + u_k, \tag{5.3}$$

with u_k the solution of the Newton system (5.2). By design, this amounts to letting x_{k+1} be the unique global minimizer of m_k. See Algorithm 5.1.

5.1 Lipschitz continuous Hessians

For gradient descent, x_{k+1} depends on ∇f evaluated at x_k. If ∇f is discontinuous, or if it is continuous but can vary wildly nonetheless, then it is difficult to imagine how we could control the behavior of the iteration. This is why back then we introduced the assumption of Lipschitz continuous gradients A2.

For Newton's method, iteration (5.3) is mathematically equivalent to the following:[1]

$$x_{k+1} = x_k + u_k = x_k - (\nabla^2 f(x_k))^{-1}[\nabla f(x_k)]. \tag{5.4}$$

In the same spirit as above, in order to analyze the behavior of the iteration, we need some control over the continuity of ∇f and $\nabla^2 f$. To this end, we introduce the following notion.

Definition 5.1. *The Hessian of f is L'-*Lipschitz continuous *if*

$$\forall x, y \in \mathcal{E}, \qquad \|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L'\|x - y\|, \tag{5.5}$$

where $L' \geq 0$ is a constant. Note: the norm on the left-hand side is the operator norm in the Euclidean space $\mathcal{E}$; see the reminders in Section 2.1, specifically eq. (2.2).

[1] In practice that is not how we compute x_{k+1}, but more on this later.

Many cost functions f encountered in practice satisfy (5.5), though perhaps not for *all* $x, y \in \mathcal{E}$. This is fine: we really only need (5.5) to hold in the regions that our algorithms explore.

When f has Lipschitz continuous Hessian, we get excellent control over its local behavior. The following theorem gives us uniform control over the error term in second-order Taylor expansions of f, and also (more importantly for our purpose) over the error term in first-order Taylor expansions of ∇f. Indeed, the theorem below states (in a precise fashion) that

$$\nabla f(x+u) = \nabla f(x) + \nabla^2 f(x)[u] + O(\|u\|^2).$$

Ordered differently, it states that

$$\nabla^2 f(x)[u] = \nabla f(x+u) - \nabla f(x) + O(\|u\|^2).$$

Both are useful perspectives, compatible with the definition of $\nabla^2 f(x)[u]$ as being the directional derivative of ∇f at x along u.

Theorem 5.2. *Let $f : \mathcal{E} \to \mathbb{R}$ be twice continuously differentiable. If $\nabla^2 f$ is L'-Lipschitz continuous, then*

$$\forall x, u \in \mathcal{E}, \quad f(x+u) \leq f(x) + \langle \nabla f(x), u \rangle + \frac{1}{2}\langle u, \nabla^2 f(x)[u] \rangle + \frac{L'}{6}\|u\|^3.$$

Moreover,

$$\forall x, u \in \mathcal{E}, \qquad \left\| \nabla f(x+u) - \nabla f(x) - \nabla^2 f(x)[u] \right\| \leq \frac{L'}{2}\|u\|^2.$$

Proof. With $c(t) = x + tu$, consider the function $G = \nabla f \circ c : \mathbb{R} \to \mathcal{E}$. This function satisfies $G(0) = \nabla f(x)$ and $G(1) = \nabla f(x+u)$. Moreover, G is continuously differentiable. This allows us to invoke the fundamental theorem of calculus to claim:[2]

$$G(1) - G(0) = \int_0^1 G'(t)\mathrm{d}t = \int_0^1 \mathrm{D}(\nabla f)(c(t))[c'(t)]\mathrm{d}t = \int_0^1 \nabla^2 f(c(t))[c'(t)]\mathrm{d}t.$$

Plugging in our expressions for $G(0)$, $G(1)$, $c(t)$ and $c'(t) = u$, it follows that:

$$\begin{aligned}\nabla f(x+u) - \nabla f(x) &= \int_0^1 \nabla^2 f(x+tu)[u]\,\mathrm{d}t \\ &= \nabla^2 f(x)[u] + \int_0^1 \nabla^2 f(x+tu)[u] - \nabla^2 f(x)[u]\,\mathrm{d}t.\end{aligned}$$

[2] If you are uncomfortable integrating vector-valued functions, you can also pick an orthonormal basis $e_1, \ldots, e_d$ for $\mathcal{E}$ and write $G'(t) = \sum_{i=1}^d h_i(t) e_i$ with $h_i(t) = \langle G'(t), e_i \rangle$. Then, apply the reasoning to each $h_i : \mathbb{R} \to \mathbb{R}$ separately, and combine using linearity of integration.

Let us bound the norm of the integrand using the definition of operator norm and our Lipschitzness assumption:

$$\begin{aligned}\|\nabla^2 f(x+tu)[u] - \nabla^2 f(x)[u]\| &= \|(\nabla^2 f(x+tu) - \nabla^2 f(x))[u]\| \\ &\leq \|\nabla^2 f(x+tu) - \nabla^2 f(x)\|\|u\| \\ &\leq L'\|x+tu-x\|\|u\| \\ &= |t|L'\|u\|^2.\end{aligned}$$

It then follows from above that

$$\begin{aligned}\|\nabla f(x+u) - \nabla f(x) - \nabla^2 f(x)[u]\| &\leq \left\| \int_0^1 \nabla^2 f(x+tu)[u] - \nabla^2 f(x)[u] \, \mathrm{d}t \right\| \\ &\leq \int_0^1 \left\| \nabla^2 f(x+tu)[u] - \nabla^2 f(x)[u] \right\| \mathrm{d}t \\ &\leq L'\|u\|^2 \int_0^1 t \, \mathrm{d}t \\ &= \frac{L'}{2}\|u\|^2.\end{aligned}$$

This establishes the second set of inequalities in the theorem statement.

Proving the first set of inequalities is left as an exercise. □

Exercise 5.3. *Prove the first set of inequalities in Theorem 5.2.*

5.2 Local convergence

Recall Definition 3.13 about linear convergence rates. When all goes according to plan, Newton's method can convergence much faster. Convergence rates as described in the following definitions are called *superlinear*.

Definition 5.4. *Let $\theta_0, \theta_1, \theta_2, \ldots$ be a sequence converging to θ in $\mathbb{R}$. We say (θ_k) converges to θ* with at least order $q > 1$ *if there exists $r > 0$ and a sequence $\varepsilon_0, \varepsilon_1, \varepsilon_2, \ldots > 0$ such that*

$$\varepsilon_k \to 0, \qquad |\theta_k - \theta| \leq \varepsilon_k, \qquad \text{and} \qquad \lim_{k\to\infty} \frac{\varepsilon_{k+1}}{\varepsilon_k^q} = r.$$

If the inequalities hold with equality, we say convergence is with order q*; if this holds with $q = 2$, the convergence is* quadratic.

A couple of remarks are in order:

1. There is no need to require $r < 1$ since $q > 1$ (think about it.)

2. It makes no sense to discuss the rate of convergence of a sequence to θ if that sequence does not converge to θ, which is why the definition above *requires* that the sequence converges to θ as an assumption. Indeed, consider the following sequence: $\theta_k = 2$ for all k, and consider the limit $\lim_{k\to\infty} \frac{|\theta_{k+1}-0|}{|\theta_k-0|^2} = \frac{1}{2} > 0$. Of course, we *cannot* conclude from this that $\theta_0, \theta_1, \theta_2 \ldots$ converges to 0 quadratically, since it does not even converge to 0 in the first place. In fewer words: always secure convergence to θ before you discuss the rate of convergence to θ.

Exercise 5.5. *Show that if (θ_k) converges to θ with at least order $q > 1$ then in particular it converges to θ at least linearly, with arbitrarily good rate. This is why we call the above superlinear convergence.*

We now proceed to show that, under some assumptions, Newton's method converges at least quadratically (that is, with at least order two).

A3. *The Hessian of f is L'-Lipschitz continuous.*

Theorem 5.6. *Let $f\colon \mathcal{E} \to \mathbb{R}$ satisfy assumption A3, and let $x^* \in \mathcal{E}$ be a critical point of f (i.e., $\nabla f(x^*) = 0$) where $\nabla^2 f(x^*) \succ 0$—in particular, x^* is a strict local minimum. There exists $r > 0$ such that, if x_0 is in $B = \{x \in \mathcal{E} : \|x - x^*\| \leq r\}$, then Newton's method generates a sequence $(x_k)_{k\geq 0} \subset B$ which converges at least quadratically to x^*.*

Proof. We must show

1. that (x_k) remains in some non-empty ball B around x^* (to be constructed) if x_0 is in B,
2. that (x_k) then converges to x^*, and
3. that the convergence rate is then at least quadratic.

By assumption, $\nabla^2 f(x^*)$ is positive definite, that is, its eigenvalues are all (strictly) positive. To fix notation, let $\mu > 0$ be such that all eigenvalues of $\nabla^2 f(x^*)$ are at least $\frac{3}{2}\mu$. Let

$$B = \{x \in \mathcal{E} : \|x - x^*\| \leq r\} \qquad \text{with} \qquad r = \frac{\mu}{2L'}.$$

Then, for all $x \in B$ and for all $u \in \mathcal{E}$ with $\|u\| = 1$,

$$\begin{aligned}
\big\langle u, \nabla^2 f(x)[u]\big\rangle &= \big\langle u, \nabla^2 f(x^*)[u]\big\rangle + \big\langle u, (\nabla^2 f(x) - \nabla^2 f(x^*))[u]\big\rangle \\
&\geq \frac{3}{2}\mu - \|\nabla^2 f(x) - \nabla^2 f(x^*)\| \\
&\geq \frac{3}{2}\mu - L'\|x - x^*\| \\
&\geq \mu,
\end{aligned}$$

where we used the Lipschitz Hessian assumption A3 and $\|x - x^*\| \leq r = \frac{\mu}{2L'}$. In particular, for all $x \in B$ we have that $\nabla^2 f(x)$ is invertible. More precisely, we have that all of its eigenvalues are at least μ.

Assume x_k is in B for some k. Let us show that x_{k+1} is also in B. Then, it will follow by induction that the sequence remains in B. Since the goal is to control the distance between x_k and x^* as a function of k, let us start by writing the Newton iteration and subtracting x^* on both sides:

$$x_{k+1} - x^* = x_k - x^* - (\nabla^2 f(x_k))^{-1}[\nabla f(x_k)].$$

Multiply the equation by $\nabla^2 f(x_k)$:

$$\nabla^2 f(x_k)[x_{k+1} - x^*] = \nabla^2 f(x_k)[x_k - x^*] - \nabla f(x_k).$$

On the one hand, $x_k \in B$ implies that[3]

$$\|\nabla^2 f(x_k)[x_{k+1} - x^*]\| \geq \mu \|x_{k+1} - x^*\|. \tag{5.6}$$

On the other hand,

$$0 = \nabla f(x^*) = \nabla f(x_k + (x^* - x_k)) = \nabla f(x_k) + \nabla^2 f(x_k)[x^* - x_k] + v_k, \tag{5.7}$$

where $\|v_k\| \leq \frac{L'}{2}\|x_k - x^*\|^2$ owing to our Lipschitz Hessian assumption A3 and Theorem 5.2. Therefore,

$$\|\nabla^2 f(x_k)[x_{k+1} - x^*]\| = \|v_k\| \leq \frac{L'}{2}\|x_k - x^*\|^2.$$

Combining these observations, we can write:

$$\|x_{k+1} - x^*\| \leq \frac{L'}{2\mu}\|x_k - x^*\|^2. \tag{5.8}$$

Since x_k is assumed to be in B, we know $\|x_k - x^*\| \leq r$ so that

$$\begin{aligned} \|x_{k+1} - x^*\| &\leq \frac{L'}{2\mu}\|x_k - x^*\| \cdot \|x_k - x^*\| \\ &\leq \frac{L'}{2\mu}\frac{\mu}{2L'} \cdot \|x_k - x^*\| \\ &= \frac{1}{4}\|x_k - x^*\|. \end{aligned}$$

[3]This is because for positive semidefinite matrices the eigenvalues and the singular values coincide.

From this inequality, we deduce that x_{k+1} is in B. Then, under the assumption that x_0 is in B, it follows by induction that the whole sequence $x_0, x_1, x_2 \dots$ is in B and that it converges at least linearly to x^*. Moreover,[4] the convergence is at least quadratic owing to (5.8). □

Exercise 5.7. *Modify the proof of Theorem 5.6 to show that the claim still holds if $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is invertible (as opposed to requiring the stronger condition $\nabla^2 f(x^*) \succ 0$). You may find the following inequality useful:*[5] *for all linear maps A, B between Euclidean spaces, it holds that*

$$\sigma_{\min}(A + B) \geq \sigma_{\min}(A) - \sigma_{\max}(B).$$

Why is this bad news for optimization with Newton's method?

5.3 Global convergence?

In general, Newton's method does not converge, even if f is μ-strongly convex with L-Lipschitz continuous gradient and L'-Lipschitz continuous Hessian (see Exercise 5.8). However, in this generous setup, it is possible to modify Newton's method to ensure convergence.[6] The idea is to introduce a line-search procedure. We let

$$x_{k+1} = x_k - t_k \cdot \nabla^2 f(x_k)^{-1}[\nabla f(x_k)]$$

where t_k is chosen to ensure sufficient improvement in each step. If $t_k = 1$ allows us to obtain this target improvement, then we set $t_k = 1$. Otherwise, we can use a backtracking-type of line-search method to select a smaller value of t_k. Eventually, x_k will be close enough to x^*, and from that point on we will always select $t_k = 1$, so that the method will become equivalent to standard Newton. In this fashion, we can ensure global convergence and preserve the local quadratic convergence rate.

The approach described above is called *damped Newton*. However, we should keep in mind that if f is not μ-strongly convex, then this fix may not work. In particular, the Hessian of f may fail to be positive definite at

[4] Inequality (5.8) is not sufficient to claim quadratic convergence. It is *necessary* first to prove that the sequence converges, and then to claim that the convergence rate is at least quadratic. As an example, consider the sequence defined by $t_0 = 2$ and $t_{k+1} = t_k^2$: it does not converge quadratically; it diverges!

[5] See for example https://math.stackexchange.com/questions/3091423/.

[6] https://mdav.ece.gatech.edu/ece-6270-spring2021/notes/08-newtons-method.pdf

some iterate x_k, in which case the Newton direction may fail to be a descent direction, or it could even be undefined when the Hessian is not invertible.

Soon enough, we will discuss a much more reliable fix to Newton's method: the trust-region method.

Exercise 5.8. *Check that the function $f(x) = \frac{1}{10}x^2 + \sqrt{x^2+1}$ is strongly convex, that its gradient (f') is Lipschitz continuous, and yet that Newton's method on f initialized at $x_0 = 2$ does not converge to the minimizer of f. Further check that even f'' is Lipschitz continuous. Hint: check this numerically first to gain some intuition; check that f is C^2, so that the first part of the exercise reduces to studying f''.*

5.4 Solving Newton systems: conjugate gradients

Consider Algorithm 5.1 again. The Newton step at $x \in \mathcal{E}$ is the vector $u \in \mathcal{E}$ such that

$$\nabla^2 f(x)[u] = -\nabla f(x), \tag{5.9}$$

assuming $\nabla^2 f(x)$ is invertible. This is simply a linear system of equations. In principle, we could use any standard algorithm to solve it: Gaussian elimination (implemented as LU factorization), QR factorization, Cholesky factorization... Those algorithms require a matrix representation, that is, $\nabla^2 f(x)$ should be available as a matrix.

In practice however, we rarely have access to $\nabla^2 f(x)$ as a matrix. More often, we can access the Hessian as an operator, that is: we can write code which, given $v \in \mathcal{E}$, computes $\nabla^2 f(x)[v]$. In principle, we could use that code to construct a matrix representation of $\nabla^2 f(x)$ which we could then give to one of the aforementioned factorization-based algorithms, but that tends to be expensive.

Exercise 5.9. *Say you have computer code to compute $\nabla^2 f(x)[v]$, that is, you have a function (in Matlab for example) which takes as input a point x and a vector v, and outputs the vector $\nabla^2 f(x)[v]$. For the special case $\mathcal{E} = \mathbb{R}^d$, how do you use that code to compute a matrix of size $d \times d$ which corresponds to the Hessian of f at x? How many times do you need to call the above function, that is, how many different vectors v do you need to pass to that function?*

Algorithms which solve linear systems using computations of the form $\nabla^2 f(x)[v]$ but which do not require access to a matrix representation of $\nabla^2 f(x)$ are called *matrix-free* solvers.

First idea: gradient descent on the model

Fix a point $x \in \mathcal{E}$: this is our current iterate. Our original goal is to compute a vector $u \in \mathcal{E}$ such that $f(x+u)$ is small, and the idea of Newton's method is to base this decision on the second-order Taylor approximation of f around x, namely,

$$f(x+u) \approx m(u) \triangleq f(x) + \langle \nabla f(x), u \rangle + \frac{1}{2} \left\langle u, \nabla^2 f(x)[u] \right\rangle,$$

and we choose u to be the minimizer of m. To further simplify notation, let

$$H = \nabla^2 f(x), \qquad\qquad b = -\nabla f(x),$$

so that we wish to minimize

$$m(u) \triangleq \frac{1}{2} \langle u, Hu \rangle - \langle b, u \rangle .$$

(The constant $f(x)$ has no effect since x is fixed, so we discard it.)

Under the assumptions we made when designing Newton's method, the Hessian $H = \nabla^2 f(x)$ is positive definite. Thus, the quadratic model m is strongly convex: it has a unique minimizer which coincides with its critical point: $0 = \nabla m(u) = Hu - b$. This is what led us to state the formula for u as $u = -\nabla^2 f(x)^{-1}[\nabla f(x)] = H^{-1}b$: the solution of a linear system.

Let us take a few steps back: since the goal is to find the minimizer of m, why not run an optimization algorithm on m directly? For example, we could run gradient descent (GD) on m. Up to (important) details regarding step-size selection, we know that GD can behave well on strongly convex functions. Also, we are allowed to initialize anywhere we want, because there are no local optima: we will always converge to the same (optimal) u.

Accordingly, a first idea is to proceed as follows. Initialize $v_0 = 0$ (this is the initial iterate, chosen arbitrarily). We would now run GD on m starting from v_0. More generally, let us choose a direction p_0 and a step-size α_1 and let

$$v_1 = v_0 + \alpha_1 p_0.$$

The standard choice for p_0 is to let

$$p_0 = -\nabla m(v_0) = b - Hv_0,$$

but let us keep the notation more general for now. We will iterate this procedure, so, more generally the update rule shall be

$$v_n = v_{n-1} + \alpha_n p_{n-1},$$

where $p_0, p_1, p_2, \ldots$ are the (nonzero) directions we chose to use for our updates. (Again, GD would simply set $p_n = b - Hv_n$ for all n.)

What should we do for the step-sizes α_n? Generally, we would want to run a line-search algorithm such as backtracking. But here, the cost function m is a simple quadratic: we can easily select the optimal step-size! Here's how. Let α be "free" for now, and let us check the model value for all possible choices of α:

$$\begin{aligned} m(v_{n-1} + \alpha p_{n-1}) &= \frac{1}{2}\langle v_{n-1} + \alpha p_{n-1}, H(v_{n-1} + \alpha p_{n-1})\rangle - \langle b, v_{n-1} + \alpha p_{n-1}\rangle \\ &= m(v_{n-1}) + \frac{\alpha^2}{2}\langle p_{n-1}, Hp_{n-1}\rangle \\ &\quad + \frac{\alpha}{2}\langle p_{n-1}, Hv_{n-1}\rangle + \frac{\alpha}{2}\langle v_{n-1}, Hp_{n-1}\rangle - \alpha\langle b, p_{n-1}\rangle \\ &= m(v_{n-1}) + \frac{\alpha^2}{2}\langle p_{n-1}, Hp_{n-1}\rangle - \alpha\langle b - Hv_{n-1}, p_{n-1}\rangle . \end{aligned} \tag{5.10}$$

(To reach the last equality, we used the fact that H is symmetric so that $\langle u, Hv\rangle = \langle Hu, v\rangle$ for all u, v.)

The expression (5.10) is quadratic in α, and the coefficient in front of α^2 is positive because H is positive definite. Thus, that function of α is strongly convex: the optimal α is obtained by setting the derivative to zero. Explicitly, the derivative of (5.10) with respect to α is

$$\alpha\langle p_{n-1}, Hp_{n-1}\rangle - \langle b - Hv_{n-1}, p_{n-1}\rangle .$$

The best possible choice for α_n is the root of that expression, namely,

$$\alpha_n = \frac{\langle b - Hv_{n-1}, p_{n-1}\rangle}{\langle p_{n-1}, Hp_{n-1}\rangle}.$$

Notice in the numerator that $b - Hv_{n-1} = -\nabla m(v_{n-1})$. We shall find it convenient to give a name to the sequence of these (negative) gradients of the model along our sequence of iterates $v_0, v_1, v_2, \ldots$, so let

$$r_n = -\nabla m(v_n) = b - Hv_n \qquad \text{for} \qquad n = 0, 1, 2, \ldots$$

Overall, the somewhat general procedure we have described looks as follows:

- Let $v_0 = 0$, $r_0 = b - Hv_0 = b$.
- Choose some p_0; for example, GD would set $p_0 = r_0$.

- For $n = 1, 2, 3 \ldots$
  - Compute $\alpha_n = \frac{\langle r_{n-1}, p_{n-1} \rangle}{\langle p_{n-1}, H p_{n-1} \rangle}$.
  - Let $v_n = v_{n-1} + \alpha_n p_{n-1}$.
  - Let $r_n = b - H v_n$.
  - Maybe decide to stop; for example, if $\|r_n\| = \|\nabla m(v_n)\|$ is small.
  - Choose the next direction p_n; GD would set $p_n = r_n$.

This is not bad. In particular, notice that in each iteration we only need to compute two Hessian-vector products, namely, Hp_{n-1} to compute α_n and Hv_n to compute r_n. We can even better with little effort. Indeed, notice that r_n can be computed recursively as follows:

$$r_n = b - Hv_n = b - H(v_{n-1} + \alpha_n p_{n-1}) = b - Hv_{n-1} - \alpha_n Hp_{n-1} = r_{n-1} - \alpha_n Hp_{n-1}.$$

This is great because now the product Hv_n has been replaced by Hp_{n-1}, which we already had to compute anyway. So, the improved algorithm is as follows:

- Let $v_0 = 0$, $r_0 = b - Hv_0 = b$.
- Choose some p_0; for example, GD would set $p_0 = r_0$.
- For $n = 1, 2, 3 \ldots$
  - Compute Hp_{n-1} (this is the only use of H in this iteration).
  - Compute $\alpha_n = \frac{\langle r_{n-1}, p_{n-1} \rangle}{\langle p_{n-1}, H p_{n-1} \rangle}$.
  - Let $v_n = v_{n-1} + \alpha_n p_{n-1}$.
  - Let $r_n = r_{n-1} - \alpha_n H p_{n-1}$.
  - Maybe decide to stop; for example, if $\|r_n\|$ is small.
  - Choose the next direction p_n; GD would set $p_n = r_n$.

You could run this as is. But we will do better still. A *lot* better in fact, for almost the same computational cost per iteration.

Upgrade: from GD to CG

The algorithm we are about to describe is called *conjugate gradients*. To see how you could have discovered it yourself, reconsider the GD algorithm

above, and observe what happens if we expand the iterates v_n simply by following what the algorithm does:

$$\begin{aligned} v_0 &= 0, \\ v_1 &= v_0 + \alpha_1 p_0 = \alpha_1 p_0, \\ v_2 &= v_1 + \alpha_2 p_1 = \alpha_1 p_0 + \alpha_2 p_1, \\ v_3 &= v_2 + \alpha_3 p_2 = \alpha_1 p_0 + \alpha_2 p_1 + \alpha_3 p_2, \ldots \end{aligned}$$

Clearly, each v_n is a linear combination of the directions $p_0, \ldots, p_{n-1}$ we chose to use. Moreover, the coefficients $\alpha_1, \alpha_2, \ldots$ were chosen one by one to be optimal along one specific direction.

Here is an idea: at each iteration, perhaps we could reconsider *all* the "step-sizes", that is, all the coefficients used to form v_n as a linear combination of all the directions we have generated so far?

Here is how this would go. At iteration n, we let

$$v_n = \alpha_{n,1} p_0 + \ldots + \alpha_{n,n} p_{n-1}, \tag{5.11}$$

for some coefficients $\alpha_{n,1}, \ldots, \alpha_{n,n}$ to be determined. (They have two indices now, because the coefficient of, say, p_0 to form v_0 may not be the same as the coefficient of p_0 to form v_1, etc.: let's keep our options open for now.) The value of the model at this v_n is as follows:

$$\begin{aligned} m(v_n) &= m\left(\sum_{i=1}^{n} \alpha_{n,i} p_{i-1}\right) \\ &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{n,i} \alpha_{n,j} \left\langle p_{i-1}, H p_{j-1} \right\rangle - \sum_{k=1}^{n} \alpha_k \left\langle b, p_{k-1} \right\rangle \\ &= \frac{1}{2} a^\top M a - c^\top a, \end{aligned}$$

where $a \in \mathbb{R}^n$, $M \in \mathbb{R}^{n \times n}$ and $c \in \mathbb{R}^n$ are defined by

$$a_i = \alpha_{n,i}, \qquad M_{ij} = \left\langle p_{i-1}, H p_{j-1} \right\rangle, \qquad c_k = \left\langle b, p_{k-1} \right\rangle.$$

This is a quadratic in the vector of coefficients a that we need to select. Notice the following linear algebra fact:

> *As long as the vectors $p_0, \ldots, p_{n-1}$ are linearly independent, and since we assume H is positive definite, the matrix $M \in \mathbb{R}^{n \times n}$ is positive definite too.*

(To verify this, go back to the calculation above and check that $a^\top Ma \geq 0$ for all a, and that this is zero if and only if $a = 0$.) This means that the quadratic $a \mapsto \frac{1}{2}a^\top Ma - c^\top a$ is strongly convex, so we can find the optimal a simply by setting the gradient to zero. The gradient of the quadratic is $Ma - c$ (make sure you know how to find this), and hence the optimal choice of coefficients is obtained by solving the linear system

$$Ma = c.$$

This is certainly doable, but there are a few caveats:

1. The linear system at iteration n is of size $n \times n$; so the more we iterate, the more expensive the iterations become.

2. Also, in order to form v_n, we need to "remember" (store in memory) all the vectors p_0, p_1, p_2 etc. generated so far, and to compute their linear combination from scratch each time: this means a lot of memory use and a lot of computations if the dimension of $\mathcal{E}$ is large.

Fortunately, we can resolve all of that with a clever choice of directions p_i.

The initial intuition is that we might aim to make the linear system $Ma = c$ easy to solve. Specifically, let us aim for M to be diagonal. This is the case exactly if

$$M_{ij} = \langle p_{i-i}, Hp_{j-i} \rangle = 0 \qquad \text{for all} \qquad i \neq j.$$

In words: M is diagonal if the directions $p_0, p_1, p_2, \ldots$ are *H-orthogonal*, that is, *orthogonal with respect to the inner product*

$$\langle u, v \rangle_H \triangleq \langle u, Hv \rangle .$$

This is indeed an inner product because H is positive definite (make sure this makes sense to you).

In the gradient descent version of the algorithm, we simply chose the vectors $r_0, r_1, r_2, \ldots$ to be the optimization directions. There is no reason a priori that these should just happen to be H-orthogonal. However, we can modify them in order to produce new directions that are H-orthogonal: this is exactly what the Gram–Schmidt algorithm is for.

Concretely, the plan is as follows: let

$$p_0 = r_0,$$

and then, for each $n > 0$, let p_n be the part of r_n that is H-orthogonal to the subspace spanned by all previous vectors generated so far:

$$p_n = r_n - \sum_{i=0}^{n-1} \frac{\langle r_n, Hp_i \rangle}{\langle p_i, Hp_i \rangle} p_i.$$

Here is a first remarkable fact:

> *It so happens that all the coefficients* $\frac{\langle r_n, Hp_i\rangle}{\langle p_i, Hp_i\rangle}$ *are zero, except the one for* p_{n-1}.

We skip the proof (it is not entirely trivial). Thus, the above expression for p_n simplifies to

$$p_n = r_n - \frac{\langle r_n, Hp_{n-1}\rangle}{\langle p_{n-1}, Hp_{n-1}\rangle} p_{n-1}.$$

This is particularly nice, as it means that computing p_n only requires us to "remember" p_{n-1} (the earlier vectors in that sequence can be forgotten). Also, this computation is rather cheap compared to a linear combination of n vectors.

Here is another particularly convenient fact. Now that M is diagonal, the linear system $Ma = c$ has a simple, explicit solution:

$$\alpha_{n,i} = a_i = \frac{c_i}{M_{ii}} = \frac{\langle b, p_{i-1}\rangle}{\langle p_{i-1}, Hp_{i-1}\rangle}.$$

Notice how this expression actually does *not* depend on n. Thus, we can legitimately rename them α_i:

$$\alpha_i = \frac{\langle b, p_{i-1}\rangle}{\langle p_{i-1}, Hp_{i-1}\rangle}.$$

Going back to (5.11), we have

$$v_n = \alpha_1 p_0 + \ldots + \alpha_n p_{n-1}.$$

For the same reason, the previous iterate is

$$v_{n-1} = \alpha_1 p_0 + \ldots + \alpha_{n-1} p_{n-2}.$$

Thus, v_n can be obtained with the following recursion:

$$v_n = v_{n-1} + \alpha_n p_{n-1}.$$

Here too, we see that the computation is cheap, and it only requires us to have access to p_{n-1}: the older vectors can safely be forgotten.

Overall, starting from the GD algorithm above and making the necessary changes, the algorithm looks like this:

- Let $v_0 = 0$, $r_0 = b$.

- Let $p_0 = r_0$.
- For $n = 1, 2, 3 \ldots$
  - Compute Hp_{n-1} (this is the only use of H in this iteration).
  - Compute $\alpha_n = \frac{\langle b, p_{n-1} \rangle}{\langle p_{n-1}, Hp_{n-1} \rangle}$.
  - Let $v_n = v_{n-1} + \alpha_n p_{n-1}$.
  - Let $r_n = r_{n-1} - \alpha_n Hp_{n-1}$.
  - Maybe decide to stop; for example, if $\|r_n\|$ is small.
  - Compute $\beta_n = -\frac{\langle r_n, Hp_{n-1} \rangle}{\langle p_{n-1}, Hp_{n-1} \rangle}$.
  - Let $p_n = r_n + \beta_n p_{n-1}$.

That's it. The only other technical thing is that the coefficients α_n and β_n are usually computed with different formulas, as follows:

$$\alpha_n = \frac{\|r_{n-1}\|^2}{\langle p_{n-1}, Hp_{n-1} \rangle}, \qquad \beta_n = \frac{\|r_n\|^2}{\|r_{n-1}\|^2}.$$

It is not obvious why these formulas are equivalent to the ones above, but they are the preferred ones for numerical reasons. The final algorithm is known as the conjugate gradients method (CG): see Algorithm 5.2.

By construction, it is clear that

- GD and CG have the same first iterate v_1.
- For all subsequent iterates, CG can only do better than GD (and typically does so by a lot).
- Yet, the computational cost of one iteration of CG is essentially the same as that of GD.

There is a lot more to say about CG, but we won't have enough time in this course. The next section is there if you are interested—feel free to skip it.

Additional notes about CG (optional)

CG is an iterative algorithm: it produces a sequence $v_0, v_1, v_2, \ldots$ in $\mathcal{E}$ whose iterates are increasingly better approximations of the solution u to the linear system $Hu = b$. Here, "better" is measured with respect to a special norm.

Indeed, instead of measuring the distance from v_n to u in the usual Euclidean norm, $\|v\| = \sqrt{\langle v, v \rangle}$, CG tries to reduce the approximation error in the H-norm, defined as:

$$\|v\|_H = \sqrt{\langle v, Hv \rangle}. \tag{5.12}$$

Exercise 5.10. *Verify that* $\|\cdot\|_H$ *is a norm, using* $H \succ 0$.

The (squared) approximation error for a vector $v \in \mathcal{E}$ then obeys:

$$\begin{aligned}\|v-u\|_H^2 &= \langle v-u, H(v-u)\rangle \\ &= \langle v, Hv\rangle - \langle u, Hv\rangle - \langle v, Hu\rangle + \langle u, Hu\rangle \\ &= \langle v, Hv\rangle - 2\langle v, b\rangle + \langle u, Hu\rangle ,\end{aligned}$$

where we used that H is symmetric and $Hu = b$. Define $g\colon \mathcal{E} \to \mathbb{R}$ as

$$g(v) = \frac{1}{2}\langle v, Hv\rangle - \langle v, b\rangle . \tag{5.13}$$

Then, we have found that

$$\|v-u\|_H^2 = 2g(v) + \langle u, Hu\rangle . \tag{5.14}$$

Since the last term on the right-hand side is independent of v, we conclude that minimizing $\|v-u\|_H$ is *equivalent* to minimizing $g(v)$.

CG (Algorithm 5.2) rests on this last observation: it is an optimization algorithm specifically designed to minimize $g(v)$. In so doing, it actually minimizes $\|v-u\|_H$, but notice an important fact: if you were asked to minimize $h(v) = \|v-u\|_H$, it would not be clear how to do it, because u is the object we are trying to find: we do not have access to it yet, so how would we compute h, its gradient etc. which we would likely need to minimize h? In contrast, we know how to compute g and its gradient: this is why it's possible to minimize it.

Here is a little bit more context for how CG works. The claims below are not obvious: they should help you understand how CG works, but we omit proofs: see [NW06, §5.1]. As you can see from Algorithm 5.2, CG iteratively generates a sequence of vectors

$$p_0, p_1, p_2, \ldots \in \mathcal{E}.$$

Under our assumption that H is positive definite, one can show that these vectors are linearly independent.[7] Moreover, they are orthogonal with respect to the special inner product $\langle u, v\rangle_H \triangleq \langle u, Hv\rangle$, that is,

$$\forall i \neq j, \qquad \langle p_i, p_j\rangle_H = 0. \tag{5.15}$$

[7] To be a bit more precise: it can be shown that the algorithm necessarily stops after a finite number of iterations; for example, it stops at iteration n, having generated $p_0, \ldots, p_{n-1}$. Then, the claim is that $p_0, \ldots, p_{n-1}$ are linearly independent.

We say that the vectors $p_0, p_1, p_2 \ldots$ are *H-conjugate directions.*

Say we are at iteration n in CG. By linear independence, the vectors $p_0, \ldots, p_{n-1}$ span a linear subspace of dimension n in $\mathcal{E}$. The algorithm is designed in such a way that v_n has the following property:

$$v_n = \operatorname*{arg\,min}_{v \in \operatorname{span}(p_0,\ldots,p_{n-1})} g(v), \tag{5.16}$$

that is, v_n is the (unique) global minimizer of g *restricted* to the subspace spanned by the available directions $p_0, \ldots, p_{n-1}$.

It is then clear that, by the time we get to iteration $n = \dim \mathcal{E}$, the directions $p_0, \ldots, p_{n-1}$ must form a basis for the whole space $\mathcal{E}$, and therefore v_n is a global minimizer of g over the whole space $\mathcal{E}$, that is: v_n is the solution to our problem, and the algorithm necessarily stops.

One can show a rather more sophisticated guarantee using Chebyshev polynomials:

Algorithm 5.2 Conjugate gradients

Input: positive definite linear map H on $\mathcal{E}$ and $b \in \mathcal{E}$

Set $v_0 = 0, r_0 = b, p_0 = r_0$

For $n = 1, 2, \ldots$

Compute Hp_{n-1} (this is the only call to H)

$\alpha_n = \frac{\|r_{n-1}\|^2}{\langle p_{n-1}, Hp_{n-1}\rangle}$

$v_n = v_{n-1} + \alpha_n p_{n-1}$

$r_n = r_{n-1} - \alpha_n H p_{n-1}$

If $r_n = 0$, **output** $u = v_n$: the solution of $Hu = b$

$\beta_n = \frac{\|r_n\|^2}{\|r_{n-1}\|^2}$

$p_n = r_n + \beta_n p_{n-1}$

Theorem 5.11. *CG terminates in n iterations if H has only n distinct eigenvalues. More generally, if $\lambda_{\min}$ and $\lambda_{\max}$ denote the smallest and largest eigenvalues of H, then $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$ is the* condition number *of H, and*

$$\|v_n - u\|_H \leq \|u\|_H \cdot 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^n, \tag{5.17}$$

so that the error decreases exponentially fast as CG iterates (we have at least linear convergence), and it eventually hits zero after at most $\dim \mathcal{E}$ *iterations.*

The fact that CG terminates in at most $\dim \mathcal{E}$ iterations is of little practical relevance, in part because numerical round-off errors typically preempt this. However, the progressive improvement of the iterates v_n as predicted by (5.17) is borne out empirically, and the role of the condition number κ is indeed critical.

In practice, CG is terminated after a set number of iterations, or when a target relative tolerance is met. For example, it is standard to replace the stopping criterion $r_n = 0$ with

$$\|r_n\| \leq \varepsilon_{\text{tolerance}} \|b\|. \tag{5.18}$$

This makes sense because

$$r_n = b - Hv_n = -\nabla g(v_n) \tag{5.19}$$

that is, r_n is the *residue* of the linear system at iteration n, and it is also the (negative) gradient of g at v_n, both of which are zero when v_n is the solution. Moreover, $b = r_0$ is the initial residue, so that it is natural to use it as the "scale" against which to judge whether r_n has become sufficiently small.

Remark 5.12. *A simple alternative to CG is to run gradient descent* with fixed step-size *on* $g(v)$ (5.13). *Since* g *is a convex quadratic, it is easy to check that it has* L*-Lipschitz continuous gradient with* $L = \lambda_{\max}(H)$. *Using previous facts in these lectures notes, one can show that running* $v_{n+1} = v_n - \frac{1}{L}\nabla g(v_n)$ *with* $v_0 = 0$ *leads to* $\|v_n - u\| \leq e^{-n/\kappa}\|u\|$ *where* $\kappa = \frac{\lambda_{\max}(H)}{\lambda_{\min}(H)}$. *In contrast, CG guarantees* $\|v_n - u\|_H \leq 2e^{-n/\sqrt{\kappa}}\|u\|_H$. *The square root is a big improvement.*

Chapter 6

Trust-region methods

Gradient descent methods enjoy global convergence under reasonable assumptions. This is highly desirable, but on the flip side they can be fairly slow overall. In particular, if convergence is to a critical point where the Hessian is positive definite but has poor conditioning, then even the linear convergence phase may be slow. Newton's method essentially solves the slow linear convergence issue, but it suffers from a much graver issue: it does not have any reasonable global behavior.

Trust-region methods (TR) form a family of algorithms which have the ability to combine the best of both worlds. TR is sometimes called a *globalization* of Newton's method, because (in its main incarnation) the purpose of TR is to wrap Newton's method with safe-guards so that it enjoys global convergence properties akin to those of GD, while preserving the quadratic local convergence rate of Newton's.

What makes Newton's method fail? Essentially, it is its blind trust in Taylor expansions of f. Indeed, Newton's method sets x_{k+1} to be the critical point of the second-order Taylor expansion of f around x_k with the *hope* (a) that this critical point is a minimizer of the quadratic, and (b) that it is a good idea to move to that minimizer. However, part (a) fails if the Hessian of f at x_k is not positive definite, and part (b) can fail even when $\nabla^2 f(x_k)$ is positive definite. The latter is because the minimizer of the second-order Taylor approximation of f at x_k may be far from x_k, yet the Taylor approximation is a good approximation to f only locally.

TR methods fix the above issues as follows. They still use quadratic models to approximate f locally (and these models may or may not be second-order Taylor expansions of f), but they are careful to *trust* those models only locally. The *region* in which TR trusts the model is a ball whose radius is adapted automatically and dynamically. Moreover, rather than blindly jumping to critical points of quadratics in the hope that the critical points

are minimizers, TR methods actively aim (at least approximately) to minimize the quadratics in the trust regions. This last point makes it possible to accommodate non-convex quadratic models.

A typical iteration of TR entails the following steps:

1. Approximately minimize a quadratic model of f around x_k inside the current trust region, producing a tentative next point x_k^+.

2. Decide whether to accept the tentative step ($x_{k+1} = x_k^+$) or to reject it ($x_{k+1} = x_k$).

3. Regarding the radius of the trust region: decide whether to increase it, decrease it or leave it unchanged.

The first step requires what is called a *subproblem solver*. We have a lot of leeway in choosing how we want to proceed here. It is important to remember the big picture: the goal is to minimize f; solving the subproblem is only a means to an end. Therefore, we should try not to invest too much computational effort in it unless it is promising. Many TR methods are designed to carefully transition from behaving mostly like GD (when we are far from a minimizer and it makes no sense to work too hard minimizing the model quadratics) to behaving mostly like Newton's method (when we are close to a minimizer and we have reasonable hope to trigger a quadratic local convergence rate by putting in more effort.)

The bible of TR methods is an almost 1000-page book by Conn, Gould and Toint [CGT00]. In these notes, we discuss just a couple of variations that offer a mix of insight and practicality. Section 6.3 is based on [CGT00, §7.5].

6.1 The trust-region mechanism

As always, our goal is to minimize a (differentiable) cost function $f : \mathcal{E} \to \mathbb{R}$. To this end, TR methods produce two sequences: a sequence of iterates $x_0, x_1, x_2, \ldots \in \mathcal{E}$ and a sequence of trust-region radii $\Delta_0, \Delta_1, \Delta_2, \ldots > 0$.

At the iterate x_k, we define the quadratic model

$$m_k(v) = f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2} \langle v, H_k(v) \rangle \tag{6.1}$$

where $H_k : \mathcal{E} \to \mathcal{E}$ is a symmetric linear map for us to choose. For example, we may want to set $H_k = \nabla^2 f(x_k)$ as then m_k is the second-order Taylor

expansion of f around x_k. We pick the tentative next iterate x_k^+ as $x_k + u_k$ such that the step u_k *approximately* solves the *trust-region subproblem*:

$$\min_{v \in \mathcal{E}} m_k(v) \quad \text{subject to} \quad \|v\| \leq \Delta_k, \tag{6.2}$$

where Δ_k is the radius of the trust region at iteration k. What does "approximately" mean? We will be specific later, but at the very least $m_k(u_k)$ should be smaller than $m_k(0)$, that is: at the very least, *the model* should judge that the step we chose is better than not moving at all. Ideally, we will do a whole lot better than that though.

The step is accepted ($x_{k+1} = x_k^+$) or rejected ($x_{k+1} = x_k$) based on the performance of x_k^+ *as judged by the actual cost function* f, compared to the expected improvement *as predicted by the model*. This is measured using the ratio ρ_k (6.3).

Depending on how the two compare, the trust-region radius may also be adapted. In particular, if the step was rejected, then we certainly want to reduce the trust-region radius for the next iterate (what might happen if we don't?) Even if the step was accepted, we may want to reduce the trust-region radius if we find that the model prediction was substantially off compared to the actual cost decrease. If agreement between the model and the cost is excellent and our selected step u_k hits the boundary of the trust region (that is, $\|u_k\| = \Delta_k$), then we may decide to increase the trust-region radius (though never beyond a fixed upper-bound $\bar{\Delta}$ set ahead of time.) In all other situations, we just keep the radius unchanged. See Algorithm 6.1.

Remark 6.1. *The parameter $\bar{\Delta}$ and the initial radius Δ_0 must be set by the user, and they may depend on the application. Generic values I like to use for no particularly good reason other than "they often work" are: $\bar{\Delta} = \sqrt{\dim \mathcal{E}}$ and $\Delta_0 = \bar{\Delta}/8$.*

Remark 6.2. *As innocuous as it may seem, computing ρ_k (6.3) can be tricky numerically when we are close to convergence. Indeed, the vector u_k is then often very small, so that x_k and x_k^+ are almost identical. As a result, $f(x_k)$ and $f(x_k^+)$ are also almost identical. This is problematic numerically because to compute ρ_k we have to compute $f(x_k) - f(x_k^+)$: a difference of two numbers which, individually, might be large, but whose difference is small. A common heuristic to reduce such issues is to add*

$$\max(1, |f(x_k)|) \cdot 10^{-13}$$

both to the numerator and to the denominator when computing ρ_k, where 10^{-13} is a parameter: a value somewhat larger than the machine precision.

Exercise 6.3. *How do you compute the denominator of ρ_k, namely, $m_k(0) - m_k(u_k)$, to avoid the issues laid out above for $f(x_k) - f(x_k^+)$?*

Algorithm 6.1 TR: trust-region method

Parameters: maximum radius $\bar{\Delta} > 0$, threshold $\rho' \in (0, 1/4)$

Input: $x_0 \in \mathcal{E}$, $\Delta_0 \in (0, \bar{\Delta}]$

For $k = 0, 1, 2, \ldots$

Pick a symmetric linear map $H_k \colon \mathcal{E} \to \mathcal{E}$ to define m_k (6.1).

Approximately solve the subproblem (6.2), yielding u_k.

The tentative next iterate is $x_k^+ = x_k + u_k$.

Compute the ratio of actual to model improvement:

$$\rho_k = \frac{f(x_k) - f(x_k^+)}{m_k(0) - m_k(u_k)}. \tag{6.3}$$

Accept or reject the tentative next iterate:

$$x_{k+1} = \begin{cases} x_k^+ & \text{if } \rho_k > \rho' \text{ (accept)}, \\ x_k & \text{otherwise (reject)}. \end{cases} \tag{6.4}$$

Update the trust-region radius:

$$\Delta_{k+1} = \begin{cases} \frac{1}{4}\Delta_k & \text{if } \rho_k < \frac{1}{4}, \\ \min(2\Delta_k, \bar{\Delta}) & \text{if } \rho_k > \frac{3}{4} \text{ and } \|u_k\| = \Delta_k, \\ \Delta_k & \text{otherwise.} \end{cases} \tag{6.5}$$

6.2 The least we can do for global convergence

In this section,[1] we determine the minimal effort we should put into choosing the models (6.1) (that is, into choosing H_k) and into approximately solving the subproblems (6.2) in order to ensure TR enjoys the same kind of "global convergence" guarantees as GD, namely: if we run $O(1/\varepsilon^2)$ iterations, we can expect to find a point where the gradient is smaller than ε.

As we shall see, with minimalist effort TR does not do better than GD—it also does not require more efforts to run. The gains lie in the flexibility of the TR framework: we have the liberty to put in much more effort at each iteration than what is described below, and that can lead to substantial improvements. Specifically, it is possible to obtain superlinear convergence. The cherry on top is that this can all be achieved with rather limited parameter tuning.

At a point x, we consider a quadratic model for $f(x+v) \approx m(v)$,

$$m(v) = f(x) + \langle \nabla f(x), v \rangle + \frac{1}{2} \langle v, Hv \rangle , \tag{6.6}$$

where H is some symmetric linear map on $\mathcal{E}$ of our choice. The subproblem consists in minimizing $m(v)$ in the ball of radius $\Delta > 0$ around x, called the trust region. Of course, the vector $v = 0$ belongs to the trust region; it attains the model value $m(0) = f(x)$. In the following lemma, we consider a more interesting (yet trivially computable) point in the trust region which attains a lower model value.

Definition 6.4. *The* Cauchy step *for the model $m(v)$ with respect to the radius Δ is the vector u^C defined by*

$$u^C = -t^C \cdot \nabla f(x) \qquad \text{with} \qquad t^C \in \operatorname*{arg\,min}_{0 \leq t \leq \frac{\Delta}{\|\nabla f(x)\|}} m(-t \cdot \nabla f(x)). \tag{6.7}$$

In words: u^C is the minimizer of $m(v)$ in the trust region restricted to the negative gradient direction.

Lemma 6.5. *Write $g = \nabla f(x)$ for short. The Cauchy step $u^C = -t^C \cdot g$ can be computed with*

$$t^C = \begin{cases} \min\left(\frac{\|g\|^2}{\langle g, Hg \rangle}, \frac{\Delta}{\|g\|}\right) & \text{if } \langle g, Hg \rangle > 0, \\ \frac{\Delta}{\|g\|} & \text{otherwise.} \end{cases} \tag{6.8}$$

[1] In spirit, the analysis of TR presented here takes after the one in [CGT12]. In that reference, the authors also show how to ensure TR finds approximate second-order critical points in a bounded number of iterations.

As a result, the Cauchy step leads to the following decrease in model value:

$$m(0) - m(u^C) \geq \frac{1}{2} \min\left(\Delta, \frac{\|g\|}{\|H\|}\right) \|g\|, \tag{6.9}$$

where as usual $\|H\|$ is the operator norm of H with respect to the Euclidean norm on $\mathcal{E}$.

Exercise 6.6. *Give a proof of Lemma 6.5. Hint: notice that t^C is the minimizer of a 1-D quadratic restricted to a closed interval.*

Recall our analysis of GD with constant step-size in Theorem 3.4. It was based on showing that the value of f decreases by some amount at each iteration; that amount was dictated by the current gradient norm. We plan to use a similar strategy here.

Let u_k^C be the Cauchy step for the model m_k (6.1) at iterate x_k with trust-region radius Δ_k. Assume the step u_k is chosen such that it does at least as well as the Cauchy step:

A4. *For all k, the trial step u_k is at least as good as the Cauchy step, that is, $\|u_k\| \leq \Delta_k$ and:*

$$m_k(u_k) \leq m_k(u_k^C). \tag{6.10}$$

Assumption A4 leaves us with a lot of freedom. For example, we could attempt to compute a true minimizer of m_k in the trust region: that would certainly satisfy the above condition. As another example, we could compute both the Newton step and the Cauchy step: if the Newton step is inside of the trust region and it outperforms the Cauchy step, then we select the Newton step as our u_k; otherwise we select the Cauchy step as a fall back. Either way, it follows from (6.10) and (6.9) that

$$m_k(0) - m_k(u_k) \geq m_k(0) - m_k(u_k^C) \geq \frac{1}{2} \min\left(\Delta_k, \frac{\|\nabla f(x_k)\|}{\|H_k\|}\right) \|\nabla f(x_k)\|. \tag{6.11}$$

As prescribed in (6.4), the step is accepted if $\rho_k > \rho'$. In that case, $x_{k+1} = x_k^+$ and equation (6.3) for ρ_k tells us that

$$\begin{aligned} f(x_k) - f(x_{k+1}) &= \rho_k \cdot (m_k(0) - m_k(u_k)) \\ &\geq \frac{\rho'}{2} \min\left(\Delta_k, \frac{\|\nabla f(x_k)\|}{\|H_k\|}\right) \|\nabla f(x_k)\|. \end{aligned} \tag{6.12}$$

We learn a lot from (6.12). Indeed, that inequality tells us that *if* $\|\nabla f(x_k)\|$ *is still large* and

1. $\|H_k\|$ is not too large,
2. Δ_k is not too small, and
3. Iteration k is successful (that is, the step is accepted),

then the value of the cost function decreases substantially from x_k to x_{k+1}. (Compare the amount of decrease in (6.12) to that we obtained for a regular gradient step in (3.3): they are similar.)

From the list above, we understand that we need to ensure that

1. $\|H_k\|$ remains bounded above,
2. Δ_k remains bounded away from zero, and
3. A large number of iterations are successful.

For the first item, we must introduce an assumption (indeed, so far we have given the user complete freedom in choosing the models).

A5. *There exists a constant $L \geq 0$ such that $\|H_k\| \leq L$ for all k.*

Remark 6.7. *The following are two simple ways to ensure A5 holds with constant L:*

1. *Set $H_k = L \cdot I$, where I is the identity map on $\mathcal{E}$ (e.g., for $\mathcal{E} = \mathbb{R}^n$, I is the identity matrix of size n). (What is the Cauchy step then?)*
2. *If f is twice differentiable and its gradient is L-Lipschitz continuous, set $H_k = \nabla^2 f(x_k)$. The claim then follows from Theorem 3.3.*

For the other items on the list above, we have the following two lemmas. We require the same assumptions as for GD, namely, A1 (f is lower-bounded by f_{low}) and A2 (∇f is L-Lipschitz continuous). The first lemma below states that Δ_k cannot become arbitrarily small. In a nutshell, this happens because m_k is at least a first-order approximation of f: when Δ_k is very small, m_k does a good job approximating f, and therefore ρ_k is close to 1. In particular, $\rho_k \geq 1/4$ and so the trust-region radius is not reduced further: Δ_{k+1} is not smaller than Δ_k.

Lemma 6.8. *Let assumptions A2 and A5 hold with the same constant L.*[2] *Further require assumption A4. If $\|\nabla f(x_k)\| > \varepsilon$ for all $k \in \{0, \ldots, K-1\}$, then*

$$\forall k \in \{0, \ldots, K-1\}, \qquad \Delta_k \geq \min\left(\Delta_0, \frac{\varepsilon}{16L}\right). \tag{6.13}$$

[2] If they hold separately but with two distinct constants L_1 and L_2, we can always set $L = \max(L_1, L_2)$ so that both assumptions hold with L.

Proof. The plan is to show that if Δ_k is small, then ρ_k is large. Because of how Algorithm 6.1 works, a large ρ_k implies that $\Delta_{k+1} \geq \Delta_k$. By definition of ρ_k (6.3), and then using $m_k(0) = f(x_k)$ and $x_k^+ = x_k + u_k$, we have

$$1 - \rho_k = 1 - \frac{f(x_k) - f(x_k^+)}{m_k(0) - m_k(u_k)} = \frac{f(x_k + u_k) - m_k(u_k)}{m_k(0) - m_k(u_k)}. \tag{6.14}$$

The numerator of (6.14) is upper-bounded because m_k is at least a first-order Taylor approximation of f around x_k:

$$\begin{aligned} f(x_k + u_k) - m_k(u_k) &= \underbrace{f(x_k + u_k) - f(x_k) - \langle \nabla f(x_k), u_k \rangle}_{\text{A2+Theorem 3.2}} - \frac{1}{2} \langle u_k, H_k(u_k) \rangle \\ &\leq \frac{1}{2}(L + \underbrace{\|H_k\|}_{\text{A5}}) \|u_k\|^2 \\ &\leq L \|u_k\|^2. \end{aligned}$$

The denominator of (6.14) is lower-bounded owing to our assumption A4 that u_k does at least as well as the Cauchy step. Indeed, by (6.11) and plugging in $\|H_k\| \leq L$ as above, we have:

$$m_k(0) - m_k(u_k) \geq \frac{1}{2} \min\left(\Delta_k, \frac{\|\nabla f(x_k)\|}{L} \right) \|\nabla f(x_k)\|.$$

We combine the bounds on the numerator and denominator of (6.14) to find (also plugging in the facts that $\|u_k\| \leq \Delta_k$ and $\|\nabla f(x_k)\| > \varepsilon$):

$$1 - \rho_k \leq \frac{2L\Delta_k^2}{\varepsilon} \frac{1}{\min\left(\Delta_k, \frac{\varepsilon}{L}\right)}. \tag{6.15}$$

Recall that we want to show that if Δ_k is small then ρ_k is large. To this end, consider the case where

$$\Delta_k \leq \frac{\varepsilon}{4L}. \tag{6.16}$$

(Otherwise, eq. (6.13) already holds.) Plugging this into the above inequality twice in a row (first to resolve the min, second to bound the remaining Δ_k) we deduce:

$$1 - \rho_k \leq \frac{2L\Delta_k}{\varepsilon} \leq \frac{1}{2}. \tag{6.17}$$

Therefore, $\rho_k \geq \frac{1}{2}$. The mechanism (6.5) in Algorithm 6.1 then implies that $\Delta_{k+1} \geq \Delta_k$. It now takes a simple proof by induction to conclude. (Specifically: in light of mechanism (6.5), consider why we must have Δ_0 in the bound (6.13), and why we must have a further factor $1/4$ in there as compared to (6.16).) □

Say we let TR run K iterations, visiting points $x_0, \ldots, x_{K-1}$. It is useful to partition $\{0, \ldots, K-1\}$ into successful and unsuccessful iterations according to (6.4), as follows:

$$S_K = \{k \in \{0, \ldots, K-1\} : \rho_k > \rho'\},$$
$$U_K = \{k \in \{0, \ldots, K-1\} : \rho_k \leq \rho'\}.$$

The following theorem states that if TR enjoys a large number of successful iterations then at least one of the iterates is a point where the gradient is small.

Theorem 6.9. *Let assumptions A2 and A5 hold with the same constant L. Further require assumptions A4 and A1 (the latter states $f(x) \geq f_{\text{low}}$ for all x). For any $0 < \varepsilon \leq 16L\Delta_0$, if the number $|S_K|$ of successful iterations among the first K iterations of Algorithm 6.1 satisfies*

$$|S_K| > \frac{32L(f(x_0) - f_{\text{low}})}{\rho'} \frac{1}{\varepsilon^2},$$

then there exists $k < K$ such that $\|\nabla f(x_k)\| \leq \varepsilon$.

Proof. Assume for contradiction that $\|\nabla f(x_k)\| > \varepsilon$ for all $k \in \{0, \ldots, K-1\}$. Then, combine Lemma 6.8 with eq. (6.12), $\|H_k\| \leq L$ and $\varepsilon \leq 16L\Delta_0$ to find that if k is a successful iteration then

$$f(x_k) - f(x_{k+1}) \geq \frac{\rho'}{2} \min\left(\Delta_0, \frac{\varepsilon}{16L}\right) \varepsilon = \frac{\rho'}{32L}\varepsilon^2. \tag{6.18}$$

On the other hand, if k is an unsuccessful iteration then $x_{k+1} = x_k$ so that $f(x_k) - f(x_{k+1}) = 0$. Therefore, using the same telescoping sum argument as when we studied GD but this time being careful to split the sum over successful and unsuccessful iterations, we find:

$$\begin{aligned} f(x_0) - f_{\text{low}} &\geq f(x_0) - f(x_K) \\ &= \sum_{k=0}^{K-1} f(x_k) - f(x_{k+1}) \\ &= \sum_{k \in S_K} f(x_k) - f(x_{k+1}) \\ &\geq \sum_{k \in S_K} \frac{\rho'}{32L}\varepsilon^2 \\ &= |S_K| \frac{\rho'}{32L}\varepsilon^2. \end{aligned}$$

Rearrange to conclude. □

Theorem 6.9 leaves one rather important question open: can we be certain that Algorithm 6.1 will enjoy many successful steps? Could it not happen that the algorithm will experience an overwhelming number of unsuccessful steps, in which case that theorem would be meaningless? Fortunately, that cannot happen. In a nutshell, the reason is that every time we suffer an unsuccessful step, the trust-region radius is decreased. Yet, we know from Lemma 6.8 that the trust-region radius cannot become arbitrarily small unless we encounter points with small gradient. Thus, each radius reduction by a factor 1/4 (which may or may not be unsuccessful) must be compensated by at least two radius increases by a factor 2 (which must be successful). Overall, we find that at least two thirds of iterations must be successful (asymptotically).

Lemma 6.10. *Let assumptions A2 and A5 hold with the same constant L. Further require assumption A4. Let $0 < \varepsilon \leq 16L\Delta_0$ be such that $\|\nabla f(x_k)\| > \varepsilon$ for all $k \in \{0, \ldots, K-1\}$. Then*

$$|S_K| \geq \frac{2}{3}K - \frac{1}{3}\log_2\left(\frac{16L\Delta_0}{\varepsilon}\right).$$

Proof. Consider the trust-region radius update mechanism (6.5). We have $\Delta_{k+1} \leq 2\Delta_k$ if k is a successful step and $\Delta_{k+1} \leq \frac{1}{4}\Delta_k$ if k is unsuccessful. By induction, we deduce that (to reach the last step use $|S_K| + |U_K| = K$):

$$\Delta_K \leq 2^{|S_K|}\frac{1}{4^{|U_K|}}\Delta_0 = 2^{|S_K|-2|U_K|}\Delta_0 = 2^{3|S_K|-2K}\Delta_0.$$

On the other hand, by Lemma 6.8:

$$\Delta_K \geq \min\left(\Delta_0, \frac{\varepsilon}{16L}\right).$$

Combine the two inequalities to see that

$$2^{3|S_K|-2K} \geq \min\left(1, \frac{\varepsilon}{16L\Delta_0}\right) = \frac{\varepsilon}{16L\Delta_0}.$$

Take the log in base 2 on both sides to reveal that

$$3|S_K| \geq 2K + \log_2\left(\frac{\varepsilon}{16L\Delta_0}\right).$$

Rearrange to conclude. □

Theorem 6.11. *Let assumptions A2 and A5 hold with the same constant L. Further require assumptions A4 and A1. For any $0 < \varepsilon \leq 16L\Delta_0$, if*

$$K > \frac{48L(f(x_0) - f_{\text{low}})}{\rho'} \frac{1}{\varepsilon^2} + \frac{1}{2} \log_2\left(\frac{16L\Delta_0}{\varepsilon}\right),$$

then there exists $k < K$ such that $\|\nabla f(x_k)\| \leq \varepsilon$.

Proof. This is a corollary of Theorem 6.9 and Lemma 6.10. □

Though it is not immediately clear from the statement for Theorem 6.11, it is also the case that $\|\nabla f(x_k)\|$ goes to zero in the limit as $k \to \infty$. (Do you see why this is not obvious from the theorem statement?) See [NW06, Thm. 4.6] if you are interested in the details.

Remark 6.12. *Comparing with Theorem 3.4 for gradient descent with constant step-size $1/L$, it would seem trust-regions is* worse *than GD in two ways: (a) the constant 48 is not as good, and (b) there is an additional log term. For (a), no need to worry: these constants have little meaning. For (b), notice that the trust-region method here does not need to know the constant L: the trust-region radius adapts automatically. The computational price of this automatic tuning is captured by the log term. In contrast, GD with step-size $1/L$ requires knowing the constant L. When we replace the constant step-size of GD with a line-search procedure, there is, in fact, also a log term.*

6.3 Truncated conjugate gradients

To compute Newton steps, we used Conjugate Gradients (CG), Algorithm 5.2. Recall from Section 5.4 that CG is effectively an iterative method to minimize

$$g(v) = \frac{1}{2} \langle v, Hv \rangle - \langle b, v \rangle,$$

under the assumption that H is positive definite. For Newton's method, we applied CG with $H = \nabla^2 f(x)$ and $b = -\nabla f(x)$.

For the trust-region method, we need to (approximately) solve the trust-region subproblem, that is, we must approximately minimize a model of the form (6.6) with the trust region:

$$m(v) = f(x) + \langle \nabla f(x), v \rangle + \frac{1}{2} \langle v, Hv \rangle, \qquad \text{subject to} \qquad \|v\| \leq \Delta,$$

Algorithm 6.2 tCG: truncated conjugate gradients

Input: symmetric linear map H on $\mathcal{E}$, $b \in \mathcal{E}$ and radius $\Delta > 0$

Output: approximate minimizer u of $m(v) = \frac{1}{2}\langle v, Hv\rangle - \langle b, v\rangle$ subject to $\|u\| \le \Delta$, and Hu as a by-product

Set $v_0 = 0, r_0 = b, p_0 = r_0$

For $n = 1, 2, \ldots$

Compute Hp_{n-1} (this is the only call to H)

Compute $\langle p_{n-1}, Hp_{n-1}\rangle$

$\alpha_n = \frac{\|r_{n-1}\|^2}{\langle p_{n-1}, Hp_{n-1}\rangle}$

$v_{n-1}^+ = v_{n-1} + \alpha_n p_{n-1}$

If $\langle p_{n-1}, Hp_{n-1}\rangle \le 0$ **or** $\|v_{n-1}^+\| \ge \Delta$

Set $v_n = v_{n-1} + tp_{n-1}$ with $t \ge 0$ such that $\|v_n\| = \Delta$

(See Exercise 6.15 for how to compute t.)

output $u = v_n$ and $Hu = b - r_{n-1} + tHp_{n-1}$

Else

$v_n = v_{n-1}^+$

$r_n = r_{n-1} - \alpha_n Hp_{n-1}$

If $\|r_n\| \le \|r_0\| \min(\|r_0\|, 0.1)$

output $u = v_n$ and $Hu = b - r_n$

$\beta_n = \frac{\|r_n\|^2}{\|r_{n-1}\|^2}$

$p_n = r_n + \beta_n p_{n-1}$

and it is typical to set $H = \nabla^2 f(x)$. Notice that for the purpose of minimizing $m(v)$ with respect to v, the presence of the constant term $f(x)$ is irrelevant (remember: for a subproblem, x is fixed).

Thus, minimizing $m(v)$ within the trust region is the same problem that CG solves in the context of computing Newton steps, up to two distinctions:

1. We are no longer allowed to assume $H \succ 0$.
2. We must remain in the trust region.

Accordingly, the key idea of the *truncated conjugate gradients* algorithm (tCG) is as follows:

> *In order to solve trust-region subproblems, let's run CG while*
>
> 1. *Keeping an eye out for signs that H is not positive definite,*
> 2. *Making sure we stay in the ball of radius Δ, and*
> 3. *Not working* too *hard, because after all we do not really care about solving the subproblem: we care about minimizing f.*

See Algorithm 6.2: this is a modification of CG. The parts of tCG that are different from CG are highlighted.

We now justify the modifications that distinguish tCG from CG, with numbering as above. Remember that the iterates of CG are the vectors $v_0, v_1, v_2, \ldots$ while the vectors $p_0, p_1, p_2, \ldots$ are the so-called *conjugate directions*.

1. When running CG, at iteration n, we consider the conjugate direction p_{n-1} and we compute the quadratic form $\langle p_{n-1}, Hp_{n-1}\rangle$. If that number is positive, then great. However, if it is negative, then this is proof that H is *not* positive definite. When this happens, we realize that the model $m(v)$ goes to $-\infty$ along the line that passes through v_{n-1} with direction p_{n-1}; indeed,

$$\begin{aligned} h(t) &= m(v_{n-1} + tp_{n-1}) \\ &= c_2 t^2 + c_1 t + c_0 \end{aligned} \tag{6.19}$$

   is a quadratic with some coefficients $c_0, c_1, c_2 \in \mathbb{R}$ (what is their value?), and $h(t)$ goes to $-\infty$ when we take $t \to \infty$ (why?). Since our goal is to make $m(v)$ small, and since we are not really sure what to do when H is not positive definite, we do the following: move along that line as far as we can, that is, until we reach the boundary of the trust region. Explicitly, that means: choose $t \geq 0$ such that $v_n \triangleq v_{n-1} + tp_{n-1}$

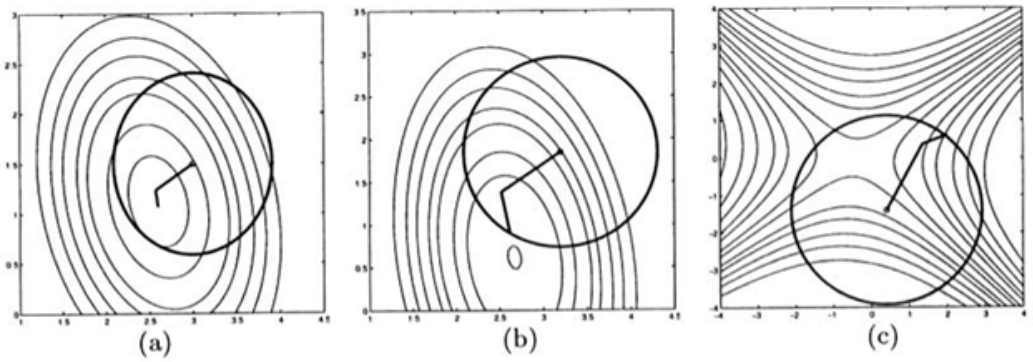

Figure 6.1: This is Figure 7.5.1 in [CGT00], describing different possible scenarios for tCG. (a) The model is convex and the solution is inside the trust region: we find it and stop there (in practice, it is more likely that we would find a point nearby and decide that it is good enough to stop); (b) the model is convex but the solution is outside the trust region and one of the iterates of tCG exits the ball: we move back, up to the boundary and stop; (c) the model is not convex: we detect negative curvature and move all the way to the boundary, where we stop. Note that other scenarios are possible. For example, the model might not be convex but we stop before "discovering" that fact.

is on the boundary: $\|v_n\|^2 = \Delta^2$. This is easy to do: the equation $\|v_n\|^2 = \Delta^2$ is a quadratic in t which has one positive root and one negative root; just set t to be the positive root. (See Exercise 6.15.) We stop here.

2. At iteration n, assume $\langle p_{n-1}, Hp_{n-1}\rangle > 0$ so that the situation we just described does not occur. We have already computed v_{n-1} and p_{n-1}, and we would like to go to our next iterate, v_n. Our hope is to let v_n be equal to the usual CG iterate, which for now we call $v_{n-1}^+ = v_{n-1} + \alpha_n p_{n-1}$. But what if this point is outside of the trust region, that is, what if $\|v_{n-1}^+\| > \Delta$? Then we are in a bit of a pickle. Perhaps if we keep iterating, eventually, the iterates would come back inside the trust region? Or maybe not? In fact, it can be shown that the norm of the iterates can only grow: if H is positive definite, then

$$0 = \|v_0\| < \|v_1\| < \|v_2\| < \cdots$$

Thus, baring the possibility that H is not positive definite (in which case we would act as described above), if v_{n-1}^+ is outside the trust region, then there is no point in continuing the iteration: we're not

going to come back to the trust region. Because of this, the strategy of tCG is as follows: v_{n-1} was inside the trust region, and v_{n-1}^+ is outside the trust region (or perhaps exactly on the boundary). Certainly, the line segment connecting v_{n-1} to v_{n-1}^+ intersects the boundary of the trust region at a single point. Let us set v_n to be that point. Here is a convenient observation: to find v_n, we need to move away from v_{n-1} toward v_{n-1}^+ until we hit the boundary of the trust region. The direction $v_{n-1}^+ - v_{n-1}$ is the same as p_{n-1} (because $\alpha_n > 0$—why?). Thus, to find v_n, we must choose $t \geq 0$ such that $v_{n-1} + tp_{n-1}$ has norm Δ. But this is exactly the same as in the previous situation! Therefore, we can use the same code for both situations. We stop here.

3. If neither of the situations above occur at iteration n, then we set $v_n = v_{n-1}^+$, as we normally would in a CG iteration. Notice that if we make it this far, it means that the special events above have never occurred (as otherwise we would have stopped). Thus, so far, everything has happened exactly as for a "normal" CG algorithm with a positive definite H.

   We now have the important task of deciding: should we keep working on this subproblem, or should we stop here and let the trust-region method (the "outer iteration") resume control? To this end, recall from (5.19) that r_n (as produced by the algorithm) is in fact the residue of the equation $Hu = b$ that CG is trying to solve:

$$r_n = b - Hv_n = -\nabla m(v_n).$$

   We know that r_n converges to zero as v_n converges to the solution. Notice moreover that

$$\|r_0\| = \|b\| = \|\nabla f(x_k)\|. \tag{6.20}$$

   The stopping criterion for tCG is designed on the following principle:

   (a) If $\|\nabla f(x_k)\|$ is still large, we should not work too hard on subproblems (after all, TR is probably still far from convergence).

   (b) On the other hand, if $\|\nabla f(x_k)\|$ is small, we *should* work hard: it's possible that we are close to convergence, so we should make steps that look more and more like Newton steps, to aim for quadratic convergence.

   This principle is encoded in the stopping criterion[3]

$$\|r_n\| \leq \|r_0\| \min(\|r_0\|, 0.1). \tag{6.21}$$

[3]The constant 0.1 is a parameter one could tune; in my experience, it is fine as is.

If $\|\nabla f(x_k)\| > 0.1$, tCG terminates as soon as $\|r_n\| \leq 0.1 \cdot \|\nabla f(x_k)\|$: it's not very ambitious. On the other hand, if $\|\nabla f(x_k)\| \leq 0.1$, then tCG terminates only once $\|r_n\| \leq \|\nabla f(x_k)\|^2$—this is ambitious: as $\nabla f(x_k)$ becomes smaller and smaller, this stopping criterion is rapidly forcing tCG to compute a solution that is very close to the Newton step (unless one of the events above occurs first.)

Let us end this section with a claim, two important exercises, and numerical remarks.

Proposition 6.13. *As tCG iterates, the value of the model can only decrease:*

$$m(v_0) \geq m(v_1) \geq m(v_2) \geq \cdots$$

Proof sketch. This holds because the same is true for CG, and because the changes we made to get tCG were carefully crafted so that this is still true. □

Exercise 6.14. *Verify that the first iterate of tCG, that is, v_1 as produced by Algorithm 6.2, is exactly the Cauchy step (Definition 6.4). What can you conclude regarding global convergence of the trust-region method with tCG? Hint: use Proposition 6.13 and Theorem 6.11.*

Exercise 6.15. *Consider again the definition $v_n \triangleq v_{n-1} + tp_{n-1}$ and the equation $\|v_n\|^2 = \Delta^2$, where the unknown is $t \in \mathbb{R}$. Show that the equation is indeed a quadratic equation in t and that, in the context above, it has one positive root and one negative root. How do you compute the positive root? Why do we choose the positive root and not the negative one? (There is a corner case where the root is double; what happens then? Why is it not an issue?)*

Remark 6.16. *A numerical comment: TR (Algorithm 6.1) makes certain decisions based on the norm of u_k; namely, it checks whether $\|u_k\| = \Delta_k$ or not. Numerically, this is tricky: it is a bad idea to write code of the form* `if norm(u_k) == Delta_k` *for example, because of round-off errors. A much more robust approach is to do the following: when running tCG, we* know *whether the solution we return, namely, u, satisfies $\|u\| = \Delta$ or not. Thus, arrange your code so that tCG also returns that information (as a boolean flag), and such that TR uses that information to make decisions.*

Remark 6.17. *As always with iterative algorithms, it is a good idea to include a bound on the maximum number of iterations, and to stop the loop if that bound is attained. Better this than to get stuck in an infinite loop*

due to some numerical issues. For tCG in particular, while it is true that mathematically the algorithm always terminates in at most $\dim \mathcal{E}$ *iterations (mostly, because CG has that property), in practice, things can get more complicated. A reasonable value for the maximum number of iterations in tCG is therefore* $\dim \mathcal{E}$*; sometimes, it is helpful to set it much lower; sometimes, it even makes sense to set it larger.*

6.4 Local convergence behavior

If we use the true Hessian in the model m_k (6.1), that is, $H_k = \nabla^2 f(x_k)$ (which is indeed the typical choice), then the trust-region method together with the truncated conjugate gradients algorithm enjoys quadratic local convergence in the same way that Newton's method does.

Essentially, the reason is this: once we get sufficiently close to a local minimizer where the Hessian is positive definite, the solution of the trust-region subproblem is the Newton step (that is because *eventually* steps stop being rejected, so that the trust-region radius stabilizes and the Newton steps are smaller than that radius), and the ambitious stopping criterion of tCG ensures that tCG terminates with a solution that is either exactly the Newton step, or very close to it.

Compare the following statement with Theorem 5.6 (local convergence of Newton's method). This result and Exercise 6.14—together—support the following claim: *trust-region methods combine the best of gradient descent and of Newton's method.*

Theorem 6.18. *Let* $f : \mathcal{E} \to \mathbb{R}$ *satisfy assumptions A2 and A3 (Lipschitz continuous gradient and Hessian). Let* x^* *be a critical point of* f *(i.e.,* $\nabla f(x^*) = 0$*) where* $\nabla^2 f(x^*) \succ 0$*. There exists a neighborhood of* x^* *such that if the sequence of iterates* $(x_k)_{k\geq 0}$ *generated by the trust-region method with tCG enters that neighborhood, then it stays in that neighborhood and converges to* x^* *at least quadratically.*

To be clear: after the convergence theorems for GD and Newton's method, we had numerous remarks about the importance of being careful in how to interpret them. For example, those theorems did not state that GD or Newton's method always converge, or that they converge only to local minima, or (certainly not!) that they only converge to global minima, etc. All of these remarks are still in order for the trust-region method: think critically about the precise statements of the theorems.

Still, we can enjoy the fact that TR with tCG is a really good method in practice, quite robust to our choices of parameters.

Part II

Constrained optimization

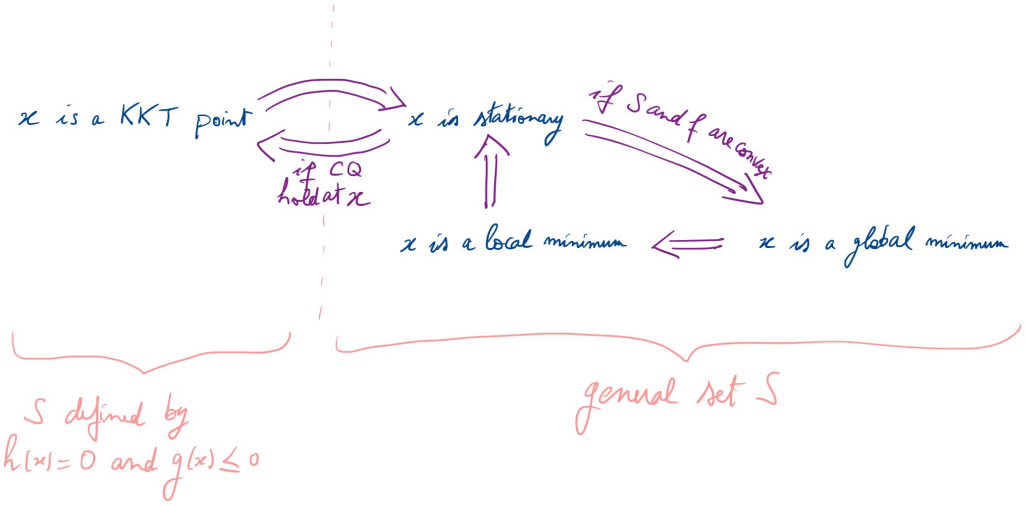

Figure 6.2: This diagram summarizes many important results we prove about constrained optimization. Make sure you know where each implication comes from (which theorems/corollaries). Make sure you understand why some implications are conditional or are altogether missing: do you have a counter-example? You should be able to draw this diagram quickly—it's useful when you work on some exercises.

Chapter 7

General principles: set constraints

In the first part of the course, we have considered the problem of minimizing a function $f\colon \mathcal{E} \to \mathbb{R}$ defined over a Euclidean space $\mathcal{E}$. In this second part of the course, we consider a more general class of problems: we aim to minimize f over a set $S \subseteq \mathcal{E}$. We write

$$\min_{x \in \mathcal{E}} f(x) \text{ subject to } x \in S, \tag{7.1}$$

or, more compactly,

$$\min_{x \in S} f(x). \tag{7.2}$$

This is called *constrained optimization*, because the variable x is constrained to the set S. The set S is called the *search space* or the *feasible set*. A point $x \in \mathcal{E}$ is called *feasible* if it is in S.

The special case where $S = \mathcal{E}$ corresponds to unconstrained optimization. We are mainly interested in the case $S \subset \mathcal{E}$. **We always assume that S is nonempty and closed.** As usual, $\mathcal{E}$ is equipped with an inner product $\langle \cdot, \cdot \rangle$ and associated norm $\| \cdot \|$.

In this first chapter, we consider the basics of constrained optimization at a general level. In particular, we discuss necessary optimality conditions for (7.2)

Recall the following elementary notions from Section 1.2.

Definition 7.1. *A* global minimizer *of $f\colon S \to \mathbb{R}$ is a point $x^* \in S$ such that $f(x) \geq f(x^*)$ for all $x \in S$.*

Definition 7.2. *A set $U \subseteq S$ is open in S if there exists an open set V in $\mathcal{E}$ such that $U = S \cap V$.*

In words: a subset of S is open in S if it is the intersection of S with an open subset of $\mathcal{E}$. A set is closed (in S) if its complement is open (in S).

Definition 7.3. *A* neighborhood *of x in S is an open subset of S which contains x.*

Definition 7.4. *A* local minimizer *of $f\colon S \to \mathbb{R}$ is a point $x^* \in S$ such that $f(x) \geq f(x^*)$ for all x in a neighborhood of x^* in S.*

In other words: $x^* \in S$ is a local minimizer of (7.2) if there exists a neighborhood V of x^* in $\mathcal{E}$ such that $f(x) \geq f(x^*)$ for all $x \in S \cap V$.

7.1 Tangent cones

Definition 7.5. *A set $K \subseteq \mathcal{E}$ is a* cone *if for all $v \in K$ and $\alpha > 0$ we also have $\alpha v \in K$.*

Definition 7.6. *A* closed cone *is a cone which is also a closed set.*

Definition 7.7. *Consider a point $x \in S$. The vector $v \in \mathcal{E}$ is a* tangent direction *at x for S if there exists a sequence $(x_k)_{k\geq 0}$ of points in S and a sequence of positive reals $(t_k)_{k\geq 0}$ such that:*

$$\lim_{k\to\infty} x_k = x, \qquad \lim_{k\to\infty} t_k = 0, \qquad \textit{and} \qquad \lim_{k\to\infty} \frac{x_k - x}{t_k} = v.$$

The set $\mathrm{T}_x S$ of all tangent directions at x for S is called the tangent cone *of S at x.*

Theorem 7.8. *Tangent cones are closed cones.*

Proof. Let us show that tangent cones are indeed cones. Given $v \in \mathrm{T}_x S$, there exist a sequence $(x_k) \subseteq S$ convergent to x and a sequence $(t_k) \subset \{t \in \mathbb{R} : t > 0\}$ convergent to zero such that $v = \lim_{k\to\infty} \frac{x_k - x}{t_k}$. We must show that αv is also in $\mathrm{T}_x S$ for an arbitrary $\alpha > 0$. That is indeed the case: simply consider the same sequence (x_k) with the new sequence (t'_k) defined by $t'_k = t_k/\alpha$. It is indeed still true that $x_k \to x$ and that $t'_k \to 0^+$. Moreover, $\lim_{k\to\infty} \frac{x_k - x}{t'_k} = \lim_{k\to\infty} \alpha \frac{x_k - x}{t_k} = \alpha v$, which confirms that αv is in $\mathrm{T}_x S$.

Let us show that the tangent cone is also a closed set, that is, let us show that if $(v_i)_{i\geq 0}$ is a sequence of tangent directions and $v = \lim_{i\to\infty} v_i$ exists, then v is also a tangent direction. For contradiction, assume v is not a tangent direction. Then, in particular, $v_i \neq v$ for all i. For each v_i, we have sequences $(x_{i,k})_{k\geq 0}$ in S and $(t_{i,k})_{k\geq 0}$ such that $\lim_{k\to\infty} x_{i,k} = x$,

$\lim_{k\to\infty} t_{i,k} = 0^+$ and $\lim_{k\to\infty} \frac{x_{i,k}-x}{t_{i,k}} = v_i$. Since $\|v_i - v\| > 0$, there exists an integer k_i such that

$$\left\| \frac{x_{i,k_i} - x}{t_{i,k_i}} - v_i \right\| \leq \|v_i - v\|.$$

Moreover, we can choose the integers k_i in such a way that $\lim_{i\to\infty} k_i = \infty$ (for example, require $k_i > k_{i-1}$ for $i = 1, 2, 3 \ldots$) By a triangular inequality,

$$\left\| \frac{x_{i,k_i} - x}{t_{i,k_i}} - v \right\| \leq 2\|v_i - v\|.$$

You can now verify using the sequences $(x_{i,k_i})_{i\geq 0}$ and $(t_{i,k_i})_{i\geq 0}$ that v is in fact a tangent direction at x: a contradiction. □

A convenient way to find tangent directions at x is to consider curves in S that start at x. This is also a useful perspective for intuition.

Theorem 7.9. *Given $x \in S$, consider a continuous curve $c\colon [0, \varepsilon] \to \mathcal{E}$ with $\varepsilon > 0$ such that $c(0) = x$, $c(t) \in S$ for all t and $c'(0)$ is well defined. Then, $c'(0) \in \mathrm{T}_x S$.*

Proof. By assumption, $c'(0) = \lim_{t\to 0^+} \frac{c(t)-c(0)}{t}$ is well defined. Consider the sequences $t_k = \varepsilon/k$ and $x_k = c(t_k)$ for $k = 1, 2, \ldots$ Notice that $t_k > 0$ and $x_k \in S$ for all k. Moreover, $t_k \to 0^+$, $x_k \to x$, and

$$\lim_{k\to\infty} \frac{x_k - x}{t_k} = \lim_{k\to\infty} \frac{c(t_k) - c(0)}{t_k} = \lim_{t\to 0^+} \frac{c(t) - c(0)}{t} = c'(0).$$

Therefore, $c'(0)$ is in $\mathrm{T}_x S$. □

The *interior of S* is the union of all open subsets of $\mathcal{E}$ contained in S.

Example 7.10. *If x is in the interior of S, then $\mathrm{T}_x S = \mathcal{E}$. Indeed, for all $v \in \mathcal{E}$, the curve $c(t) = x + tv$ stays in S for $t \geq 0$ sufficiently small, and $c'(0) = v$.*

Example 7.11. *Let S be a half-space, that is,*

$$S = \{x \in \mathcal{E} : \langle w, x\rangle \geq b\}$$

for some $w \in \mathcal{E}$ (nonzero) and $b \in \mathbb{R}$. The boundary of S is the affine space

$$\partial S = \{x \in \mathcal{E} : \langle w, x\rangle = b\}.$$

If x is in the interior of S, then $\mathrm{T}_x S = \mathcal{E}$ by Example 7.10. If x is on the boundary of S, then

$$\mathrm{T}_x S = \{v \in \mathcal{E} : \langle w, v\rangle \geq 0\}.$$

We can show this in two steps:

1. *If $v \in \mathrm{T}_x S$, then $\langle w, v\rangle \geq 0$. Indeed, there exist sequences (t_k) and (x_k) such that $t_k \to 0^+$ and $x_k \to x$ with $x_k \in S$ for all k and $v = \lim_{k\to\infty} \frac{x_k - x}{t_k}$. Thus,*

$$\langle w, v\rangle = \left\langle w, \lim_{k\to\infty} \frac{x_k - x}{t_k} \right\rangle = \lim_{k\to\infty} \frac{\langle w, x_k - x\rangle}{t_k} = \lim_{k\to\infty} \frac{\langle w, x_k\rangle - b}{t_k} \geq 0,$$

where the inequality follows from the facts that $\langle w, x_k\rangle \geq b$ and $t_k > 0$ for all k.

2. *Given $v \in \mathcal{E}$ such that $\langle w, v\rangle \geq 0$, we see that the curve $c(t) = x + tv$ stays in S for all $t \geq 0$ because*

$$\langle w, c(t)\rangle = \langle w, x + tv\rangle = \langle w, x\rangle + t\,\langle w, v\rangle \geq b,$$

so that $v = c'(0) \in \mathrm{T}_x S$.

Example 7.12. *Let S be an affine space defined by k linear equality constraints, that is*

$$S = \{x \in \mathcal{E} : L(x) = b\}$$

for some linear map $L\colon \mathcal{E} \to \mathbb{R}^k$ and $b \in \mathbb{R}^k$. (For example, consider the set $\{x : \langle w, x\rangle = 1\}$ which corresponds to $k = 1$.) The tangent cone at $x \in S$ is

$$\mathrm{T}_x S = \ker L = \{v \in \mathcal{E} : L(v) = 0\}.$$

Indeed, the curve $c(t) = x + tv$ stays in S for all t when v is in $\ker L$, which shows that $\ker L \subseteq \mathrm{T}_x S$. To verify that the reverse inclusion also holds, consider $v \in \mathrm{T}_x S$ together with sequences (x_k) and (t_k) as usual. Notice that

$$L(v) = L\left(\lim_{k\to\infty} \frac{x_k - x}{t_k}\right) = \lim_{k\to\infty} \frac{L(x_k) - L(x)}{t_k} = \lim_{k\to\infty} \frac{0}{t_k} = 0,$$

where we used linearity of L and $L(x) = b, L(x_k) = b$.

Example 7.13. *Let S be the unit ball in $\mathcal{E}$, that is,*

$$S = \{x \in \mathcal{E} : \|x\| \leq 1\}. \tag{7.3}$$

The boundary of S is the unit sphere:

$$\partial S = \{x \in \mathcal{E} : \|x\| = 1\}. \tag{7.4}$$

If x is in the interior of the ball (i.e., $\|x\| < 1$), we have $\mathrm{T}_x S = \mathcal{E}$ by Example 7.10. If x is on the boundary of the ball (i.e., $\|x\| = 1$), we have

$$\mathrm{T}_x S = \{v \in \mathcal{E} : \langle x, v\rangle \leq 0\}.$$

We show this in three steps:

1. *If v is in $\mathrm{T}_x S$, then $\langle x, v\rangle \leq 0$. Indeed, let (t_k) and (x_k) be sequences proving that v is in $\mathrm{T}_x S$; then*

$$\langle x, v\rangle = \left\langle x, \lim_{k\to\infty} \frac{x_k - x}{t_k} \right\rangle = \lim_{k\to\infty} \frac{\langle x, x_k - x\rangle}{t_k} = \lim_{k\to\infty} \frac{\langle x, x_k\rangle - 1}{t_k}.$$

   By Cauchy–Schwarz and since both x and x_k are in the unit ball, we have $\langle x, x_k\rangle \leq 1$. Combined with $t_k > 0$, we find indeed that $\langle x, v\rangle \leq 0$.

2. *If $\langle x, v\rangle < 0$, then $v \in \mathrm{T}_x S$. Indeed, it is easy to check that $c(t) = x + tv$ remains in the ball for sufficiently small $t \geq 0$.*

3. *It remains to show that if $\langle x, v\rangle = 0$ then $v \in \mathrm{T}_x S$. We can do this in two different ways:*

   (a) *The short way is to argue that since tangent cones are closed, and since we already know the tangent cone contains the set $\{v \in \mathcal{E} : \langle x, v\rangle < 0\}$, the tangent cone must also contain the closure of that set, i.e., $\{v \in \mathcal{E} : \langle x, v\rangle \leq 0\}$.*

   (b) *The explicit way is to construct an appropriate curve c. For example, consider $c(t) = \frac{x+tv}{\|x+tv\|}$. It is an exercise to check that c is continuous, that $c(t)$ is in the ball for all t, that $c(0) = x$ and that $c'(0) = v$—which confirms v is in $\mathrm{T}_x S$.*

Example 7.14. *Let S be the unit sphere in $\mathcal{E}$, that is,*

$$S = \{x \in \mathcal{E} : \|x\| = 1\}.$$

Given $x \in S$, the tangent cone is a linear space:

$$\mathrm{T}_x S = \{v \in \mathcal{E} : \langle x, v\rangle = 0\}.$$

Indeed, reasoning as in Example 7.13 we see that $\langle x, v\rangle = 0$ implies $v \in \mathrm{T}_x S$ using the curve $c(t) = \frac{x+tv}{\|x+tv\|}$. The other way around, we can prove that every tangent direction satisfies $\langle x, v\rangle = 0$. Indeed, let (t_k) and (x_k) be sequences which prove that $v \in \mathrm{T}_x S$. Then,

$$\langle x, v\rangle = \left\langle x, \lim_{k\to\infty} \frac{x_k - x}{t_k} \right\rangle = \lim_{k\to\infty} \frac{\langle x, x_k\rangle - 1}{t_k}.$$

Moreover, we know that, for all k:

$$\|x - x_k\|^2 = \|x\|^2 + \|x_k\|^2 - 2\langle x, x_k\rangle = 2(1 - \langle x, x_k\rangle).$$

Reorganizing, we get

$$\frac{\langle x, x_k\rangle - 1}{t_k} = -\frac{1}{2}\frac{\|x - x_k\|^2}{t_k}.$$

Notice that

$$\|v\| = \lim_{k\to\infty} \frac{\|x_k - x\|}{t_k}$$

Therefore,

$$\lim_{k\to\infty} \frac{\|x_k - x\|^2}{t_k} = \lim_{k\to\infty} \|x_k - x\| \frac{\|x_k - x\|}{t_k} = \|v\| \lim_{k\to\infty} \|x_k - x\| = 0.$$

Combining, we conclude that $\langle x, v\rangle = 0$.

Example 7.15. *Let* S *be the complement of the open unit ball in* $\mathcal{E}$*, that is,*

$$S = \{x \in \mathcal{E} : \|x\| \geq 1\}.$$

The boundary of S *is the unit sphere. If* x *is in the interior of* S *(i.e.,* $\|x\| > 1$*), we have* $\mathrm{T}_x S = \mathcal{E}$ *by Example 7.10. If* x *is on the boundary of* S *(i.e.,* $\|x\| = 1$*), then*

$$\mathrm{T}_x S = \{v \in \mathcal{E} : \langle x, v\rangle \geq 0\}.$$

Proving this is left as an exercise.

Example 7.16. *(This is a side note for the curious, based on [RW98, §6.A].) Theorem 7.9 tells us that some tangent directions of* S *at* x *can be realized as* $v = c'(0)$ *with a curve* c *in* S *such that* $c(0) = x$*. For "reasonable" sets* S*,* all *tangent directions are of that form. When that is the case,* S *is said to be* geometrically derivable*. But not all sets have that property. For example, consider*

$$S = \{(0, 0)\} \cup \{(x_1, x_2) : x_1 \neq 0 \text{ and } x_2 = x_1 \sin(\log(x_1))\},$$

a subset of $\mathbb{R}^2$*: plot it and zoom around the origin. At* $x = (0, 0)$*, it is easy to check that the tangent cone is* $\mathrm{T}_x S = \{(v_1, v_2) : |v_2| \leq |v_1|\}$*. Yet, no continuous curve in* S *passes through* x*, hence none of the tangent directions at* x *are of the form* $c'(0)$*.*

Exercise 7.17. *Complete the proof for Example 7.15. (Hint: take careful inspiration from the unit sphere example.)*

Exercise 7.18. *Let* S, S' *be two subsets of* $\mathcal{E}$ *such that* $S \subseteq S'$*. Show that* $\mathrm{T}_x S \subseteq \mathrm{T}_x S'$ *for all* $x \in S$*.*

Exercise 7.19. *Let* S_1, S_2 *be two subsets of* $\mathcal{E}$*, and let* x *belong to their intersection. Give an example where* $\mathrm{T}_x(S_1 \cap S_2) \neq (\mathrm{T}_x S_1) \cap (\mathrm{T}_x S_2)$*.*

7.2 A necessary optimality condition

The most important fact about tangent cones is that they provide us with necessary optimality conditions for constrained optimization problems.

Theorem 7.20. *Let $f\colon \mathcal{E} \to \mathbb{R}$ be differentiable, and $S \subseteq \mathcal{E}$. If $x^* \in S$ is a local minimum for f constrained to S, then*

$$\mathrm{D}f(x^*)[v] \geq 0 \quad \textit{for all} \quad v \in \mathrm{T}_{x^*}S.$$

However, the converse does not necessarily hold (exercise).

To prove this theorem, it is convenient to consider a lemma first.

Lemma 7.21. *Consider $x \in S$ and a tangent direction $v \in \mathrm{T}_x S$. Let (x_k) be a sequence in S convergent to x and let (t_k) be a sequence of positive reals convergent to 0 such that $v = \lim_{k\to\infty} \frac{x_k - x}{t_k}$. Then, if $f\colon \mathcal{E} \to \mathbb{R}$ is differentiable at x it holds that*

$$\mathrm{D}f(x)[v] = \lim_{k\to\infty} \frac{f(x_k) - f(x)}{t_k}.$$

Proof of Lemma 7.21. Since f is differentiable at x, it holds (by definition) that

$$\lim_{w\to 0} \frac{|f(x+w) - f(x) - \mathrm{D}f(x)[w]|}{\|w\|} = 0,$$

where w can approach 0 in an arbitrary way. In particular, let $w_k = x_k - x$, which indeed converges to zero. Plugging into the above, we get:

$$\lim_{k\to\infty} \frac{|f(x_k) - f(x) - \mathrm{D}f(x)[x_k - x]|}{\|x_k - x\|} = 0.$$

Divide both the numerator and the denominator by t_k (this is fine since t_k is nonzero for all k); then:

$$\lim_{k\to\infty} \frac{\left| \frac{f(x_k)-f(x)}{t_k} - \mathrm{D}f(x)\left[\frac{x_k - x}{t_k}\right] \right|}{\left\| \frac{x_k - x}{t_k} \right\|} = 0.$$

The denominator goes to $\|v\|$ with $k \to \infty$. This may or may not be nonzero. Either way, the numerator must converge to zero, hence:

$$\lim_{k\to\infty} \left| \frac{f(x_k) - f(x)}{t_k} - \mathrm{D}f(x)\left[\frac{x_k - x}{t_k}\right] \right| = 0.$$

Observe that the second term converges to $\mathrm{D}f(x)[v]$, hence the first term must converge to the same value. □

Now that we have the above lemma, it is easy to prove the main theorem of this section.

Proof of Theorem 7.20. Since x^* is a local minimum, there exists a neighborhood U of x^* in S such that $f(x) \geq f(x^*)$ for all $x \in U$. For contradiction, assume $\mathrm{D}f(x^*)[v] < 0$ for some $v \in \mathrm{T}_{x^*}S$. Let (x_k) and (t_k) be sequences associated to v as usual. By Lemma 7.21, we know that

$$\lim_{k\to\infty} \frac{f(x_k) - f(x^*)}{t_k} = \mathrm{D}f(x^*)[v] < 0.$$

Let $\varepsilon = -\frac{1}{2}\mathrm{D}f(x^*)[v] > 0$. There exists K such that, for all $k \geq K$,

$$\frac{f(x_k) - f(x^*)}{t_k} \leq -\varepsilon.$$

Reorganizing these terms, we get:

$$\forall k \geq K, \qquad f(x_k) \leq f(x^*) - t_k\varepsilon < f(x^*).$$

However, that is a contradiction because the sequence (x_k) converges to x^*: it must eventually enter the neighborhood U, yet we know that for all $x \in U$ we have $f(x) \geq f(x^*)$. □

Using the gradient of f, we can also write the necessary optimality condition as

$$\langle \nabla f(x^*), v\rangle \geq 0 \quad \text{for all} \quad v \in \mathrm{T}_{x^*}S. \tag{7.5}$$

This leads to other equivalent ways of phrasing the condition, through the notions of *dual* and *polar* of the tangent cone. Consider the following general definition first.

Definition 7.22. *Let C be a cone in $\mathcal{E}$. The* dual *of C is the set*

$$C^* = \{w \in \mathcal{E} : \langle w, v\rangle \geq 0 \text{ for all } v \in C\}.$$

The polar *of C is the set*

$$C^\circ = \{w \in \mathcal{E} : \langle w, v\rangle \leq 0 \text{ for all } v \in C\}.$$

It is an exercise to show that C^* and C° are closed cones.

Definition 7.23. *By Definition 7.22, the following conditions are equivalent:*

1. $\mathrm{D}f(x^*)[v] \geq 0$ *for all* $v \in \mathrm{T}_{x^*}S$.

2. $\nabla f(x^*) \in (\mathrm{T}_{x^*}S)^*$.

3. $-\nabla f(x^*) \in (\mathrm{T}_{x^*}S)^\circ$.

If x^* satisfies any (hence all) of them, we call x^* a critical *or a* stationary *point for the problem of minimizing f constrained to S.*

When we combine Theorem 7.20 with Definition 7.23, we get the following convenient statement.

Corollary 7.24. *Let $f\colon \mathcal{E} \to \mathbb{R}$ be differentiable, and $S \subseteq \mathcal{E}$. If $x^* \in S$ is a local minimum for f restricted to S, then x^* is stationary for that problem.*

The polar of the tangent cone has a special name.

Definition 7.25. *The* normal cone *to S at $x \in S$ is the set $\mathrm{N}_x S \triangleq (\mathrm{T}_x S)^\circ$.*

Thus, yet another way of defining stationary points, that is, of writing the necessary optimality conditions, is:

$$-\nabla f(x^*) \in \mathrm{N}_{x^*} S. \tag{7.6}$$

This expression is particularly nice to interpret: *a point is stationary if the steepest descent direction at that point lies in the normal cone at that point.*

It is important to note that Theorem 7.20 only provides *necessary* optimality conditions. In general, those conditions are not sufficient. It is an exercise to verify this claim on a specific example.

Exercise 7.26. *Show that C^* and C° are closed cones (even if C is not).*

Exercise 7.27. *Let C, C' be two cones in $\mathcal{E}$ such that $C \subseteq C'$. Show that $(C')^\circ \subseteq C^\circ$, and likewise that $(C')^* \subseteq C^*$.*

Exercise 7.28. *For each example in Section 7.1, draw the set S and the tangent cones and normal cones at some interesting points. Think about the meaning of the necessary optimality condition $-\nabla f(x^*) \in \mathrm{N}_{x^*} S$. Do the same also for the following sets:*

1. *The unit square in $\mathbb{R}^2$, that is, $\{x \in \mathbb{R}^2 : 0 \leq x_1, x_2 \leq 1\}$.*

2. *The half disk $\{x \in \mathbb{R}^2 : \|x\| \leq 1$ and $x_1 \geq 0\}$.*

3. *The "opposite" of the half disk, $\mathbb{R}^2 \backslash \{x \in \mathbb{R}^2 : \|x\| < 1$ and $x_1 > 0\}$.*

Exercise 7.29. *Let $\mathcal{E} = \mathbb{R}^2$. Consider the function $f(x) = x_2$, to be minimized on the set $S = \{x \in \mathbb{R}^2 : \|x\| \geq 1\}$ (the complement of the open unit disk). Consider the special point $x^* = (0, 1)^\top$. Show that x^* is stationary for f on S. Show that x^* is not a local minimum for f on S. This example proves that the converse of Theorem 7.20 does not hold in general.*

Exercise 7.30. *When $S = \mathcal{E}$, the optimization problem is unconstrained. We already showed that a necessary optimality condition at x^* in that case is $\nabla f(x^*) = 0$. Show how we can recover that fact using normal cones of $\mathcal{E}$.*

Exercise 7.31. *Given a non-empty set $Q \subseteq \mathcal{E}$, the* projection *of a point $z \in \mathcal{E}$ to Q is defined as the set of solutions of*

$$\min_{x \in Q} \|x - z\|. \tag{7.7}$$

We let $\mathrm{Proj}_Q(z)$ denote this set. Fact: if Q is closed, the set $\mathrm{Proj}_Q(z)$ is non-empty. However, even then, $\mathrm{Proj}_Q(z)$ might not be a singleton: give an example. (See also Exercise 9.17)

Exercise 7.32. *Let C be a closed cone. Show that*

$$\mathrm{Proj}_C(z) = \{0\} \iff z \in C^\circ.$$

Hint: Notice that the elements of $\mathrm{Proj}_C(z)$ are the solutions of an optimization problem: we know they satisfy certain conditions.

Exercise 7.33. *Show that $x^* \in S$ is stationary for f constrained to S if and only if*

$$\mathrm{Proj}_{\mathrm{T}_{x^*}S}(-\nabla f(x^*)) = \{0\}.$$

This provides yet another way to interpret the necessary optimality conditions for constrained optimization problems: the steepest descent direction vanishes when we project it to the tangent cone. Hint: solve Exercise 7.32 first.

Chapter 8

Search spaces defined by equalities and inequalities

It is common for the search space S of an optimization problem

$$\min_{x \in \mathcal{E}} f(x) \text{ subject to } x \in S$$

to be defined through equality constraints and inequality constraints, as follows:

$$S = \Big\{ x \in \mathcal{E} : h_i(x) = 0,\ i = 1, \dots, p, \text{ and} \\ g_i(x) \leq 0,\ i = 1, \dots, m \Big\}, \tag{8.1}$$

where $h_1, \dots, h_p \colon \mathcal{E} \to \mathbb{R}$ and $g_1, \dots, g_m \colon \mathcal{E} \to \mathbb{R}$ are continuously differentiable functions. More compactly, we write

$$S = \{x \in \mathcal{E} : h(x) = 0 \text{ and } g(x) \leq 0\}, \tag{8.2}$$

with $h \colon \mathcal{E} \to \mathbb{R}^p$ and $g \colon \mathcal{E} \to \mathbb{R}^m$ defined by

$$h(x) = \begin{bmatrix} h_1(x) \\ \vdots \\ h_p(x) \end{bmatrix}, \qquad g(x) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_m(x) \end{bmatrix},$$

and the notation $u \leq 0$ for a vector $u \in \mathbb{R}^m$ means $u_i \leq 0$ for $i = 1, \dots, m$.

The fundamental necessary optimality conditions are still the same as those expressed in (7.6), that is: local minima are stationary points, and a point $x^* \in S$ is stationary if

$$-\nabla f(x^*) \in \mathrm{N}_{x^*} S = (\mathrm{T}_{x^*} S)^\circ.$$

Thus, our goals in this chapter are clear:

1. Figure out nice expressions for the tangent cones of S,
2. Deduce nice expressions for the normal cones of S,
3. Conclude with a nice statement about the necessary optimality conditions for minimization problems constrained to S.

This is not as straightforward as it might first seem to be. We will get there through a few examples and intermediate steps.

8.1 A single inequality constraint

Let $g\colon \mathcal{E} \to \mathbb{R}$ be a continuously differentiable function. Consider the set

$$S = \{x \in \mathcal{E} : g(x) \leq 0\}. \tag{8.3}$$

Many of the sets we have considered can be described in this fashion. For example, the unit ball can be described with the function $g(x) = \|x\|^2 - 1$ (and we could also have picked a different function g to get the same set).

Consider a point $x \in S$. If $g(x) < 0$, then it is easy to determine the tangent cone at x.

Theorem 8.1. *If $g(x) < 0$, then $\mathrm{T}_x S = \mathcal{E}$.*

Proof. Since g is continuous, there exists a neighborhood U of x in $\mathcal{E}$ such that $g(y) < 0$ for all $y \in U$. Thus, x is in the interior of S. It follows from Example 7.10 that $\mathrm{T}_x S = \mathcal{E}$. □

If $g(x) = 0$, things can get more complicated: it all depends on the gradient of the constraint at x. Specifically, it all depends on whether or not $\nabla g(x)$ is zero. To build some intuition, consider the two following examples.

Example 8.2. *With $\mathcal{E} = \mathbb{R}^2$, let $g(x) = x_1^2 + x_2^2 - 1$. The point $x = (1, 0)$ satisfies $g(x) = 0$. Notice that $\nabla g(x) = (2x_1, 2x_2)^\top = (2, 0)^\top$. The tangent cones and normal cones at x are given by:*

$$\mathrm{T}_x S = \{v \in \mathbb{R}^2 : v_1 \leq 0\}, \qquad \mathrm{N}_x S = \{v \in \mathbb{R}^2 : v_1 \geq 0 \text{ and } v_2 = 0\}.$$

We can also express these cones as follows:

$$\mathrm{T}_x S = \{v \in \mathcal{E} : \langle \nabla g(x), v\rangle \leq 0\}, \qquad \mathrm{N}_x S = \{\lambda \nabla g(x) : \lambda \geq 0\}.$$

This is not an accident: you can check that this holds for all x such that $g(x) = 0$. We shall see that this extends to other functions g as well.

Example 8.3. *With $\mathcal{E} = \mathbb{R}^2$, let $g(x) = x_2^2 - x_1^3$. The point $x = (0,0)$ satisfies $g(x) = 0$. Notice that $\nabla g(x) = (-3x_1^2, 2x_2)^\top = (0,0)$. Moreover,*

$$\mathrm{T}_x S = \{v \in \mathbb{R}^2 : v_1 \geq 0 \text{ and } v_2 = 0\}, \qquad \mathrm{N}_x S = \{v \in \mathbb{R}^2 : v_1 \leq 0\}.$$

In this case, we do not have a simple relationship between the gradient $\nabla g(x)$ and the tangent or normal cone. This is because $\nabla g(x) = 0$.

In line with the first example above, if $\nabla g(x) \neq 0$, then we can get an expression for the tangent and normal cones to S at x. The second example above illustrates why we need to restrict our attention to settings where $\nabla g(x) \neq 0$.

Theorem 8.4. *If $x \in S$ satisfies $g(x) = 0$ and $\nabla g(x) \neq 0$, then*

$$\mathrm{T}_x S = \{v \in \mathcal{E} : \langle \nabla g(x), v \rangle \leq 0\}, \qquad \mathrm{N}_x S = \{\lambda \nabla g(x) : \lambda \geq 0\}.$$

Proof. Consider an arbitrary direction $v \in \mathcal{E}$. There are three cases to study:

1. $\langle \nabla g(x), v \rangle > 0$: in this case, we show that $v \notin \mathrm{T}_x S$. Indeed, assume for contradiction that we can find $(x_k) \to x$ in S and $(t_k) \to 0^+$ such that $\frac{x_k - x}{t_k} \to v$. Then, Lemma 7.21 tells us that

$$\langle \nabla g(x), v \rangle = \mathrm{D}g(x)[v] = \lim_{k \to \infty} \frac{g(x_k) - g(x)}{t_k} \leq 0$$

because $g(x) = 0$ and $g(x_k) \leq 0, t_k > 0$ for all k.

2. $\langle \nabla g(x), v \rangle < 0$: let us show that $v \in \mathrm{T}_x S$. Notice that

$$g(x + tv) = g(x) + t \langle \nabla g(x), v \rangle + o(t) = t \langle \nabla g(x), v \rangle + o(t).$$

Since $\langle \nabla g(x), v \rangle < 0$, we deduce that for $t \in [0, \varepsilon]$ with $\varepsilon > 0$ small enough the points $c(t) = x + tv$ satisfy $g(x) \leq 0$. Therefore, $c'(0) = v$ belongs to $\mathrm{T}_x S$ (use Theorem 7.9).

3. $\langle \nabla g(x), v \rangle = 0$: Above we have shown that

$$\{v \in \mathcal{E} : \langle \nabla g(x), v \rangle < 0\} \subseteq \mathrm{T}_x S.$$

Vectors v such that $\langle \nabla g(x), v \rangle = 0$ are in the closure of the left-hand side. Yet, we know that $\mathrm{T}_x S$ is closed. Therefore, the closure of the left-hand side is included in $\mathrm{T}_x S$.

It remains to deduce $\mathrm{N}_x S$ from $\mathrm{T}_x S$: this is left as an exercise. □

Exercise 8.5. *Complete the proof of Theorem 8.4.*

8.2 A single equality constraint

Let $h\colon \mathcal{E} \to \mathbb{R}$ be a continuously differentiable function. Consider the set

$$S = \{x \in \mathcal{E} : h(x) = 0\}. \tag{8.4}$$

Here too we can learn a lot about the tangent cones of S by studying the gradient of h. Let us do so through two examples.

Example 8.6. *With $h(x) = x_1^2 + x_2^2 - 1$ defined on $\mathbb{R}^2$, the set S is the unit circle in the plane. Notice that $\nabla h(x) = (2x_1, 2x_2)^\top$ never vanishes for $x \in S$: it is always "linearly independent". As detailed earlier, it is easy to check that*

$$\mathrm{T}_x S = \{v \in \mathcal{E} : \langle \nabla h(x), v\rangle = 0\}, \quad \mathrm{N}_x S = \{\mu \nabla h(x) \in \mathcal{E} : \mu \in \mathbb{R}\}. \tag{8.5}$$

As we shall see, this observation generalizes well under some conditions.

Example 8.7. *With $h(x) = x_2^2 - x_1^3$ defined on $\mathbb{R}^2$, the set S is the so-called* cuspidal cubic. *Notice that $\nabla h(x) = (-3x_1^2, 2x_2)^\top$ vanishes at $x = (0, 0)$, which belongs to S. By inspection on a drawing, it is easy to see that for this choice of x we have:*

$$\mathrm{T}_x S = \{v \in \mathcal{E} : v_1 \geq 0 \text{ and } v_2 = 0\}, \qquad \mathrm{N}_x S = \{v \in \mathcal{E} : v_1 \leq 0\}.$$

These cones are not *what we would obtain using the descriptions in (8.5); that is because $\nabla h(x) = 0$: it is not "linearly independent".*

8.3 Two inequality constraints

Consider the set $S \subseteq \mathbb{R}^2$ defined by the two following inequality constraints:

$$g_1(x) \triangleq x_1^2 + x_2^2 - 1 \leq 0, \qquad g_2(x) \triangleq x_1 + x_2 - 1 \leq 0.$$

The first inequality describes the unit disk in the plane and the second inequality defines a half-space; the set S is the intersection of these two sets.

What can we say about the tangent cones to S? There are five scenarios depending on where the point x is in $\mathbb{R}^2$:

1. If $g_1(x) > 0$ or $g_2(x) > 0$ (or both), then $x \notin S$: there are no tangent cones here because x is infeasible.

2. If $g_1(x) < 0$ and $g_2(x) < 0$, then x is in the interior of S. Indeed, g_1 and g_2 are continuous functions: there exists a neighborhood U_1 of x such that $g_1(y) < 0$ for all $y \in U_1$, and likewise there exists a neighborhood U_2 of x such that $g_2(y) < 0$ for all $y \in U_2$. Therefore, $U = U_1 \cap U_2$ is a neighborhood of x which is entirely contained in S. It then follows that $\mathrm{T}_x S = \mathcal{E} = \mathbb{R}^2$. We say that neither of the constraints are active.

3. If $g_1(x) = 0$ and $g_2(x) < 0$, then only the first constraint is active. The second constraint has no effect on what S looks like locally around x. Therefore, we can ignore the inactive constraint and use everything we have learned about tangent cones under a single inequality constraint to treat this case. Explicitly, $\mathrm{T}_x S = \{v \in \mathbb{R}^2 : \langle \nabla g_1(x), v \rangle \leq 0\}$.

4. If $g_1(x) < 0$ and $g_2(x) = 0$, then only the second constraint is active: we can use the same reasoning as above. Explicitly, $\mathrm{T}_x S = \{v \in \mathbb{R}^2 : \langle \nabla g_2(x), v \rangle \leq 0\}$.

5. If $g_1(x) = 0$ and $g_2(x) = 0$, then both constraints are active: this is the most interesting case for us now because it is the most different from everything else we have considered so far. We reason on this particular example. Let $x = (1, 0)$: both constraints are indeed active at this point. Consider the gradients of the constraints at this point:

$$\nabla g_1(x) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \qquad \nabla g_2(x) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Notice that these gradients are linearly independent: this will be important. By inspection on a drawing, it is not difficult to convince ourselves that the following identities hold:

$$\mathrm{T}_x S = \left\{v \in \mathbb{R}^2 : \langle \nabla g_1(x), v \rangle \leq 0 \text{ and } \langle \nabla g_2(x), v \rangle \leq 0\right\},$$
$$\mathrm{N}_x S = \left\{\lambda_1 \nabla g_1(x) + \lambda_2 \nabla g_2(x) : \lambda_1, \lambda_2 \geq 0\right\}.$$

In the sequel, we will formalize this observation in broad generality.

8.4 Linearized feasible directions

To summarize the introductory sections above: We can learn a lot about $S = \{x \in \mathcal{E} : h(x) = 0 \text{ and } g(x) \leq 0\}$ by studying the linearization of the constraints around feasible points. Indeed, consider $x \in S$ and an arbitrary

direction $v \in \mathcal{E}$. Taylor expansions of the constraints $h_i(x) = 0$ tell us that[1]

$$\begin{aligned} h_i(x + tv) &= h_i(x) + t \langle \nabla h_i(x), v \rangle + o(t) \\ &= t \langle \nabla h_i(x), v \rangle + o(t). \end{aligned}$$

Thus, if $\langle \nabla h_i(x), v \rangle = 0$ then $x+tv$ nearly satisfies the constraint: $h_i(x+tv) = o(t) \approx 0$. Likewise, consider a Taylor expansion for one of the inequality constraints:

$$g_i(x + tv) = g_i(x) + t \langle \nabla g_i(x), v \rangle + o(t).$$

If $g_i(x) < 0$, then for small enough t it is the case that $g_i(x + tv) \leq 0$, and therefore the constraint remains satisfied: there is no particular condition on v. On the other hand, if $g_i(x) = 0$ (that is, if this inequality constraint is active at x), then we must require $\langle \nabla g_i(x), v \rangle \leq 0$ to ensure that $x + tv$ (nearly) satisfies the inequality constraint for small $t \geq 0$.

The above considerations amount to *linearizing* the active constraints at $x \in S$. This leads to the notion of *linearized feasible directions* defined below. These are nothing but the tangent directions of the linearized constraints.

Definition 8.8. *Consider the set S defined by equalities $h(x) = 0$ and inequalities $g(x) \leq 0$ as in (8.2). Given $x \in S$, the* cone of linearized feasible directions *at x to S with respect to these constraints is the set*

$$\begin{aligned} \mathrm{F}_x S = \Big\{ v \in \mathcal{E} : \langle \nabla h_i(x), v \rangle = 0,\ i = 1, \ldots, p,\ \textit{and} \\ \langle \nabla g_i(x), v \rangle \leq 0,\ i = 1, \ldots, m \textit{ such that } g_i(x) = 0 \Big\}. \end{aligned} \tag{8.6}$$

It is an exercise to verify that this is a closed cone.

Theorem 8.9. *The tangent cone is always included in the cone of linearized feasible directions, that is,*

$$\mathrm{T}_x S \subseteq \mathrm{F}_x S.$$

However, the two cones are not always equal (exercise).

Proof. Consider a tangent direction $v \in \mathrm{T}_x S$. By definition, we can select a sequence (x_k) in S and a sequence (t_k) of positive reals such that $x_k \to x$, $t_k \to 0^+$ and $\frac{x_k - x}{t_k} \to v$. For each $1 \leq i \leq p$, we have $h_i(x) = 0$ and $h_i(x_k) = 0$

[1] Recall that $o(t)$ stands for a quantity which is "much smaller than t when t is small". If h_i is twice continuously differentiable we can write $O(t^2)$ instead of $o(t)$.

since x and all the points in the sequence (x_k) are feasible. It then follows from Lemma 7.21 that

$$\langle \nabla h_i(x), v \rangle = \mathrm{D}h_i(x)[v] = \lim_{k\to\infty} \frac{h_i(x_k) - h_i(x)}{t_k} = 0.$$

Likewise, for $1 \leq i \leq m$ such that $g_i(x) = 0$ we see that

$$\langle \nabla g_i(x), v \rangle = \mathrm{D}g_i(x)[v] = \lim_{k\to\infty} \frac{g_i(x_k) - g_i(x)}{t_k} \leq 0$$

since $g_i(x_k) \leq 0$ for all points in the sequence (x_k). □

Remark 8.10. *Note the following: tangent cones are a fundamental property of the set S, whereas cones of linearized feasible directions depend on how we describe the set S. In other words: $\mathrm{T}_x S$ depends only on the set S, whereas $\mathrm{F}_x S$ can depend on the specific equality and inequality constraints we choose to define S. Keep that in mind when we use the notation "$\mathrm{F}_x S$".*

Exercise 8.11. *Verify that $\mathrm{F}_x S$ is a closed cone.*

Exercise 8.12. *Consider the set S which is simply the origin in $\mathbb{R}^2$. What is the tangent cone to S at $x = (0, 0)$? Consider a first way to define this set S, through two equality constraints $h_1(x) = 0$ and $h_2(x) = 0$ with*

$$h_1(x) = x_1, \qquad h_2(x) = x_2.$$

What is the corresponding cone of linearized feasible directions at x? Now consider a second way to define the set S, through two inequality constraints $g_1(x) \leq 0$ and $g_2(x) \leq 0$ with

$$g_1(x) = (x_1 - 1)^2 + x_2^2 - 1, \qquad g_2(x) = x_1.$$

What is the corresponding cone of linearized feasible directions at x? Comment on your findings with respect to Theorem 8.9 and Remark 8.10

8.5 Constraint qualifications

Theorem 8.9 tells us that the tangent cone is always included in the cone of linearized feasible directions, that is, $\mathrm{T}_x S \subseteq \mathrm{F}_x S$, but that these two cones may not be equal to one another. If they are not equal, then we have learned little. Indeed,

$$x^* \text{ is a local minimum} \qquad \Longrightarrow \qquad -\nabla f(x^*) \in (\mathrm{T}_{x^*} S)^\circ,$$

yet

$$\mathrm{T}_x S \subseteq \mathrm{F}_x S \qquad \Longrightarrow \qquad (\mathrm{F}_x S)^\circ \subseteq (\mathrm{T}_x S)^\circ.$$

In general, we *cannot* conclude anything useful relating $-\nabla f(x^*)$ to $(\mathrm{F}_{x^*} S)^\circ$.

Accordingly, an important topic in constrained optimization is to determine conditions on the equality and inequality constraints that lead to $\mathrm{T}_x S = \mathrm{F}_x S$. Such conditions are called *constraint qualification conditions*. Let us stress again that these are conditions on the way in which we describe S (that is, conditions on our choices of equality and inequality constraints)—they are *not* conditions on the set S itself.

Definition 8.13. *Given a set S described by equality and inequality constraints as in (8.1), we say that* constraint qualification conditions *(CQ) hold at $x \in S$ if $\mathrm{T}_x S = \mathrm{F}_x S$.*

There exist many different kinds of constraint qualification conditions. Let us consider a particularly simple one in the theorem below. Define $\mathcal{I}(x)$ as the set of active inequality constraints at x:

$$\mathcal{I}(x) = \{i \in \{1, \dots, m\} : g_i(x) = 0\} \tag{8.7}$$

Theorem 8.14. *Consider a point $x \in S$ where S is defined by (8.1). If the gradients of the active constraints at x are linearly independent, that is, if*

$$\nabla h_1(x), \dots, \nabla h_p(x) \text{ and } \nabla g_i(x) \ \forall i \in \mathcal{I}(x)$$

are linearly independent vectors in $\mathcal{E}$, then $\mathrm{T}_x S = \mathrm{F}_x S$ and we say the linear independence constraint qualification condition (LICQ) *holds at x.*

Proof. You can find a proof in [NW06, Lem. 12.2(ii)]. The main ingredient is the *inverse function theorem* from multivariate calculus. The proof uses that theorem to construct curves in S that can be used with Theorem 7.9 to show that each element of $\mathrm{F}_x S$ is in $\mathrm{T}_x S$. □

We mention two further constraint qualification conditions. The first one below is fairly clear: if the constraints are already linear, then linearizing them does not create particular issues (in fact, it does nothing at all).

Theorem 8.15. *Consider a point $x \in S$ where S is defined by (8.1). If all active constraints at $x \in S$ are linear, that is, if all equality constraint functions $h_1, \dots, h_p$ and all inequality constraint functions g_i with $i \in \mathcal{I}(x)$ are affine functions, then $\mathrm{T}_x S = \mathrm{F}_x S$.*

Proof. See [NW06, Lem. 12.7]. The idea is as follows: (a) we already know that $\mathrm{T}_x S \subseteq \mathrm{F}_x S$ from Theorem 8.9; (b) to show that $\mathrm{F}_x S \subseteq \mathrm{T}_x S$, we consider curves of the form $c(t) = x + tv$ where $v \in \mathrm{F}_x S$. It suffices to argue that $c(t) \in S$ for all $t \in [0, \varepsilon]$ with some $\varepsilon > 0$, then to apply Theorem 7.9. □

The last constraint qualification we mention is a standard generalization of LICQ.

Theorem 8.16. *Consider a point $x \in S$ where S is defined by (8.1). If the gradients $\nabla h_1(x), \ldots, \nabla h_p(x)$ are linearly independent and there exists a point $\bar{x} \in \mathcal{E}$ such that*

$$\begin{aligned} &\langle \nabla h_i(x), \bar{x} - x \rangle = 0, && \text{for all } i = 1, \ldots, p \text{ and} \\ &\langle \nabla g_i(x), \bar{x} - x \rangle < 0, && \text{for all } i \in \mathcal{I}(x), \end{aligned}$$

then $\mathrm{T}_x S = \mathrm{F}_x S$ and we say the Mangasarian–Fromowitz constraint qualification condition (MFCQ) *holds at x.*

If LICQ holds, then MFCQ holds. However, the converse is not true.

The following technical lemma about MFCQ is sometimes useful.

Lemma 8.17. *If MFCQ holds at x, then there exists $\epsilon > 0$ such that for all $v \in \mathbb{R}^p$ with $\|v\| \leq \epsilon$ there exists $\hat{x} \in \mathcal{E}$ satisfying*

$$\begin{aligned} &\langle \nabla h_i(x), \hat{x} - x \rangle = v_i, && \text{for all } i = 1, \ldots, p \text{ and} \\ &\langle \nabla g_i(x), \hat{x} - x \rangle < 0, && \text{for all } i \in \mathcal{I}(x). \end{aligned}$$

Proof. Let $\hat{x} = \bar{x} + w$ where $\bar{x}$ is provided by MFCQ and w is to be determined. The requirements from the equality constraints then state

$$v_i = \langle \nabla h_i(x), \bar{x} + w - x \rangle = \langle \nabla h_i(x), w \rangle \qquad \text{for all } i = 1, \ldots, p.$$

Since the gradients $\nabla h_i(x)$ are linearly independent, solutions w exist for any given v: pick the one of minimal norm. As long as w is small enough, the inequalities $\langle \nabla g_i(x), \hat{x} - x \rangle < 0$ will hold by continuity. The norm of w is bounded by the norm of v times a constant (that is a general property of the minimal norm solutions of consistent linear systems). Thus, we get the result by selecting $\epsilon > 0$ small enough. □

Exercise 8.18. *Show that if LICQ holds then MFCQ holds.*

Exercise 8.19. *(See [NW06, Exercise 12.13].) Consider the set S defined by the following inequality constraints on $\mathcal{E} = \mathbb{R}^2$:*

$$\begin{aligned} g_1(x) &= (x_1 - 1)^2 + (x_2 - 1)^2 - 2 \leq 0, \\ g_2(x) &= (x_1 - 1)^2 + (x_2 + 1)^2 - 2 \leq 0, \\ g_3(x) &= -x_1 \leq 0. \end{aligned}$$

Draw the three sets defined by $g_1(x) \leq 0$, $g_2(x) \leq 0$ and $g_3(x) \leq 0$, and highlight the set S on your drawing. Draw the gradients of the constraints at the origin. Show that MFCQ holds at the origin yet LICQ does not hold at the origin.

8.6 Polar of the linearized directions

When constraint qualifications hold at a point x^*, we have the identity $\mathrm{T}_{x^*}S = \mathrm{F}_{x^*}S$ which allows us to claim:

$$x^* \text{ is a local minimum} \qquad \Longrightarrow \qquad -\nabla f(x^*) \in (\mathrm{F}_{x^*}S)^\circ.$$

Statements of the form "$w \in (\mathrm{F}_x S)^\circ$" are not so convenient to exploit though. Explicitly, they mean that w satisfies the following conditions:

$$\langle w, v\rangle \leq 0 \text{ for all } v \in \mathcal{E} \text{ such that } \begin{cases} \langle \nabla h_i(x), v\rangle = 0 & \text{for } i = 1, \ldots, p, \text{ and} \\ \langle \nabla g_i(x), v\rangle \leq 0 & \text{for } i = 1, \ldots, m \text{ such that } g_i(x) = 0. \end{cases} \tag{8.8}$$

It would be much more convenient to have an expression stating that w is of a certain form, such as a linear combination of particular vectors for example. In other words: we need an explicit description of the polar of $\mathrm{F}_x S$.

Based on the examples we have encountered so far, we may surmise that vectors of the following form merit our attention:

$$w = \sum_{i=1}^{p} \mu_i \nabla h_i(x) + \sum_{i \in \mathcal{I}(x)} \lambda_i \nabla g_i(x), \tag{8.9}$$

where λ_i, μ_i are real coefficients and $\mathcal{I}(x)$ indexes the active inequality constraints at x as in (8.7). Consider an arbitrary vector $v \in \mathrm{F}_x S$: it satisfies the conditions stated in (8.8). Therefore,

$$\langle w, v\rangle = \sum_{i=1}^{p} \mu_i \underbrace{\langle \nabla h_i(x), v\rangle}_{=0} + \sum_{i \in \mathcal{I}(x)} \lambda_i \underbrace{\langle \nabla g_i(x), v\rangle}_{\leq 0}.$$

In particular, if $\lambda_i \geq 0$ for all $i \in \mathcal{I}(x)$ then $\langle w, v\rangle \leq 0$ for all $v \in \mathrm{F}_x S$, that is, $w \in (\mathrm{F}_x S)^\circ$. As it turns out, this is a complete description of the polar of $\mathrm{F}_x S$. The tool of choice to complete the proof is the *Farkas lemma* (see [NW06, Lem. 12.4] for a proof, or https://arxiv.org/abs/2208.11678 for a short one in case $p = 0$.)

Lemma 8.20 (Farkas lemma)**.** *Let $u_1, \dots, u_p \in \mathcal{E}$ and $v_1, \dots, v_{m'} \in \mathcal{E}$ be arbitrary vectors. Consider the cone*

$$K = \left\{ \sum_{i=1}^{p} \mu_i u_i + \sum_{i=1}^{m'} \lambda_i v_i : \mu_i \in \mathbb{R}, \lambda_i \geq 0 \right\}.$$

Then, given a vector $w \in \mathcal{E}$, exactly one *of the following statements is true:*

1. *Either $w \in K$, or*
2. *There exists $v \in \mathcal{E}$ such that $\langle u_i, v \rangle = 0$ for all i, $\langle v_i, v \rangle \leq 0$ for all i, and $\langle w, v \rangle > 0$,*

but not both.

Theorem 8.21. *The polar of $\mathrm{F}_x S$ is the following cone:*

$$(\mathrm{F}_x S)^\circ = \left\{ \sum_{i=1}^{p} \mu_i \nabla h_i(x) + \sum_{i \in \mathcal{I}(x)} \lambda_i \nabla g_i(x) : \mu_i \in \mathbb{R}, \lambda_i \geq 0 \right\}, \tag{8.10}$$

where $\mathcal{I}(x)$ indexes the active inequality constraints at x, as in (8.7).

Proof. Denote the right-hand side of (8.10) by K: this is the same cone as in Lemma 8.20 with $u_1, \dots, u_p$ corresponding to $\nabla h_1(x), \dots, \nabla h_p(x)$ and $v_1, \dots, v_{m'}$ corresponding to the gradients of the $m' = |\mathcal{I}(x)|$ active inequality constraints. With this notation and recalling Definition 8.8 for $\mathrm{F}_x S$, notice that

$$v \in \mathrm{F}_x S \iff \langle u_i, v \rangle = 0 \text{ for all } i \text{ and } \langle v_i, v \rangle \leq 0 \text{ for all } i.$$

Therefore, the Farkas lemma states the following: given $w \in \mathcal{E}$, exactly one of the following statements is true:

1. Either $w \in K$, or
2. There exists $v \in \mathrm{F}_x S$ such that $\langle w, v \rangle > 0$.

However, the latter statement is equivalent to the claim that $w \notin (\mathrm{F}_x S)^\circ$. Let's write this out once more:

1. If $w \in K$, then it is not true that $w \notin (\mathrm{F}_x S)^\circ$, i.e., $w \in (\mathrm{F}_x S)^\circ$.
2. If $w \notin K$, then $w \notin (\mathrm{F}_x S)^\circ$.

We conclude that $K = (\mathrm{F}_x S)^\circ$. □

8.7 Karush–Kuhn–Tucker conditions

Combining the results of the last few sections, we obtain a key theorem known as the *KKT theorem*, named after Karush, Kuhn and Tucker. To state it, we first introduce a definition followed by simple facts about that definition.

Throughout this section, we consider the optimization problem

$$\min_{x\in\mathcal{E}} f(x) \text{ subject to } x \in S \tag{8.11}$$

where the set S is defined by equality and inequality constraints as

$$S = \Big\{ x \in \mathcal{E} : h_i(x) = 0,\ i = 1,\dots,p,\ \text{and} \\ g_i(x) \le 0,\ i = 1,\dots,m \Big\},$$

and $f\colon \mathcal{E} \to \mathbb{R}$, $h\colon \mathcal{E} \to \mathbb{R}^p$ and $g\colon \mathcal{E} \to \mathbb{R}^m$ are continuously differentiable.

Definition 8.22. *A point $x \in \mathcal{E}$ is a* KKT point *for (8.11) if it is feasible ($x \in S$) and there exist* Lagrange multipliers *$\mu \in \mathbb{R}^p$ and $\lambda \in \mathbb{R}^m$ with $\lambda \ge 0$ satisfying the* KKT conditions*:*

$$-\nabla f(x) = \sum_{i=1}^{p} \mu_i \nabla h_i(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x) \tag{8.12}$$

and

$$\lambda_i g_i(x) = 0 \quad \text{for} \quad i = 1,\dots,m. \tag{8.13}$$

The conditions (8.13) are also called complementarity conditions*.*

Lemma 8.23. *A point $x \in S$ is KKT if and only if $-\nabla f(x) \in (\mathrm{F}_x S)^\circ$.*

Proof. The complementarity conditions (8.13) express the fact that if $g_i(x) \neq 0$ then we must have $\lambda_i = 0$: this is a convenient way to state that the second sum in (8.12) must only include indices $1 \le i \le m$ for which $g_i(x) = 0$. With this observation in hand, apply Theorem 8.21 to conclude. □

The above lemma yields two corollaries.

Corollary 8.24. *If $x \in S$ is a KKT point, then it is a stationary point.*

Proof. The KKT conditions imply $-\nabla f(x) \in (\mathrm{F}_x S)^\circ$. Theorem 8.9 tells us that $\mathrm{T}_x S \subseteq \mathrm{F}_x S$, hence also that $(\mathrm{F}_x S)^\circ \subseteq (\mathrm{T}_x S)^\circ = \mathrm{N}_x S$ (do you see why?). Thus, the KKT conditions imply $-\nabla f(x) \in \mathrm{N}_x S$, i.e., x is stationary. □

Corollary 8.25. *If $x \in S$ is a stationary point **and** a constraint qualification condition holds at x, then x is a KKT point.*

Proof. The CQ implies that $\mathrm{T}_x S = \mathrm{F}_x S$. Thus, stationarity of x implies $-\nabla f(x) \in \mathrm{N}_x S = (\mathrm{T}_x S)^\circ = (\mathrm{F}_x S)^\circ$. It follows that x is a KKT point by Lemma 8.23. □

The above result can be further stated in the following main theorem, known as the KKT theorem. It is the motivation behind algorithms which aim to find KKT points, in the hope that these will be local minima.

Theorem 8.26 (KKT)**.** *If $x^* \in S$ is a local minimum **and** a constraint qualification condition holds at x^*, then x^* is a KKT point.*

Proof. Combine Theorem 7.20 and Corollary 8.25. □

Exercise 8.27. *Establish the following statements:*

1. *If LICQ holds at $x \in S$, then there exists at most one valid choice of Lagrange multipliers to satisfy the KKT conditions.*

2. *In contrast, without LICQ there might exist more than one valid choice of Lagrange multipliers.*

3. *The set of valid Lagrange multipliers at $x \in S$ is closed and convex (see Definition 9.1 for the notion of convex set).*

Remark 8.28. *This remark is based on notes by Jim Burke.*[2] *Notice that for the KKT theorem to hold it is sufficient to have $(\mathrm{T}_x S)^\circ = (\mathrm{F}_x S)^\circ$. We defined constraint qualifications as any conditions which ensure the (stronger) property $\mathrm{T}_x S = \mathrm{F}_x S$: this is called* Abadie CQ*. The somewhat weaker condition $(\mathrm{T}_x S)^\circ = (\mathrm{F}_x S)^\circ$ is called* Guignard CQ*. You can check on the following example that Guignard CQ may hold (hence the KKT theorem may apply) even if Abadie CQ fails:*

$$\min_{x \in \mathbb{R}^2} x_1^2 + x_2^2 \qquad s.t. \qquad x_1, x_2 \geq 0 \text{ and } x_1 x_2 = 0.$$

Upon drawing the feasible set, you will see that the most interesting feasible point is $x = (0, 0)$. In this course, we use "CQ" to mean Abadie CQ.

[2] https://sites.math.washington.edu/~burke/crs/516/notes/cq_lec.pdf; More: https://epubs.siam.org/doi/10.1137/1015075 (SIAM Review, Peterson, 1972)

8.8 Lagrangian duality

We continue within the same setting as Section 8.7. Consider the following function, called the *Lagrangian* of problem (8.11):

$$\begin{aligned} L &: \mathcal{E} \times \mathbb{R}^p \times \mathbb{R}^m_+ \to \mathbb{R} \\ &: (x, \mu, \lambda) \mapsto L(x, \mu, \lambda) = f(x) + \sum_{i=1}^{p} \mu_i h_i(x) + \sum_{i=1}^{m} \lambda_i g_i(x), \end{aligned} \tag{8.14}$$

where $\mathbb{R}^m_+ = \{\lambda \in \mathbb{R}^m : \lambda_i \geq 0 \text{ for } i = 1, \ldots, m\}$.

We define a *primal function* associated with problem (8.11) as follows:

$$L_P(x) = \sup_{\mu \in \mathbb{R}^p, \lambda \in \mathbb{R}^m_+} L(x, \mu, \lambda). \tag{8.15}$$

Formally, L_P is a function from $\mathcal{E}$ to the *extended reals* $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$. Indeed, it could very well be that the supremum evaluates to $+\infty$ for certain values of x. Case in point, it is easy to check (exercise) that

$$L_P(x) = \begin{cases} f(x) & \text{if } x \in S, \\ +\infty & \text{otherwise.} \end{cases} \tag{8.16}$$

Thus, the following problem is *equivalent* to (8.11)—we call it the *primal problem*:

$$\min_{x \in \mathcal{E}} L_P(x). \tag{primal}$$

Indeed, if S is non-empty then it is always better to pick $x \in S$ to minimize L_P; and if S is empty then it makes sense to assign the value $+\infty$ to the minimization problem (8.11).

We can also define the so-called *dual function* associated with (8.11):

$$L_D(\mu, \lambda) = \inf_{x \in \mathcal{E}} L(x, \mu, \lambda). \tag{8.17}$$

Formally, L_D is a function from $\mathbb{R}^p \times \mathbb{R}^m_+$ to $\bar{\mathbb{R}}$. This allows us to define the *dual problem*:

$$\max_{\mu \in \mathbb{R}^p, \lambda \in \mathbb{R}^m_+} L_D(\mu, \lambda). \tag{dual}$$

Notice how the (primal) and the (dual) problems are related by simply swapping the order of minimization over x and maximization over (μ, λ).

In general, the dual function L_D may be a complicated object: it is not always computable. Still, we can make the following observation that should motivate us to investigate more.

Theorem 8.29 (weak duality)**.** *For all $x \in \mathcal{E}$ and $(\mu, \lambda) \in \mathbb{R}^p \times \mathbb{R}^m_+$ it holds that $L_D(\mu, \lambda) \leq L_P(x)$. In particular,*

$$d^* \triangleq \left[\max_{\mu \in \mathbb{R}^p, \lambda \in \mathbb{R}^m_+} L_D(\mu, \lambda)\right] \leq \left[\min_{x \in \mathcal{E}} L_P(x)\right] = \left[\min_{x \in S} f(x)\right] \triangleq p^*.$$

Proof. Fix $\bar{x} \in \mathcal{E}, \bar{\mu} \in \mathbb{R}^p, \bar{\lambda} \in \mathbb{R}^m_+$ arbitrarily. We see that

$$L_D(\bar{\mu}, \bar{\lambda}) = \left[\inf_{x \in \mathcal{E}} L(x, \bar{\mu}, \bar{\lambda})\right] \leq L(\bar{x}, \bar{\mu}, \bar{\lambda}) \leq \left[\sup_{\mu \in \mathbb{R}^p, \lambda \in \mathbb{R}^m_+} L(\bar{x}, \mu, \lambda)\right] = L_P(\bar{x}),$$

as announced. Since the inequality holds at arbitrary points, it also holds at optimal points or along sequences that realize the extremal values. □

Why is Theorem 8.29 relevant? Consider the following issue: You run an optimization algorithm on (8.11) (equivalently, on the (primal)) and obtain a tentative solution $\bar{x} \in S$. How can you convince yourself (or someone else!) that $\bar{x}$ is, in fact, optimal? That is: how are we supposed to *certify* that $\bar{x}$ is optimal? To certify that $\bar{x}$ is *not* optimal, it "suffices" to exhibit a better point in S. But to certify that no such point exists is a much harder task. That is where the dual problem may help, as follows: Given any $(\bar{\mu}, \bar{\lambda}) \in \mathbb{R}^p \times \mathbb{R}^m_+$, Theorem 8.29 tells us that all $x \in S$ satisfy $f(x) \geq L_D(\bar{\mu}, \bar{\lambda})$. Therefore, if we can find $(\bar{\mu}, \bar{\lambda}) \in \mathbb{R}^p \times \mathbb{R}^m_+$ such that $f(\bar{x}) = L_D(\bar{\mu}, \bar{\lambda})$, we can be certain that $\bar{x}$ is a global optimum.

A reasonable objection here would be that, for some problems (e.g., under convexity and with a valid CQ), optimality can be certified with the KKT theorem. The added benefit of duality is that it may also be applied to certify *approximate* optimality. That is important in practice since algorithms almost never return an exact solution.

Still, there are two potential pitfalls to the story:

1. The weak duality theorem only says that the optimal value of the dual is *less than* the optimal value of the primal: it does not say that the two coincide. Therefore, there may not exist a $(\bar{\mu}, \bar{\lambda})$ satisfying $f(\bar{x}) = L_D(\bar{\mu}, \bar{\lambda})$ even if $\bar{x}$ is optimal. The difference between the optimal value of the primal problem and the optimal value of the dual problem is the *duality gap*.

2. The dual function L_D may be difficult to compute (let alone maximize).

The first pitfall is avoided if we are fortunate enough to have *strong duality*, that is, if the duality gap for our problem is zero. This notably happens under the assumptions laid out in the next theorem. Note the implicit role of

constraint qualifications (to ensure existence of Lagrange multipliers) and the explicit role of convexity (which we explore much more in the next chapter, especially Section 9.6).

Theorem 8.30 (strong duality)**.** *Assume both of the following:*

1. *The* (primal) *problem admits a KKT point $x^* \in S$ with valid Lagrange multipliers $\mu^* \in \mathbb{R}^p$ and $\lambda^* \in \mathbb{R}^m_+$, and*
2. *The function $x \mapsto L(x, \mu^*, \lambda^*)$ is convex.*

Then, we have strong duality*, namely:*

$$d^* = \left[\max_{\mu\in\mathbb{R}^p, \lambda\in\mathbb{R}^m_+} L_D(\mu, \lambda)\right] = \left[\min_{x\in\mathcal{E}} L_P(x)\right] = \left[\min_{x\in S} f(x)\right] = f(x^*) = p^*. \tag{8.18}$$

Moreover, (μ^*, λ^*) is optimal for the (dual) *problem and x^* is optimal for the* (primal) *problem as well as for*

$$\min_{x\in\mathcal{E}} L(x, \mu^*, \lambda^*)$$

which is an unconstrained *problem.*

Proof. Denote the gradient of $x \mapsto L(x, \mu^*, \lambda^*)$ by $\nabla_x L(\cdot, \mu^*, \lambda^*)$ (that is, the gradient of L with respect to x only, keeping μ^* and λ^* fixed). We have:

$$\nabla_x L(x, \mu^*, \lambda^*) = \nabla f(x) + \sum_{i=1}^{p} \mu_i^* \nabla h_i(x) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x). \tag{8.19}$$

It is easy to check (make sure you see why) that

$$L(x^*, \mu^*, \lambda^*) = f(x^*) \qquad \text{and} \qquad \nabla_x L(x^*, \mu^*, \lambda^*) = 0. \tag{8.20}$$

Since $x \mapsto L(x, \mu^*, \lambda^*)$ is convex by assumption, it follows from Corollary 4.23 that x^* is a minimizer of that function, and hence the minimal value of that function is $f(x^*)$.

Weak duality (Theorem 8.29) provides $d^* \le p^*$. It remains to show that $d^* \ge p^*$. Make sure you agree with each step below:

$$\begin{aligned} d^* &= \max_{\mu\in\mathbb{R}^p, \lambda\in\mathbb{R}^m_+} L_D(\mu, \lambda) \\ &\ge L_D(\mu^*, \lambda^*) \\ &= \inf_{x\in\mathcal{E}} L(x, \mu^*, \lambda^*) \\ &= L(x^*, \mu^*, \lambda^*) \\ &= f(x^*) \\ &\ge p^* \ge d^*. \end{aligned}$$

We conclude that all inequalities are equalities. □

Notice how the second part of the statement in Theorem 8.30 provides a nice interpretation for the Lagrange multipliers (μ^*, λ^*) at x^* in the stated context:

1. (μ^*, λ^*) provides a global optimum for the dual problem.
2. x^* is a global minimum for an *unconstrained* minimization problem where the cost function is $L(\cdot, \mu^*, \lambda^*)$, that is, $f(x)$ plus additional *penalty terms* $\mu_i^* h_i(x)$ for $i = 1, \dots, p$ and $\lambda_i^* g_i(x)$ for $i = 1, \dots, m$ such that $g_i(x^*) = 0$. Since there are no constraints, we might in principle benefit from violating the constraints. Yet, x^* (which satisfies the constraints) is a global minimum for $L(\cdot, \mu^*, \lambda^*)$. This reveals that, with the selected "weights" (μ^*, λ^*), there is no incentive to violate the constraints, because the cost of doing so would be offset by the penalty terms. Therefore, we can think of the Lagrange multipliers as the "currency exchange rate" between the cost function f and the constraint functions h_i and g_i.

We close this section with an observation that will be best understood in light of the next chapter: the (dual) problem is convex, in the sense that it is equivalent to $\min_{\mu \in \mathbb{R}^p, \lambda \in \mathbb{R}^m_+} -L_D(\mu, \lambda)$ where $-L_D$ is convex and the search space $\mathbb{R}^p \times \mathbb{R}^m_+$ (as we can verify through Definition 9.1) is a convex set.

Theorem 8.31. *The dual function L_D (8.17) is always concave. Thus, the (dual) problem is convex even if the (primal) problem is not convex.*

Proof. The search space of the (dual) problem, namely, $\mathbb{R}^p \times \mathbb{R}^m_+$ is convex. To see that L_D is concave, it is sufficient to notice that L_D is the infimum over a collection of affine (hence concave) functions. To build intuition, draw a handful of affine functions from $\mathbb{R}$ to $\mathbb{R}$ and highlight their minimum. □

Example 8.32 (Linear programming)**.** *One possible format to express linear programs is as:*

$$\min_{x \in \mathbb{R}^n} c^\top x \text{ subject to } Ax \geq b,$$

with $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. The corresponding Lagrangian function $L \colon \mathbb{R}^n \times \mathbb{R}^m_+ \to \mathbb{R}$ is

$$L(x, \lambda) = c^\top x + \lambda^\top (b - Ax) = (c - A^\top \lambda)^\top x + b^\top \lambda.$$

It is an exercise to verify that the dual function $L_D \colon \mathbb{R}^m_+ \to \bar{\mathbb{R}}$ is then

$$L_D(\lambda) = \inf_{x \in \mathcal{E}} L(x, \lambda) = \begin{cases} b^\top \lambda & \text{if } c - A^\top \lambda = 0, \\ -\infty & \text{otherwise.} \end{cases}$$

Therefore, the dual problem is

$$\max_{\lambda \in \mathbb{R}^m} b^\top \lambda \text{ subject to } A^\top \lambda = c \text{ and } \lambda \geq 0.$$

Notice that the dual problem is also a linear program: it consists in a linear cost function $\lambda \mapsto b^\top \lambda$ and linear equality ($A^\top \lambda = c$) and inequality ($\lambda \geq 0$) constraints. What can you say about strong duality (check the assumptions of Theorem 8.30)? It is an interesting exercise to compute the dual of the dual: what do you find?

Example 8.33 (Rayleigh quotient)**.** *Given a symmetric matrix $A \in \mathbb{R}^{n \times n}$, consider the following primal optimization problem:*

$$\min_{x \in \mathbb{R}^n} x^\top A x \quad \text{subject to} \quad h(x) = 1 - x^\top x = 0.$$

Depending on A, the cost function may or may not be convex. The constraint is not convex. The Lagrangian function is:

$$L(x, \mu) = x^\top A x + \mu(1 - x^\top x) = x^\top (A - \mu I_n) x + \mu,$$

where I_n is the identity matrix of size n. Notice that this function is bounded below exactly if $A - \mu I_n$ is positive semidefinite. Thus, the dual function satisfies

$$L_D(\mu) = \begin{cases} \mu & \text{if } A - \mu I_n \succeq 0, \\ -\infty & \text{otherwise.} \end{cases}$$

The condition $A - \mu I_n \succeq 0$ holds if and only if $\mu \leq \lambda_{\min}(A)$. Accordingly, the dual problem can be stated as:

$$\max_{\mu \in \mathbb{R}} \mu \quad \text{subject to} \quad \mu \leq \lambda_{\min}(A).$$

It is clear that the maximal value of the dual problem is $\mu = \lambda_{\min}(A)$. Weak duality therefore implies that the minimal value of the primal is lower-bounded by $\lambda_{\min}(A)$. If we let x be a unit-norm eigenvector of A associated to its smallest eigenvalue, we also find that $f(x) = x^\top A x = \lambda_{\min}(A)$. Therefore, the primal and the dual problems have the same optimal value: strong duality holds and x is optimal for the primal.

We could also have tried to determine whether or not strong duality holds by considering the assumptions of Theorem 8.30:

1. *The primal admits a minimizer because the cost function is continuous and the search space is compact.*

2. *Constraint qualifications hold everywhere since LICQ holds at all feasible points. Thus, any minimizer x^* has a Lagrange multiplier μ^*.*

3. *The function $x \mapsto L(x, \mu^*)$ is convex if and only if $A - \mu^* I_n \succeq 0$, that is, if and only if $\mu^* \leq \lambda_{\min}(A)$.*

It is not obvious a priori whether or not the last condition above holds. Fortunately, we have seen already that for this (very special) problem, the condition works out favorably.

Example 8.34 (Solution of minimum norm)**.** *Consider the following primal problem:*

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|x\|^2 \text{ subject to } Ax = b,$$

where $A \in \mathbb{R}^{p \times n}$ and $b \in \mathbb{R}^p$ are given and we assume the feasible set is non-empty. The Lagrangian function is $L \colon \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ with:

$$L(x, \mu) = \frac{1}{2}\|x\|^2 + \mu^\top (b - Ax).$$

Consider the assumptions of Theorem 8.30:

1. *The primal consists in finding the point of the affine subspace $\{x \in \mathbb{R}^n : Ax = b\}$ closest to the origin: intuitively, it is clear that this problem has a minimizer; you should be able to make this intuition precise.*

2. *Since the constraints are affine, constraint qualifications hold everywhere. Therefore, any minimizer has Lagrange multipliers.*

3. *L is convex in x for all values of μ.*

We conclude that strong duality holds for this problem.

The primal function is easily determined from (8.16):

$$L_P(x) = \begin{cases} \frac{1}{2}\|x\|^2 & \text{if } Ax = b, \\ +\infty & \text{otherwise.} \end{cases}$$

To get a hold of the dual function, we need to work harder. By definition,

$$L_D(\mu) = \inf_{x \in \mathbb{R}^n} \frac{1}{2}\|x\|^2 + \mu^\top (b - Ax).$$

Notice that the function we must "infimize" is a strongly convex quadratic function of x: we know that it has a unique minimizer. Moreover, that

minimizer is the critical point of the quadratic. The gradient of $x \mapsto \frac{1}{2}\|x\|^2 + \mu^\top(b - Ax)$ *is* $x - A^\top\mu$. *This is zero exactly when* $x = A^\top\mu$. *Plugging this into the expression for* L_D *above yields:*

$$\begin{aligned} L_D(\mu) &= \frac{1}{2}\|A^\top\mu\|^2 + \mu^\top(b - AA^\top\mu) \\ &= -\frac{1}{2}\mu^\top AA^\top\mu + b^\top\mu. \end{aligned}$$

As expected, L_D *is concave; moreover, it is quadratic.*

Let us solve the dual problem, that is, let us maximize $L_D(\mu)$ *for* $\mu \in \mathbb{R}^p$ *unconstrained. This is equivalent to minimizing the convex quadratic* $\frac{1}{2}\mu^\top AA^\top\mu - b^\top\mu$, *whose critical points are characterized by*

$$AA^\top\mu - b = 0.$$

All $\mu \in \mathbb{R}^p$ *which satisfy the above are global optima for the dual. Let* μ^* *be one such solution. Motivated by Theorem 8.30, we consider the following unconstrained optimization problem:*

$$\min_{x\in\mathbb{R}^n} L(x, \mu^*) \qquad \text{where} \qquad L(x, \mu^*) = \frac{1}{2}\|x\|^2 + (\mu^*)^\top(b - Ax).$$

This is again a strongly convex quadratic in x. *As per our work above, its unique minimizer* x^* *is given by*

$$x^* = A^\top\mu^*.$$

Notice that $Ax^* = AA^\top\mu^* = b$, *i.e.,* x^* *is feasible for the primal even though we did not enforce it explicitly. Moreover,*

$$L_D(\mu^*) = -\frac{1}{2}(\mu^*)^\top AA^\top\mu^* + b^\top\mu^* = \frac{1}{2}(\mu^*)^\top AA^\top\mu^* = \frac{1}{2}\|A^\top\mu^*\|^2 = f(x^*).$$

Therefore, strong duality confirms that x^* *is optimal for the primal and* μ^* *is optimal for the dual: they certify each other's optimality.*

Example 8.35 (Max-Cut)**.** *Given a symmetric matrix* $W \in \mathbb{R}^{n\times n}$, *consider the following binary optimization problem:*

$$\min_{x\in\mathbb{R}^n} x^\top Wx \text{ subject to } x_i \in \{-1, +1\} \text{ for } i = 1, \ldots, n.$$

The cost function expands as

$$f(x) = x^\top Wx = \sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}x_ix_j.$$

In other words: if x_i, x_j have the same sign, then we "pay" W_{ij}, whereas if x_i, x_j have opposite signs then we "pay" $-W_{ij}$. Think about the meaning of this if W or $-W$ is the adjacency matrix of an undirected graph (look-up the Max-Cut problem). This is a computationally hard problem to solve.

Stated as above, the problem does not fit in our framework. However, notice that we can restate each constraint $x_i \in \{-1, +1\}$ as $1 - x_i^2 = 0$. Thus, the above problem is equivalent to the following primal problem:

$$\min_{x \in \mathbb{R}^n} x^\top W x \text{ subject to } h_i(x_i) = 0 \text{ for } i = 1, \dots, n,$$

where each $h_i \colon \mathbb{R} \to \mathbb{R}$ is defined by $h_i(t) = 1 - t^2$. The problem is not convex.

The Lagrangian function is given by:

$$\begin{aligned} L(x, \mu) &= x^\top W x + \sum_{i=1}^{n} \mu_i (1 - x_i^2) \\ &= x^\top W x - x^\top \mathrm{diag}(\mu) x + \mathbf{1}^\top \mu \\ &= x^\top (W - \mathrm{diag}(\mu)) x + \mathbf{1}^\top \mu, \end{aligned}$$

where $\mathbf{1} \in \mathbb{R}^n$ is the vector of all ones and $\mathrm{diag}(\mu)$ is the $n \times n$ diagonal matrix whose diagonal entries are $\mu_1, \dots, \mu_n$. Notice that $x \mapsto L(x, \mu)$ is a convex quadratic if $W - \mathrm{diag}(\mu)$ is positive semidefinite, and that it is unbounded below otherwise. It follows that the dual function satisfies:

$$L_D(\mu) = \begin{cases} \mathbf{1}^\top \mu & \text{if } W - \mathrm{diag}(\mu) \succeq 0, \\ -\infty & \text{otherwise.} \end{cases}$$

As a result, the dual problem can be stated as:

$$\max_{\mu \in \mathbb{R}^n} \mathbf{1}^\top \mu \text{ subject to } W - \mathrm{diag}(\mu) \succeq 0.$$

We do not usually have strong duality for the combinatorial problem. However, weak duality still holds. Therefore, any $\mu \in \mathbb{R}^n$ such that $\mathrm{diag}(\mu) \preceq W$ can be used to provide a lower-bound $\mathbf{1}^\top \mu$ on the optimal value of the combinatorial problem. Such bounds can be tremendously useful for the design of algorithms (e.g., branch-and-bound algorithms)—this is beyond the scope of our course though.

Exercise 8.36. *Verify formula (8.16) for the primal function.*

Exercise 8.37 (Quadratic programming)**.** *Work out the dual of the* quadratic program

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^\top H x + c^\top x \text{ subject to } Ax \geq b,$$

where H (symmetric) is positive definite. What can you say about strong duality? It is interesting (but more difficult) to work this out assuming only that H is positive semidefinite.

Chapter 9

Convex constraints

In Chapter 4, we discussed the concept of convex functions. We discovered how convexity in unconstrained optimization ensures that all local minima are in fact global minima, thus making our life as an optimizer much simpler. In this chapter, we define a notion of convexity for the search space S. The impetus for this chapter is the fact that if *both* the search space S and the cost function f are convex, then it is still true that local minima are global minima. This is a generalization of our earlier results because, as will be clear in a moment, linear spaces $\mathcal{E}$ are convex.

We can state the main result right away. With the remainder of this chapter, we shall explore its ramifications. **Throughout this chapter, as per usual, we assume f is differentiable.**

Definition 9.1. *A set $S \in \mathcal{E}$ is* convex *if the following holds:*

$$x, y \in S \quad \Longrightarrow \quad (1-t)x + ty \in S \text{ for all } t \in [0,1].$$

In words: S is convex if it contains the line segment connecting each pair of points in S.

Theorem 9.2. *Consider the constrained minimization problem*

$$\min_{x \in \mathcal{E}} f(x) \text{ subject to } x \in S.$$

If S is convex and *f is differentiable and convex, then the (first-order) necessary optimality conditions are also sufficient for global optimality, that is:*

$$x^* \in S \text{ is a global minimum} \qquad \Longleftrightarrow \qquad -\nabla f(x^*) \in \mathrm{N}_{x^*} S.$$

In particular, all local minima are global minima.

Proof. We already know from Theorem 7.20 that if x^* is a global minimum then $-\nabla f(x^*) \in \mathrm{N}_{x^*}S$. Let us use convexity to show the converse. Assume $x^* \in S$ satisfies $-\nabla f(x^*) \in \mathrm{N}_{x^*}S$. We must show that x^* is a global minimum of f in S. For contradiction, assume there exists $x \in S$ such that $f(x) < f(x^*)$. By convexity of S, it holds that $c(t) \triangleq x^* + t(x - x^*)$ is in S for all $t \in [0, 1]$. Therefore, $c'(0) = x - x^*$ belongs to the tangent cone $\mathrm{T}_{x^*}S$. Our assumption $-\nabla f(x^*) \in \mathrm{N}_{x^*}S$ therefore implies that

$$\langle -\nabla f(x^*), x - x^* \rangle \leq 0.$$

Moreover, convexity of f implies that

$$f(x) \geq f(x^*) + \langle \nabla f(x^*), x - x^* \rangle .$$

Combining the two last statements reveals $f(x) \geq f(x^*)$: a contradiction.

The last statement holds because we have in fact shown that local minima, global minima and stationary points coincide. □

Because convexity has such profound ramifications for optimization, we also define the notion of convexity for an optimization problem.

Definition 9.3. *A minimization problem as in Theorem 9.2 is called a* convex optimization problem *if both S and f are convex.*

Exercise 9.4. *Mind the following: in Theorem 9.2, it is important to check* all three *of the following boxes:*

1. *We are* minimizing *(and not maximizing).*
2. *The cost function f is convex.*
3. *The search space S is convex.*

To really appreciate this fact, do the following:

1. *Give three examples of optimization problems which check two of the above but not all three of the above, and for which there exists a non-optimal stationary point.*
2. *If given a maximization problem, explain how you can get an equivalent minimization problem. What is a good equivalent of "convex optimization problem" for a maximization problem?*
3. *If the cost function is not convex, explain how you can get an equivalent problem with a convex cost function (you can even make the cost function linear).*

4. *If the constraint set is not convex, explain how you can get an equivalent problem with a convex search space (you can even make the problem unconstrained)—you will need to make the cost function a bit weird for this though.*

For each of the above, explain how we should understand Theorem 9.2 against the modified problem, specifically to verify that, sadly, there is no free lunch.

Exercise 9.5. *Consider a function $f\colon \mathcal{E} \to \mathbb{R}$. The* epigraph *of f is the set*

$$\{(x, s) \in \mathcal{E} \times \mathbb{R} : s \geq f(x)\}. \tag{9.1}$$

Show that this set is convex if and only if f is convex. It is instructive to draw an example with $\mathcal{E} = \mathbb{R}$.

9.1 Convex sets and their cones

The tangent cones to an arbitrary set are defined in a somewhat abstract way: recall Definition 7.7. For convex sets, the tangent cones can be described in a more straightforward way as we now show. A consequence is that the normal cones to a convex set also admit a particularly nice description.

Definition 9.6. *Given a point x in a set $S \subseteq \mathcal{E}$, we let*

$$\mathrm{K}_x S = \{\alpha \cdot (y - x) : y \in S, \alpha \geq 0\} \tag{9.2}$$

denote the cone of feasible directions.

Given a set A, we write $\overline{A}$ to denote the closure of A, that is, the set of all limits of sequences in A.

Theorem 9.7. *Let $S \subseteq \mathcal{E}$ be a convex set. Then, the tangent cone to S at x is the closure of the cone of feasible directions:*

$$\mathrm{T}_x S = \overline{\mathrm{K}_x S}. \tag{9.3}$$

Consequently, the normal cone at x is:

$$\mathrm{N}_x S = \{v \in \mathcal{E} : \langle v, y - x\rangle \leq 0 \text{ for all } y \in S\}. \tag{9.4}$$

Proof. The proof of (9.3) is in three steps:

1. Let us show that $\mathrm{K}_x S \subseteq \mathrm{T}_x S$. To this end, consider $v \in \mathrm{K}_x S$ and the associated curve $c(t) = x + tv$. We know that $v = \alpha \cdot (y - x)$ for some $y \in S$ and $\alpha \geq 0$. If $\alpha = 0$, then $v = 0$ and indeed $0 \in \mathrm{T}_x S$. Assume $\alpha > 0$. Then, $c(0) = x$ and $c(1/\alpha) = y$. By convexity of S, we deduce that $c(t) \in S$ for all $t \in [0, 1/\alpha]$. It follows that $c'(0) = v$ is in $\mathrm{T}_x S$.

2. Since $\mathrm{T}_x S$ is closed, the inclusion $\mathrm{K}_x S \subseteq \mathrm{T}_x S$ implies $\overline{\mathrm{K}_x S} \subseteq \mathrm{T}_x S$.

3. It remains to show that $\mathrm{T}_x S \subseteq \overline{\mathrm{K}_x S}$. Given $v \in \mathrm{T}_x S$, let (x_k) be a sequence in S convergent to x and (t_k) be a sequence of positive reals convergent to zero such that $v = \lim_{k\to\infty} \frac{x_k - x}{t_k}$. For each k, we have $\frac{1}{t_k}(x_k - x) \in \mathrm{K}_x S$. Therefore, v is the limit of a sequence of points in $\mathrm{K}_x S$, that is, $v \in \overline{\mathrm{K}_x S}$. (We did not use convexity of S here.)

It is an exercise to establish (9.4). □

Combining Theorems 9.2 and 9.7, we get the following corollary.

Corollary 9.8. *Consider the constrained minimization problem*

$$\min_{x\in\mathcal{E}} f(x) \text{ subject to } x \in S.$$

If S is convex and *f (differentiable) is convex, then:*

$$x^* \in S \text{ is a global minimum} \iff \langle \nabla f(x^*), x - x^* \rangle \geq 0 \text{ for all } x \in S.$$

Exercise 9.9. *Let S be a convex set. Show that $\mathrm{K}_x S$ (9.2) is a convex cone. Deduce that $\mathrm{T}_x S$ is a convex cone. In contrast, show that if S is not convex then it is possible for both $\mathrm{K}_x S$ and $\mathrm{T}_x S$ not to be convex, and it is possible to have $\mathrm{T}_x S \neq \overline{\mathrm{K}_x S}$.*

Exercise 9.10. *Let C be a cone (not necessarily convex). Show that C° and C^* are convex. Deduce that $\mathrm{N}_x S$ is always a closed convex cone, even if S is not convex.*

Exercise 9.11. *Let C be a cone (not necessarily convex) and let $\overline{C}$ denote its closure. Show that their polars are the same: $C^\circ = (\overline{C})^\circ$. Use this to verify (9.4) in Theorem 9.7.*

Exercise 9.12. *Consider the closed unit ball in $\mathbb{R}^n$:*

$$S = \{x \in \mathbb{R}^n : \|x\| \leq 1\}.$$

This set is convex of course. Fix some $x \in S$ with $\|x\| = 1$. What is $\mathrm{K}_x S$ in this case? What is $\mathrm{T}_x S$? This exercise should convince you that it is indeed necessary to take the closure of $\mathrm{K}_x S$ in the statement $\mathrm{T}_x S = \overline{\mathrm{K}_x S}$ from (9.3).

Exercise 9.13. *Prove Corollary 9.8 more directly, without normal cones.*

9.2 Global minima of convex problems

In Section 4.4, we started a discussion on the sets of global minima of convex functions. We can now continue this discussion using the concept of convex set. As was already the case in the unconstrained case, in general, the set of global minima of a convex optimization problem could be empty, it could be a singleton and it could contain more than one point. In all cases, that set is convex.

Theorem 9.14. *If $f\colon \mathcal{E} \to \mathbb{R}$ is convex and $S \subseteq \mathcal{E}$ is convex, then the set of global minima of $\min_{x\in S} f(x)$ is a (possibly empty) convex set.*

Proof. Assume $x, y \in S$ are global minima of f on S, so that $f(x) = f(y) = f_{\min}$ and $f(z) \geq f_{\min}$ for all $z \in S$. Then, for all $t \in [0, 1]$ we know that $(1-t)x + ty$ is in S and therefore:

$$f_{\min} \leq f((1-t)x + ty) \leq (1-t)f(x) + tf(y) = f_{\min},$$

where the first inequality holds because $f_{\min}$ is the minimal value of f on S and the second inequality holds because f is convex. Thus, $(1-t)x + ty$ is a global minimum as well. □

We can also extend the results of Theorems 4.27 and 4.28.

Theorem 9.15. *Let $S \subseteq \mathcal{E}$ be convex. If $f\colon \mathcal{E} \to \mathbb{R}$ is strictly convex, then there exists at most one global minimum $x^* \in S$ for $\min_{x\in S} f(x)$.*

Proof. For contradiction, assume $x, y \in S$ are distinct global minima of f on S, so that $f(x) = f(y) = f_{\min}$ and $f(z) \geq f_{\min}$ for all $z \in S$. Then, for all $t \in (0, 1)$ we know that $(1-t)x + ty$ is in S and we have

$$f_{\min} \leq f((1-t)x + ty) < (1-t)f(x) + tf(y) = f_{\min},$$

where the first inequality holds because $f_{\min}$ is the minimal value of f on S and the second inequality holds because f is strictly convex. This is a contradiction. □

Theorem 9.16. *Let $S \subseteq \mathcal{E}$ be non-empty, closed and convex. If $f\colon \mathcal{E} \to \mathbb{R}$ is strongly convex, then there exists exactly one global minimum $x^* \in S$ for $\min_{x\in S} f(x)$.*

Proof. By Theorem 9.15, we know f admits at most one global minimum on S. It remains to show that it also admits at least one global minimum. This can be done with slight modifications to the proof sketch of Theorem 4.28 if we assume f is differentiable at (at least one) point $x \in S$. The statement holds even without making that assumption explicitly: we omit the proof. □

Exercise 9.17. *Show that the projection of a point $z \in \mathcal{E}$ to a closed and non-empty convex set $S \subseteq \mathcal{E}$ exists and is unique. (See also Exercise 7.31)*

Exercise 9.18 (Separation theorem)**.** *Let S be a closed and non-empty convex set in $\mathcal{E}$. Consider a point $z \notin S$. Show that there exists a hyperplane separating S and z, in the sense that z is on one side of the plane and S is on the other side. Hint: use Exercise 9.17.*

Exercise 9.19 (Separation theorem, bis)**.** *Now let S_1, S_2 be two closed and non-empty convex sets in $\mathcal{E}$ such that $S_1 \cap S_2 = \emptyset$. Show that there exists a hyperplane separating S_1 and S_2. (Hint: argue the existence of $x_1 \in S_1$ and $x_2 \in S_2$ that are as close to each other as possible; reason as in Exercise 9.18.)*

Remark 9.20. *Let us note a subtlety here regarding Theorem 9.16. (This is a side-note that won't be a concern in this introductory course: ignore it if you find it confusing.) It is important that f should be continuous. That is indeed the case in the theorem as stated, because convex, real-valued functions on a linear space are continuous [Roc70, Cor. 10.1.1]. However, consider this function from* `math.stackexchange.com/questions/231335`*:*

$$f\colon [0,1] \to \mathbb{R}\colon x \mapsto f(x) = \begin{cases} x^2 & \text{if } x \in (0,1], \\ 1 & \text{if } x = 0. \end{cases}$$

On the (non-empty, closed, convex) set $S = [0,1] \subset \mathbb{R}$, that function is strongly convex (indeed, $x \mapsto f(x) - x^2$ is convex on $[0,1]$). Yet, notice that f is discontinuous and that it does not have a minimizer.

9.3 Convexity through (in)equality constraints

Recall from Chapter 8 the special case of interest where the search space S is defined through equality and inequality constraints (8.1):

$$S = \Big\{ x \in \mathcal{E} : h_i(x) = 0,\ i = 1, \dots, p, \text{ and } \\ g_i(x) \leq 0,\ i = 1, \dots, m \Big\}. \tag{9.5}$$

Equivalently, we can define S as an intersection of sets, each defined by a single equality or inequality constraint:

$$S = \{x \in \mathcal{E} : h_1(x) = 0\} \cap \cdots \cap \{x \in \mathcal{E} : h_p(x) = 0\} \\ \cap \{x \in \mathcal{E} : g_1(x) \leq 0\} \cap \cdots \cap \{x \in \mathcal{E} : g_m(x) \leq 0\}. \tag{9.6}$$

This is instructive in view of the following theorem (the proof is an exercise):

Theorem 9.21. *If S_1 and S_2 are two convex sets, then their intersection $S_1 \cap S_2$ is convex. By extension, the intersection of any collection of convex sets is convex.*

From Theorem 9.21, we see that S is convex in particular if each one of the $p + m$ sets in (9.6) is convex. This is easily assessed, as we now see.

Theorem 9.22. *Given $g\colon \mathcal{E} \to \mathbb{R}$, the set $S = \{x \in \mathcal{E} : g(x) \leq 0\}$ is convex if g is convex.*

Proof. If g is convex and $x, y \in S$, then for all $t \in [0, 1]$ we have

$$g((1-t)x + ty) \leq (1-t)g(x) + tg(y) \leq 0,$$

hence $(1-t)x + ty$ is in S. Thus, S is convex. □

Theorem 9.23. *Given $h\colon \mathcal{E} \to \mathbb{R}$, the set $S = \{x \in \mathcal{E} : h(x) = 0\}$ is convex if h is affine, that is $h(x) = \langle w, x\rangle + b$ for some $w \in \mathcal{E}$ and $b \in \mathbb{R}$.*

Proof. There are many direct ways to show this. Here is a less direct but instructive proof: S can be defined through the inequality constraints $h(x) \leq 0$ and $-h(x) \leq 0$; if h is affine, then both h and $-h$ are convex, hence S is the intersection of two convex sets, which itself is convex. □

The results of this section can be summarized as follows.

Corollary 9.24. *The set S defined in (9.5) is convex if the equality constraint functions $h_1, \ldots, h_p$ are affine and the inequality constraint functions $g_1, \ldots, g_m$ are convex.*

Example 9.25. *Since affine functions (and their negative) are convex, it readily follows that minimizing (or maximizing) an affine function under affine equality and inequality constraints is a convex optimization problem. This is called* linear programming. *There is a lot to say about it: see [NW06, Ch. 13] for more.*

Exercise 9.26. *Prove Theorem 9.21.*

Exercise 9.27. *Show through an example that the conclusion of Theorem 9.22 may fail if g is not convex.*

Exercise 9.28. *Show through an example that the conclusion of Theorem 9.23 may fail if h is not affine, even if h is convex.*

Exercise 9.29. *Let S be a set (not necessarily convex) defined by equality and inequality constraints. Show that $\mathrm{F}_x S$ (Definition 8.8) is always a closed convex cone.*

9.4 Slater's constraint qualification

Recall the general notion of constraint qualification conditions (CQ) we discussed in Section 8.5. For a search space S defined through equality and inequality constraints, we introduced the notion of cone of linearized feasible directions $\mathrm{F}_x S$ (Definition 8.8), and we stated that CQs hold at x if $\mathrm{T}_x S = \mathrm{F}_x S$. The CQs we discussed back then (LICQ, MFCQ, linear CQ) apply with wide generality. In particular, they still apply for convex optimization problems.

In the special case where the equality constraints are affine and the inequality constraints are convex, we know that S is convex. As a result, Theorem 9.7 further tells us that $\mathrm{T}_x S = \overline{\mathrm{K}_x S}$. Therefore, in this special case, CQs hold if $\overline{\mathrm{K}_x S} = \mathrm{F}_x S$. We exploit this to define a new CQ specifically for such convex optimization problems, by the name of *Slater's condition*. It is often nicer to check than more general CQs such as LICQ and MFCQ because it provides a result for *all* feasible points at once.

Theorem 9.30. *Let S be defined by equality constraints $h_i(x) = 0$ for $i = 1, \ldots, p$ and inequality constraints $g_i(x) \leq 0$ for $i = 1, \ldots, m$ where each h_i is affine and each g_i is convex and continuously differentiable. If there exists a feasible point $x_{\mathrm{s}} \in S$ which satisfies all inequalities strictly, that is, $g_i(x_{\mathrm{s}}) < 0$ for $i = 1, \ldots, m$, then $\mathrm{T}_x S = \mathrm{F}_x S$ for* all *$x \in S$ and we say* Slater's constraint qualification condition *holds.*

Proof. Fix an arbitrary point $x \in S$. We know from the general case that $\mathrm{T}_x S \subseteq \mathrm{F}_x S$—see Theorem 8.9. Theorem 9.7 provides $\mathrm{T}_x S = \overline{\mathrm{K}_x S}$. Thus, we only need to show that $\mathrm{F}_x S \subseteq \overline{\mathrm{K}_x S}$. To do so, pick an arbitrary $v \in \mathrm{F}_x S$, that is, $v \in \mathcal{E}$ satisfies

$$\begin{aligned} &\langle \nabla h_i(x), v \rangle = 0, \ \ i = 1, \ldots, p, \ \text{and} \\ &\langle \nabla g_i(x), v \rangle \leq 0, \ \ i = 1, \ldots, m \ \text{such that} \ g_i(x) = 0. \end{aligned}$$

We need to construct a sequence of directions $(v_k)_{k \geq 0}$ in $\mathrm{K}_x S$ such that $v_k \to v$. Indeed, that will show that $v \in \overline{\mathrm{K}_x S}$, as desired. To this end, define

$$v_k = (1 - \alpha_k) v + \alpha_k (x_{\mathrm{s}} - x)$$

with some sequence $(\alpha_k)_{k \geq 0}$ of nonnegative real numbers satisfying $\alpha_k \to 0$. We are about to show that

$$\alpha_k \in (0, 1) \implies v_k \in \mathrm{K}_x S.$$

Then, we can conclude by setting $\alpha_k = \frac{1}{2+k}$ for example.

Let us establish the remaining piece. Consider the curve

$$c(t) = x + tv_k,$$

assuming $\alpha_k \in (0, 1)$. For all $t > 0$ we have

$$v_k = \frac{1}{t}(c(t) - x).$$

Thus, if $c(t)$ is in S for some $t > 0$, it follows that v_k is in $\mathrm{K}_x S$. Let us prove that this is indeed the case. To see it, first assess the equality constraints against $c(t)$. Since each h_i is affine, we have $h_i(y) = \langle w_i, y\rangle + b_i$ for some $w_i \in \mathcal{E}$ and $b_i \in \mathbb{R}$. Thus,

$$\begin{aligned} h_i(c(t)) &= \langle w_i, x + tv_k\rangle + b_i \\ &= \langle w_i, x\rangle + b_i + t\,\langle w_i, v_k\rangle \\ &= h_i(x) + t\,\langle w_i, (1-\alpha_k)v + \alpha_k(x_{\mathrm{s}} - x)\rangle \\ &= 0. \end{aligned}$$

Above, the last equality follows from $h_i(x) = 0$ (since $x \in S$), $\langle w_i, v\rangle = \langle \nabla h_i(x), v\rangle = 0$ (since $v \in \mathrm{F}_x S$) and $\langle w_i, x_{\mathrm{s}} - x\rangle = 0$ (since both x_{s} and x are feasible). We have established that the equality constraints are satisfied for $c(t)$ with all t. Let us now assess the inequality constraints against $c(t)$. Since $c(0) = x$ and g_i is continuously differentiable, we have:

$$g_i(c(t)) = g_i(x) + \langle \nabla g_i(x), c(t) - x\rangle + o(\|c(t) - x\|).$$

Below, we consider each $1 \le i \le m$ in turn.

If $g_i(x) < 0$, then by continuity it is certainly possible to pick $t_i > 0$ small enough such that $g_i(c(t)) \le 0$ for all $t \in [0, t_i]$.

If $g_i(x) = 0$, then we can further write

$$\begin{aligned} g_i(c(t)) &= t\,\langle \nabla g_i(x), v_k\rangle + o(t) \\ &= t(1-\alpha_k)\,\langle \nabla g_i(x), v\rangle + t\alpha_k\,\langle \nabla g_i(x), x_{\mathrm{s}} - x\rangle + o(t) \\ &\le t\alpha_k\,\langle \nabla g_i(x), x_{\mathrm{s}} - x\rangle + o(t), \end{aligned}$$

where in the last step we used $\langle \nabla g_i(x), v\rangle \le 0$ owing to $v \in \mathrm{F}_x S$. It remains to show that

$$\langle \nabla g_i(x), x_{\mathrm{s}} - x\rangle < 0. \tag{9.7}$$

Indeed, once (9.7) is established, we can claim that there exists $t_i > 0$ small enough such that $g_i(c(t)) \le 0$ for all $t \in [0, t_i]$. Then, we can conclude the proof by setting $\bar{t} = \min_{1 \le i \le m} t_i$ which ensures $c(t) \in S$ for all $t \in [0, \bar{t}]$.

To see that (9.7) holds, we use (a) the Slater condition $g_i(x_s) < 0$, (b) convexity of g, and (c) $g_i(x) = 0$:

$$0 > g_i(x_s) \geq g_i(x) + \langle \nabla g_i(x), x_s - x \rangle = \langle \nabla g_i(x), x_s - x \rangle .$$

This concludes the proof. □

Exercise 9.31. *Show that Slater's condition implies MFCQ at all feasible points. (This is another way to prove that Slater's condition is a CQ for the whole feasible set.)*

Exercise 9.32. *Show the small changes that are needed in the proof of Slater's CQ theorem to see that we only need to require $g_i(x_s) < 0$ for i such that g_i is not affine. In other words: if some of the inequality constraints are affine, then it's fine to pick x_s in a way that those constraints are active.*

9.5 KKT for convex optimization problems

Recall the general KKT results from Section 8.7. Below, we write a special version of that theorem for the case of affine equality constraints and convex inequality constraints, summarizing much of what we learned in this chapter. The parts that are new compared to Theorem 8.26 are emphasized in bold.

Theorem 9.33 (KKT convex)**.** *Consider the optimization problem*

$$\min_{x \in \mathcal{E}} f(x) \text{ subject to } x \in S$$

where the set S is defined by equality and inequality constraints as

$$S = \Big\{ x \in \mathcal{E} : h_i(x) = 0,\ i = 1, \dots, p, \text{ and } g_i(x) \leq 0,\ i = 1, \dots, m \Big\},$$

and $f \colon \mathcal{E} \to \mathbb{R}$, $h \colon \mathcal{E} \to \mathbb{R}^p$ and $g \colon \mathcal{E} \to \mathbb{R}^m$ are continuously differentiable. ***Assume moreover that f and $g_1, \dots, g_m$ are convex, and that $h_1, \dots, h_p$ are affine. In particular, S is convex.***

The following statements hold (recall Definition 8.22 for KKT points):

1. *If the KKT conditions hold at $x^* \in S$, then x^* is a* ***global minimum.***
2. *If constraint qualification conditions hold at $x^* \in S$ and x^* is a local* ***(hence also global)*** *minimum, then the KKT conditions hold at x^*.*

Proof. Owing to convexity, Theorem 9.2 tells us that stationary points, local minima and global minima coincide. Then, the first statement follows from Corollary 8.24 and the second statement follows from Corollary 8.25. □

We get a particularly convenient corollary as a special case. This is one of the most important results in optimization.

Corollary 9.34. *In the setting of Theorem 9.33, if CQs hold at all feasible points (e.g., if Slater's condition holds or if all constraints are affine), then KKT conditions are necessary and sufficient for global optimality.*

9.6 Duality

Recall the notion of Lagrangian duality we defined in Section 8.8. It is particularly fruitful in the convex setting.

Corollary 9.35. *In the setting of Theorem 9.33, if the* (primal) *problem admits a global minimizer x^* and a constraint qualification condition holds at x^*, then strong duality holds and the Lagrange multipliers for the* (primal) *are optimal for the* (dual)*.*

Proof. CQs guarantee the existence of Lagrange multipliers μ^*, λ^* at x^* through Theorem 8.26 (KKT). In the proposed setting, $x \mapsto L(x, \mu^*, \lambda^*)$ (with L as in (8.14)) is convex without any conditions on μ^*, λ^*. Therefore, Theorem 8.30 applies. □

Slater's condition permits a further simplified statement, continued from Corollary 9.34.

Corollary 9.36. *In the setting of Theorem 9.33, if CQs hold at all feasible points (e.g., if Slater's condition holds or if all constraints are affine) and if the* (primal) *problem admits a global minimizer, then strong duality holds and the Lagrange multipliers for the* (primal) *are optimal for the* (dual)*.*

The above results provide a crisp interpretation for Lagrange multipliers: in the convex setting, under the stated assumptions, we can use Lagrange multipliers to *certify* optimality.

9.7 Conic programming

Consider an optimization problem of the following form:

$$\min_{x \in \mathcal{E}} \langle c, x \rangle \text{ subject to } Ax = b \text{ and } x \in C, \tag{CP}$$

where

- The cost vector $c \in \mathcal{E}$ defines a linear cost function $f(x) = \langle c, x \rangle$.
- The linear map $A\colon \mathcal{E} \to \mathbb{R}^p$ and the vector $b \in \mathbb{R}^p$ describe p equality constraints.
- The **closed, convex cone** C defines a *conic constraint*.

Problems of the form (CP) are called *conic programs*. Notice that they are convex optimization problems.

If the cone C is easily described through equality and inequality constraints, then we can express the feasible set $S = \{x \in C : Ax = b\}$ in the usual format as $\{x \in \mathcal{E} : h(x) = 0 \text{ and } g(x) \leq 0\}$. This is trivially the case for the cone $C = \{x \in \mathbb{R}^n : x \geq 0\}$ of vectors in $\mathbb{R}^n$ with nonnegative entries.

More generally, we can gain insight by treating C separately from the other constraints. To this end, we introduce the following Lagrangian function with restricted domain (compare with (8.14)):

$$\begin{aligned} L\colon C \times \mathbb{R}^p &\to \mathbb{R} \\ : (x, \mu) &\mapsto L(x, \mu) = f(x) + \mu^\top (b - Ax). \end{aligned} \tag{9.8}$$

The primal function $L_P\colon C \to \mathbb{R}$ is given by

$$L_P(x) = \sup_{\mu \in \mathbb{R}^p} L(x, \mu) = \begin{cases} f(x) & \text{if } Ax = b, \\ +\infty & \text{otherwise.} \end{cases} \tag{9.9}$$

The dual function $L_D\colon \mathbb{R}^p \to \mathbb{R}$ is defined by

$$L_D(\mu) = \inf_{x \in C} L(x, \mu) = \inf_{x \in C} \langle c, x \rangle - \mu^\top Ax + \mu^\top b. \tag{9.10}$$

(Notice that the infimum is taken over C only: not over all of $\mathcal{E}$.) Consider the expression $\mu^\top Ax$. Since $A\colon \mathcal{E} \to \mathbb{R}^p$ is a linear map from the Euclidean space $\mathcal{E}$ with the inner product $\langle \cdot, \cdot \rangle$ to the Euclidean space $\mathbb{R}^p$ with the canonical inner product, we can write $\mu^\top Ax = \langle A^*\mu, x \rangle$ where $A^*\colon \mathbb{R}^p \to \mathcal{E}$ is the adjoint of A. Therefore,

$$L_D(\mu) = \inf_{x \in C} \langle c - A^*\mu, x \rangle + \mu^\top b. \tag{9.11}$$

Consider the two following scenarios for some given $\mu \in \mathbb{R}^p$:

- Either $c - A^*\mu$ is in the dual of C. That means $\langle c - A^*\mu, x \rangle \geq 0$ for all x in C. Accordingly, taking the infimum over x in C yields zero for that first term (take $x = 0$ for example).

- Or $c - A^*\mu$ is not in the dual of C. That means there exists $x \in C$ such that $\langle c - A^*\mu, x\rangle < 0$. However, C being a cone, we can now multiply x by an arbitrarily large scalar: the resulting vector is still in the cone, and $\langle c - A^*\mu, x\rangle$ becomes arbitrarily small. Accordingly, taking the infimum over x in C yields $-\infty$ for that first term.

This discussion allows us to conclude that the dual function is given by:

$$L_D(\mu) = \begin{cases} b^\top \mu & \text{if } c - A^*\mu \in C^*, \\ -\infty & \text{otherwise.} \end{cases} \tag{9.12}$$

This leads to the following primal-dual pair:

$$\min_{x \in \mathcal{E}} \langle c, x\rangle \quad \text{s.t. } Ax = b \text{ and } x \in C, \tag{primal-CP}$$

$$\max_{\mu \in \mathbb{R}^p} b^\top \mu \quad \text{s.t. } c - A^*\mu \in C^*. \tag{dual-CP}$$

This setup is particularly interesting when C is a self-dual cone, that is, when $C^* = C$, as it can then be argued that (primal-CP) is itself the dual of (dual-CP).

We discuss some examples below, after stating the key duality theorem.

Theorem 9.37. *Assume there exists x_s in the interior of C such that $Ax_s = b$—this is a Slater condition for (primal-CP). Then strong duality holds: any pair (x^*, μ^*) such that*

$$x^* \in C, \qquad Ax^* = b, \qquad c - A^*\mu^* \in C^*, \qquad \text{and} \qquad \langle c, x^*\rangle = b^\top \mu^*$$

has the property that x^* is a global minimum for (primal-CP) and μ^* is a global maximum for (dual-CP). Moreover, if there exists a global minimum x^* for the primal, then there exists a μ^* to certify optimality as above.

Proof. The theorem does not immediately follow from our work in Sections 8.8, 9.4 and 9.6 because we have redefined the Lagrangian with a domain restricted by C. It takes some work (not too difficult) to revisit our earlier developments to this more general setting: we omit the details. □

Example 9.38 (Linear programming)**.** *With $\mathcal{E} = \mathbb{R}^n$ and the usual inner product $\langle c, x\rangle = c^\top x$, the adjoint of $A \in \mathbb{R}^{p\times n}$ is simply its transpose $A^* = A^\top \in \mathbb{R}^{n\times p}$. You can check that the nonnegative orthant is a self-dual cone:*

$$C = \{x \in \mathbb{R}^n : x \geq 0\}, \qquad C^* = C. \tag{9.13}$$

This leads to a particular instance of conic programming called linear programming, with the following primal-dual pairs:

$$\min_{x\in\mathbb{R}^n} c^\top x \quad s.t.\ Ax = b \text{ and } x \geq 0,$$
$$\max_{\mu\in\mathbb{R}^p} b^\top \mu \quad s.t.\ A^\top \mu \leq c.$$

As we already noted earlier, strong duality always holds for linear programming (no need for a Slater condition). That is because C is polyhedral*: it can be described through linear inequality constraints, namely, $x_1 \geq 0, \dots, x_n \geq 0$.*

Example 9.39 (Semidefinite programming)**.** *Let $\mathcal{E} = \mathrm{Sym}(n)$ be the Euclidean space of symmetric matrices of size n with its usual inner product $\langle C, X\rangle = \mathrm{Tr}(C^\top X) = \mathrm{Tr}(CX)$. A linear map $A\colon \mathrm{Sym}(n) \to \mathbb{R}^p$ is defined by p symmetric matrices $A_1, \dots, A_p \in \mathrm{Sym}(n)$ as follows:*

$$A(X) = \begin{bmatrix} \langle A_1, X\rangle \\ \vdots \\ \langle A_p, X\rangle \end{bmatrix}.$$

We endow $\mathbb{R}^p$ with its usual inner product. It is an exercise to check that the adjoint of A is given by

$$A^*(\mu) = \sum_{i=1}^{p} \mu_i A_i.$$

It is a nice exercise in linear algebra to check that the set of positive semidefinite matrices is a self-dual cone:

$$C = \{X \in \mathrm{Sym}(n) : X \succeq 0\}, \qquad C^* = C.$$

This leads to a particular instance of conic programming called semidefinite programming, with the following primal-dual pairs:

$$\min_{X\in\mathrm{Sym}(n)} \langle M, X\rangle \quad s.t.\ A(X) = b \text{ and } X \succeq 0,$$
$$\max_{\mu\in\mathbb{R}^p} b^\top \mu \quad s.t.\ \mu_1 A_1 + \cdots + \mu_p A_p \preceq M.$$

Strong duality holds if there exists a matrix $X_{\mathrm{s}} \succ 0$ such that $A(X_{\mathrm{s}}) = b$.

Do you see how Example 8.35 fits in this story? Can you figure out the dual of the dual of the combinatorial problem described in that example? You should find that it is a semidefinite program: a convex relaxation *with marvelous properties (google Goemans and Williamson).*

Chapter 10

Algorithms for constrained optimization

We consider a few different algorithms for optimization under constraints. This chapter is based in large parts on [Rus06, Ch. 6], and to a lesser extent on [NW06, Ch. 17].

10.1 Projected gradient descent for convex sets

Consider the optimization problem

$$\min_{x \in S} f(x) \tag{10.1}$$

where we assume the following:

- S is a non-empty, closed, convex set in a Euclidean space $\mathcal{E}$,
- $f \colon \mathcal{E} \to \mathbb{R}$ has L-Lipschitz continuous gradient (and may or may not be convex).

We let $\mathrm{Proj}_S \colon \mathcal{E} \to S$ denote the projection to the set S, as defined by:

$$\mathrm{Proj}_S(z) = \arg\min_{x \in S} \|x - z\|. \tag{10.2}$$

Recall from Exercise 9.17 that this projection exists and is unique under our assumptions on S. We will need the two following observations about Proj_S.

Lemma 10.1. *For all $z \in \mathcal{E}$ and $x \in S$, it holds that*

$$\langle z - \mathrm{Proj}_S(z), x - \mathrm{Proj}_S(z) \rangle \leq 0.$$

Proof. By definition, $\bar{z} \triangleq \mathrm{Proj}_S(z)$ is the unique minimizer of $g(y) = \frac{1}{2}\|y - z\|^2$ constrained to S. Since S is convex, g is (strongly) convex and $-\nabla g(y) = z - y$, we deduce by Corollary 9.8 that $\langle z - \bar{z}, x - \bar{z}\rangle \leq 0$ for all $x \in S$. □

Lemma 10.2. *Projection to S is non-expansive: for all $y, z \in \mathcal{E}$, we have*

$$\|\mathrm{Proj}_S(y) - \mathrm{Proj}_S(z)\| \leq \|y - z\|.$$

Stated differently: Proj_S *is 1-Lipschitz continuous.*

Proof. By Lemma 10.1, we get both of the following:

$$\langle y - \mathrm{Proj}_S(y), \mathrm{Proj}_S(z) - \mathrm{Proj}_S(y)\rangle \leq 0 \text{ and}$$
$$\langle z - \mathrm{Proj}_S(z), \mathrm{Proj}_S(y) - \mathrm{Proj}_S(z)\rangle \leq 0.$$

Add these inequalities up to find:

$$\langle \mathrm{Proj}_S(y) - y + z - \mathrm{Proj}_S(z), \mathrm{Proj}_S(y) - \mathrm{Proj}_S(z)\rangle \leq 0.$$

Reorganize to get

$$\|\mathrm{Proj}_S(y) - \mathrm{Proj}_S(z)\|^2 \leq \langle y - z, \mathrm{Proj}_S(y) - \mathrm{Proj}_S(z)\rangle . \tag{10.3}$$

On the other hand, it is clear that

$$\begin{aligned} 0 &\leq \|(\mathrm{Proj}_S(y) - \mathrm{Proj}_S(z)) - (y - z)\|^2 \\ &= \|\mathrm{Proj}_S(y) - \mathrm{Proj}_S(z)\|^2 + \|y - z\|^2 - 2\langle y - z, \mathrm{Proj}_S(y) - \mathrm{Proj}_S(z)\rangle . \end{aligned} \tag{10.4}$$

Combine (10.3) and (10.4) to reveal

$$\begin{aligned} 0 &\leq \|\mathrm{Proj}_S(y) - \mathrm{Proj}_S(z)\|^2 + \|y - z\|^2 - 2\|\mathrm{Proj}_S(y) - \mathrm{Proj}_S(z)\|^2 \\ &= \|y - z\|^2 - \|\mathrm{Proj}_S(y) - \mathrm{Proj}_S(z)\|^2. \end{aligned}$$

This concludes the proof. □

Assuming projection to the set S is reasonably easy to compute (which happens but isn't particularly frequent), it makes sense to consider the following algorithm called *projected gradient descent* (PGD): given $x_0 \in S$, iterate

$$x_{k+1} = \mathrm{Proj}_S(x_k - \alpha_k \nabla f(x_k)), \qquad \text{for } k = 0, 1, 2, \ldots \tag{PGD}$$

where $\alpha_0, \alpha_1, \ldots$ are step-sizes to be determined. Note that this is a direct generalization of gradient descent (GD) from the case $S = \mathcal{E}$.

Under our assumptions on S and f, we can make a clean statement regarding limit points of PGD.

Theorem 10.3. *If the sublevel set $\{x \in S : f(x) \leq f(x_0)\}$ is bounded and if the step-size is a constant $\alpha_k = \alpha$ for all k with $\alpha \in (0, 2/L)$, then the sequence $(x_k)_{k\geq 0}$ generated by (PGD) has at least one accumulation point x^*, and all of them are stationary points, that is,*

$$-\nabla f(x^*) \in \mathrm{N}_{x^*} S.$$

In particular, if f is convex then x^* is optimal.

Proof. Apply Lemma 10.1 with $z = x_k - \alpha_k \nabla f(x_k)$ and $x = x_k$—using $x_{k+1} = \mathrm{Proj}_S(z)$—to see that

$$\langle x_k - \alpha_k \nabla f(x_k) - x_{k+1}, x_k - x_{k+1} \rangle \leq 0.$$

Reorganize that inequality to get

$$\langle \nabla f(x_k), x_{k+1} - x_k \rangle \leq -\frac{1}{\alpha_k} \|x_k - x_{k+1}\|^2.$$

This is useful because the Lipschitz continuity of ∇f provides the following by Theorem 3.2:

$$f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2} \|x_{k+1} - x_k\|^2.$$

Combining the two last statement we find

$$f(x_k) - f(x_{k+1}) \geq \left(\frac{1}{\alpha_k} - \frac{L}{2} \right) \|x_{k+1} - x_k\|^2.$$

Under our assumption that $\alpha_k = \alpha \in (0, 2/L)$ for all k, the coefficient $\frac{1}{\alpha_k} - \frac{L}{2}$ is positive. It then holds that $f(x_{k+1}) \leq f(x_k)$ for all k. Thus, the sequence $(x_k)_{k\geq 0}$ remains in the subset $S_0 = \{x \in S : f(x) \leq f(x_0)\}$. By assumption, that subset is compact (it is bounded and closed). Therefore, f is lower-bounded on that subset. It follows that the sequence of real numbers $(f(x_k))_{k\geq 0}$ is convergent. We deduce that $f(x_k) - f(x_{k+1}) \to 0$, hence also that $\|x_{k+1} - x_k\| \to 0$. Equivalently,

$$\lim_{k\to\infty} \|\mathrm{Proj}_S(x_k - \alpha \nabla f(x_k)) - x_k\| = 0. \tag{10.5}$$

Since S_0 is compact and (x_k) is a sequence in S_0, we know that (x_k) admits a subsequence converging to an accumulation point x^* in S_0. (In particular, x^* is in S.) Moreover, we know that Proj_S is continuous by Lemma 10.2.

Therefore, we can "bring the limit inside" of (10.5) along the subsequence to deduce that

$$\|\mathrm{Proj}_S(x^* - \alpha\nabla f(x^*)) - x^*\| = 0. \tag{10.6}$$

Stated differently, x^* is a fixed point of (PGD):

$$x^* = \mathrm{Proj}_S(x^* - \alpha\nabla f(x^*)).$$

Lemma 10.1 applied to that last statement with $z = x^* - \alpha\nabla f(x^*)$ yields:

$$\forall x \in S, \qquad \langle -\alpha\nabla f(x^*), x - x^*\rangle \leq 0.$$

In other words, x^* is stationary for f on S (by Theorem 9.7). □

Exercise 10.4. *Show with a drawing that* Proj_S *may be discontinuous if* S *is non-empty and closed but fails to be convex. This reveals why the PGD iteration map* $x \mapsto \mathrm{Proj}_S(x - \alpha\nabla f(x))$ *could be discontinuous if* S *is not convex. It would be much harder to analyze the algorithm if we allowed that to happen.*

Exercise 10.5. *Let* $S = \{x \in \mathbb{R}^n : \|x\| \leq 1\}$ *be the unit norm ball. Give an expression for* Proj_S*. Given a symmetric matrix* A *of size* n*, let* $f(x) = \frac{1}{2}x^\top Ax$*. What is the Lipschitz constant of* ∇f*? Can you easily compute an upper bound for it? Implement* (PGD) *for this problem, with a proper choice of step-size. What does this algorithm (aim to) compute?*

Exercise 10.6. *Let* S *be a box in* $\mathbb{R}^n$ *with* $\ell, u \in \mathbb{R}^n$ *defining its limits:*

$$S = \{x \in \mathbb{R}^n : \ell_i \leq x_i \leq u_i \text{ for } i = 1, \ldots, n\}.$$

For example, $\{x \in \mathbb{R}^n : \|x\|_\infty \leq 1\}$ *is the box corresponding to* $\ell = -\mathbf{1}$ *and* $u = \mathbf{1}$*, where* $\mathbf{1} \in \mathbb{R}^n$ *is the all-ones vector. Give a formula for* Proj_S*.*

Hint: It is quite intuitive to guess the formula with a 2D drawing. You can then justify it using what we've learned about optimality conditions for strongly convex functions constrained to convex sets.

10.2 Quadratic penalty methods

We consider optimization problems under equality and inequality constraints as introduced in Chapter 8. Namely, we consider

$$\min_{x \in \mathcal{E}} f(x) \text{ subject to } x \in S \tag{10.7}$$

with

$$S = \{x \in \mathcal{E} : h(x) = 0 \text{ and } g(x) \leq 0\},$$

where $f\colon \mathcal{E} \to \mathbb{R}$, $h\colon \mathcal{E} \to \mathbb{R}^p$ and $g\colon \mathcal{E} \to \mathbb{R}^m$ are assumed to be continuously differentiable.

The general aim of penalty methods for constrained optimization is to "move" the constraints to the cost function as penalty terms. In so doing, we create a new, unconstrained optimization problem for which we hope to be able to use the algorithms we already have. For this to happen, we need the new cost function to retain certain properties. In this course, we have only discussed algorithms for differentiable cost functions: we aim to preserve that property.

Remark 10.7. *We could also resort to nondifferentiable penalty terms (this has some advantages), in which case we would need algorithms for unconstrained optimization with nondifferentiable cost functions: those exist. Also, we could decide only to penalize* some *of the constraints, keeping the other constraints as is. This makes sense if the remaining constraints are easy to handle, e.g., using projected gradient descent.*

Consider the following simple claim (its proof is an exercise in calculus):

Lemma 10.8. *The real functions*

$$t \mapsto t^2 \qquad \text{and} \qquad t \mapsto \max(0, t)^2$$

are continuously differentiable, with derivatives $t \mapsto 2t$ and $t \mapsto 2\max(0, t)$ respectively.

We use that fact to transform equality and inequality constraints into penalty terms. Given a parameter $\beta > 0$ called the *penalty weight*, we define a cost function F_β by adding a *quadratic penalty term* $P(x)$ to the cost function $f(x)$:

$$F_\beta(x) = f(x) + \beta P(x) \quad \text{with} \tag{10.8}$$

$$P(x) = \frac{1}{2}\sum_{i=1}^{p}(h_i(x))^2 + \frac{1}{2}\sum_{i=1}^{m}\max(0, g_i(x))^2. \tag{10.9}$$

From Lemma 10.8, we deduce that P (hence also F_β) is continuously differentiable (by composition). Also, if there are no inequality constraints and f, h are twice continuously differentiable, then so are P and F_β. Thus, we can apply algorithms for unconstrained minimization to F_β.

The penalty term punishes constraint violations:

$$P(x) = 0 \text{ if } x \in S, \qquad \text{and} \qquad P(x) > 0 \text{ if } x \notin S.$$

If x is infeasible and β is large, then the penalty term $\beta P(x)$ can be huge. Intuitively, if we let β be large, then the global minimizers of F_β should be close to S. Moreover, for points in S, the function F_β reduces to f. Thus, minimizing F_β *should* become *mostly* equivalent to minimizing f on S. The first goal of this section is to make these intuitions more precise, and also to highlight certain caveats.

Based on the above, we may be tempted to proceed as follows:

1. Fix a huge value for β.
2. Run gradient descent, trust-regions or some other algorithm on F_β, for example with a random initialization.

Unfortunately, that approach is likely to fail, mainly for two reasons:

1. Even if β is huge, minimizers of F_β are typically not feasible for (10.7), that is: they typically do not satisfy the constraints exactly.
2. The function F_β is numerically tricky to optimize when β is large, especially if our initial guess is far from a minimizer.

We justify both claims later. For now, we act on these claims by proposing Algorithm 10.1: this is a "framework" more than an algorithm, in the sense that it stipulates guidelines rather than a precise method. Echoing the above, the general idea is:

1. Start with a moderate value for β, and *gradually* increase it: we solve a sequence of minimization problems rather than a single one.
2. Initialize each minimization with the solution found at the previous stage: this is called *warm-starting*.

Algorithm 10.1 is not great, but it is a start. It forms the basis for better algorithms we describe later.

To provide some support for Algorithm 10.1, we start with a basic theorem that uses very few properties of P. We should think of it as a "sanity check." In words, it states that **if**

- The target problem (10.7) has a solution (which is reasonable), and
- We can find global minimizers of F_β for a growing sequence of βs (which is a *big* if!), and

- That sequence has a limit point (which is reasonable),

then the aforementioned limit point is a solution to our target problem.

Theorem 10.9. *Assume problem (10.7) has a global minimum x^*. Pick a sequence of positive penalty weights $(\beta_k)_{k\geq 1}$ with $\beta_k \to \infty$ and assume each F_{β_k} (10.8) has a global minimum x_k. Then every accumulation point of the sequence $(x_k)_{k\geq 1}$ is a global minimum of (10.7). Such an accumulation point exists in particular if $\{x \in \mathcal{E} : f(x) \leq f(x^*)\}$ is bounded.*

Proof. Let x^* be a global minimizer for (10.7). Since x_k is a global minimizer for F_{β_k} and x^* is feasible for (10.7) ($x^* \in S$), we have in particular that

$$f(x_k) + \beta_k P(x_k) = F_{\beta_k}(x_k) \leq F_{\beta_k}(x^*) = f(x^*). \tag{10.10}$$

Therefore, for all k it holds that

$$P(x_k) \leq \frac{f(x^*) - f(x_k)}{\beta_k}. \tag{10.11}$$

Let $\bar{x}$ be an accumulation point of $(x_k)_{k\geq 1}$: there exists a subsequence $(x_{k_i})_{i\geq 1}$ such that $\lim_{i\to\infty} x_{k_i} = \bar{x}$. Take the limit along that subsequence on both sides of (10.11) to reveal that

$$P(\bar{x}) = \lim_{i\to\infty} P(x_{k_i}) \leq \lim_{i\to\infty} \frac{f(x^*) - f(x_{k_i})}{\beta_{k_i}} = 0.$$

(The first equality holds because P is continuous; the second equality holds because $\frac{1}{\beta_{k_i}} \to 0$ and $f(x^*) - f(x_{k_i}) \to f(x^*) - f(\bar{x})$ as $i \to \infty$.) Thus, $P(\bar{x}) = 0$, that is, $\bar{x}$ satisfies all the constraints of (10.7). Moreover,

$$f(\bar{x}) = \lim_{i\to\infty} f(x_{k_i}) \leq \lim_{i\to\infty} f(x_{k_i}) + \beta_{k_i} P(x_{k_i}) \leq f(x^*)$$

(The first inequality holds because $\beta_{k_i} P(x_{k_i}) \geq 0$; the second inequality holds by (10.10).) Therefore, $\bar{x}$ is a global minimizer for (10.7). The last claim of the theorem holds because (10.10) implies $f(x_k) \leq f(x^*)$, so that the sequence $(x_k)_{k\geq 0}$ is bounded under the stated assumption. □

So far, we have used very few properties of the quadratic penalty function P. We can gain much more insight into the sequence (x_k) discussed in Theorem 10.9 by using finer properties of P. This is what we aim to do now.

Recall the definition of the set of active inequality constraints at a point x, namely, $\mathcal{I}(x)$ (8.7):

$$\mathcal{I}(x) = \{i \in \{1, \ldots, m\} : g_i(x) = 0\}.$$

Algorithm 10.1 Quadratic penalty method (general framework)

1: Pick an initial guess $x_0 \in \mathcal{E}$.
2: Pick an initial weight $\beta_1 > 0$.
3: **for** k in $1, 2, 3 \ldots$ **do**
4: Compute[1] x_k by applying a minimization algorithm to F_{β_k} initialized at x_{k-1} (a *warm start*).
5: Set $\beta_{k+1} \geq \beta_k$, making sure $\beta_k \to \infty$ (e.g.: $\beta_{k+1} = 2\beta_k$).
6: **end for**

We shall require the MFCQ constraint qualification: recall Theorem 8.16. Remember also that LICQ at x implies MFCQ at x. Moreover, if the equality constraints are affine and the inequality constraints are convex, we know from Exercise 9.31 that Slater's condition implies MFCQ at all feasible points.

Theorem 10.10. *Let $(\beta_k)_{k\geq 1}$ be a sequence of positive penalty weights and let $(x_k)_{k\geq 1}$ be a sequence such that each x_k is a critical point of F_{β_k}. Assume $(x_k)_{k\geq 1}$ converges to a feasible point x^* of (10.7)[2] and that MFCQ holds at x^*. Then, the sequences $(\mu^k)_{k\geq 1}$ in $\mathbb{R}^p$ and $(\lambda^k)_{k\geq 1}$ in $\mathbb{R}^m_+$ defined by*

$$\mu_i^k \triangleq \beta_k h_i(x_k), \qquad i = 1, \ldots, p, \tag{10.12}$$

$$\lambda_i^k \triangleq \beta_k \max(0, g_i(x_k)), \qquad i = 1, \ldots, m, \tag{10.13}$$

are bounded, and each accumulation point (μ^*, λ^*) of $(\mu^k, \lambda^k)_{k\geq 1}$ is a valid pair of Lagrange multipliers for (10.7) at x^*. In particular, x^* is stationary.

Proof. The proof is in three steps.

1. Each x_k satisfies the first-order necessary optimality conditions for F_{β_k}, that is,

$$\begin{aligned} 0 &= \nabla F_{\beta_k}(x_k) \\ &= \nabla f(x_k) + \sum_{i=1}^{p} \beta_k h_i(x_k) \nabla h_i(x_k) + \sum_{i=1}^{m} \beta_k \max(0, g_i(x_k)) \nabla g_i(x_k) \\ &= \nabla f(x_k) + \sum_{i=1}^{p} \mu_i^k \nabla h_i(x_k) + \sum_{i=1}^{m} \lambda_i^k \nabla g_i(x_k), \end{aligned} \tag{10.14}$$

[1] We hope x_k is a minimizer, but absent special structure such as convexity, that may be difficult to guarantee / verify.

[2] If each x_k is a global minimizer of F_{β_k}, this assumption might follow from Theorem 10.9 after passing to a subsequence.

where we used the definitions of μ_i^k and λ_i^k. Moreover, for $i \notin \mathcal{I}(x^*)$ (that is, if $g_i(x^*) < 0$), we necessarily have $g_i(x_k) < 0$ for all sufficiently large k (that is because g is continuous and $x_k \to x^*$). Thus, $\lambda_i^k = 0$ for all sufficiently large k when $i \notin \mathcal{I}(x^*)$. This allows us to rewrite (10.14) as

$$0 = \nabla f(x_k) + \sum_{i=1}^{p} \mu_i^k \nabla h_i(x_k) + \sum_{i \in \mathcal{I}(x^*)} \lambda_i^k \nabla g_i(x_k), \tag{10.15}$$

valid for all sufficiently large k.

2. The sequence (μ^k, λ^k) is bounded. We shall use the MFCQ assumption to prove this by contradiction. Suppose (μ^k, λ^k) is unbounded. This allows us to pass to a subsequence such that $\|(\mu^{k_j}, \lambda^{k_j})\| \to \infty$ with $j \to \infty$. Divide (10.15) by $\|(\mu^{k_j}, \lambda^{k_j})\|$:

$$0 = \frac{1}{\|(\mu^{k_j}, \lambda^{k_j})\|} \nabla f(x_{k_j}) + \sum_{i=1}^{p} \frac{\mu_i^{k_j}}{\|(\mu^{k_j}, \lambda^{k_j})\|} \nabla h_i(x_{k_j}) + \sum_{i \in \mathcal{I}(x^*)} \frac{\lambda_i^{k_j}}{\|(\mu^{k_j}, \lambda^{k_j})\|} \nabla g_i(x_{k_j}). \tag{10.16}$$

Normalize $(\mu^{k_j}, \lambda^{k_j})$ to define

$$(\bar{\mu}^{k_j}, \bar{\lambda}^{k_j}) = \frac{(\mu^{k_j}, \lambda^{k_j})}{\|(\mu^{k_j}, \lambda^{k_j})\|}.$$

The sequence $(\bar{\mu}^{k_j}, \bar{\lambda}^{k_j})_{j \geq 0}$ is bounded by construction, hence it admits a convergent subsequence: let $(\bar{\mu}, \bar{\lambda})$ denote its limit. Of course, $\|(\bar{\mu}, \bar{\lambda})\| = 1$. Then, taking the limit of (10.16) along that subsequence of the subsequence and using $x_k \to x^*$ reveals:

$$0 = \sum_{i=1}^{p} \bar{\mu}_i \nabla h_i(x^*) + \sum_{i \in \mathcal{I}(x^*)} \bar{\lambda}_i \nabla g_i(x^*). \tag{10.17}$$

Through Lemma 8.17, the MFCQ condition at x^* provides us with $\hat{x} \in \mathcal{E}$ such that

$$\begin{aligned}
&\langle \nabla h_i(x^*), \hat{x} - x^* \rangle < 0, && \text{for all } i = 1, \ldots, p \text{ with } \bar{\mu}_i > 0, \\
&\langle \nabla h_i(x^*), \hat{x} - x^* \rangle > 0, && \text{for all } i = 1, \ldots, p \text{ with } \bar{\mu}_i \leq 0 \text{ and} \\
&\langle \nabla g_i(x^*), \hat{x} - x^* \rangle < 0, && \text{for all } i \in \mathcal{I}(x^*).
\end{aligned}$$

Take the inner product of (10.17) with $\hat{x} - x^*$: this implies that $\bar{\mu} = 0$ and $\bar{\lambda} = 0$, yet $\|(\bar{\mu}, \bar{\lambda})\| = 1$. This is our contradiction.

3. Since $(\mu^k, \lambda^k)_{k\geq 1}$ is bounded, we can extract a convergent subsequence. Let (μ^*, λ^*) be its limit. By continuity, we can take the limit along that subsequence in (10.15) and use $x_k \to x^*$ to deduce that

$$0 = \nabla f(x^*) + \sum_{i=1}^{p} \mu_i^* \nabla h_i(x^*) + \sum_{i \in \mathcal{I}(x^*)} \lambda_i^* \nabla g_i(x^*).$$

   If $g_i(x^*) = 0$, then of course $\lambda_i^* g_i(x^*) = 0$; if $g_i(x^*) < 0$, then $g_i(x_k) < 0$ for all k sufficiently large, and therefore $\lambda_i^k = 0$ for all k sufficiently large. Thus, $\lambda_i^* g_i(x^*) = 0$ in all cases.

We conclude that x^* is KKT for (10.7) with Lagrange multipliers (μ^*, λ^*). □

Theorem 10.10 suggests that we could use a solution of a penalized problem F_β with large β to estimate the Lagrange multipliers at a solution of (10.7). That will be important later.

Exercise 10.11. *Write code to apply the quadratic penalty method to the simple optimization problem*

$$\min_{x \in \mathbb{R}^2} x_1 + x_2 \text{ subject to } x_1^2 + x_2^2 - 2 = 0.$$

To solve the subproblems, that is, to minimize F_β for each individual value β, you can use code you wrote in previous exercise sessions / homework assignments, or you can use Matlab's Optimization Toolbox: the code below requires you to provide a function `[val, grad] = F(x, beta)` *implementing $F_\beta(x)$ (as the first output) and $\nabla F_\beta(x)$ (as the second output) as well as an initialization* `x_in`*, and it (attempts to) return a minimizer* `x_out`*.*

```
% See 'help fminunc': Matlab's unconstrained
   minimizer.
options = optimoptions('fminunc', '
   SpecifyObjectiveGradient', true);
x_out = fminunc(@(x) F(x, beta), x_in, options);
```

It is instructive to visualize the penalized function F_β for various values of β to get a sense of how the penalty shapes the 'landscape' of the cost function, and to display on those plots the sequence of solutions (x_k) that you compute.

Exercise 10.12. *The function F_β may fail to be bounded below, in which case minimizing it can completely fail. Verify this claim on the following example:*

$$\min_{x \in \mathbb{R}^2} -5x_1^2 + x_2^2 \text{ subject to } x_1 = 1.$$

Argue that for $\beta < 10$ the function F_β is unbounded below. What is the situation like for $\beta \geq 10$? Can we still hope to find a solution for this problem via some instantiation of Algorithm 10.1?

Exercise 10.13. *Verify that if the equality constraints are affine and the inequality constraints are convex then the quadratic penalty P (10.9) is convex. (Hint: first check that $\phi(s) = \max(0, s)^2$ is convex and non-decreasing.)*

Interpret the results of this section in that context, for example assuming also that f is strongly convex and that Slater's condition holds.

10.3 The trouble with quadratic penalties

From (10.12) and (10.13), we can clearly see that even for very large β the constraints won't be satisfied exactly at a global minimizer x_k of F_{β_k}. At best, we can hope to get

$$h_i(x_k) = \frac{\mu_i^k}{\beta_k} \approx \frac{\mu_i^*}{\beta_k}, \qquad \max(0, g_i(x_k)) = \frac{\lambda_i^k}{\beta_k} \approx \frac{\lambda_i^*}{\beta_k}, \tag{10.18}$$

where μ^*, λ^* would be Lagrange multipliers at a global minimizer x^* of (10.7). Intuitively, this makes sense: x^* cannot be a minimizer of F_β for finite β because moving away slightly from x^* incurs a quadratic penalty in F_β for constraint violation (because $P(x^*) = 0$ implies $\nabla P(x^*) = 0$), but it might be rewarded with a linear improvement in the value of f. Thus, it is generically better to move away from the constraints at least slightly. More precisely,

$$\begin{aligned}\nabla F_\beta(x^*) &= \nabla f(x^*) + \sum_{i=1}^{p} \beta h_i(x^*) \nabla h_i(x^*) + \sum_{i=1}^{m} \beta \max(0, g_i(x^*)) \nabla g_i(x^*) \\ &= \nabla f(x^*) \\ &= -\sum_{i=1}^{p} \mu_i^* \nabla h_i(x^*) - \sum_{i \in \mathcal{I}(x^*)} \lambda_i^* \nabla g_i(x^*).\end{aligned}$$

(The second equality holds because x^* is feasible; the third equality holds because x^* is KKT with Lagrange multipliers μ^*, λ^*.) Thus, for the unconstrained problem of minimizing F_β, the steepest descent direction at x^* is

$$-\nabla F_\beta(x^*) = \sum_{i=1}^{p} \mu_i^* \nabla h_i(x^*) + \sum_{i \in \mathcal{I}(x^*)} \lambda_i^* \nabla g_i(x^*).$$

In general, this is a nonzero vector which pushes us to violate the constraints: x^* is not critical for F_β, regardless of how large β is.

This is problematic in practice because it forces us to work with very large values of β to find solutions which are nearly feasible, as per (10.18). Yet, minimizing F_β becomes increasingly difficult numerically as β grows. That is because of *conditioning issues*. Recall from Theorems 3.15, 4.31 and 5.11 that the condition number of the Hessian of the cost function at a minimizer controls the local rate of convergence. Explicitly, if the Hessian at a minimizer is positive definite, its condition number is the ratio of its largest to its smallest eigenvalues. If that number is large, then optimization algorithms can be slowed down. Below, we argue that the condition number of the Hessian of F_β deteriorates with β.

Consider the case with only equality constraints as an example (we do this so that it is possible to compute the Hessian, assuming f and h are twice continuously differentiable):

$$\min_{x \in \mathcal{E}} f(x) \text{ subject to } h(x) = 0.$$

The penalized cost function and its derivatives are:

$$F_\beta(x) = f(x) + \frac{\beta}{2} \sum_{i=1}^{p} (h_i(x))^2,$$

$$\nabla F_\beta(x) = \nabla f(x) + \beta \sum_{i=1}^{p} h_i(x) \nabla h_i(x),$$

$$\nabla^2 F_\beta(x)[v] = \nabla^2 f(x)[v] + \beta \sum_{i=1}^{p} h_i(x) \nabla^2 h_i(x)[v] + \beta \sum_{i=1}^{p} \langle \nabla h_i(x), v \rangle \, \nabla h_i(x).$$

Remark 10.14. *It is time for a side note about notation. Our expression for the Hessian of F_β involves terms of the form*

$$v \mapsto A(v) \triangleq \langle \nabla h_i(x), v \rangle \, \nabla h_i(x).$$

These are linear maps from $\mathcal{E}$ to $\mathcal{E}$. Indeed, $\nabla h_i(x)$ is a (fixed) vector in $\mathcal{E}$, and $v \mapsto \langle \nabla h_i(x), v \rangle = \mathrm{D} h_i(x)[v]$ is a linear function from $\mathcal{E}$ to $\mathbb{R}$. In fact, A is symmetric *(Definition 2.9): for all $u, v \in \mathcal{E}$ we have*

$$\begin{aligned} \langle u, A(v) \rangle &= \langle u, \langle \nabla h_i(x), v \rangle \, \nabla h_i(x) \rangle \\ &= \langle \nabla h_i(x), v \rangle \, \langle u, \nabla h_i(x) \rangle \\ &= \langle \langle u, \nabla h_i(x) \rangle \, \nabla h_i(x), v \rangle = \langle A(u), v \rangle . \end{aligned}$$

We can make this fact more visual by introducing a bit of notation. Formally, let us think of the (fixed) vector $\nabla h_i(x)$ as a linear map from $\mathbb{R}$ to $\mathcal{E}$, as follows:

$$\nabla h_i(x) \colon \mathbb{R} \to \mathcal{E} \colon \alpha \mapsto \alpha \nabla h_i(x).$$

What is the adjoint of that map, which we denote by $(\nabla h_i(x))^*$ *or, more simply, by* $\nabla h_i(x)^*$*? It is a linear map from* $\mathcal{E}$ *to* $\mathbb{R}$ *defined by the following property, where* $\langle \alpha_1, \alpha_2 \rangle_{\mathbb{R}} = \alpha_1 \alpha_2$ *denotes the inner product on* $\mathbb{R}$*:*

$$\forall v \in \mathcal{E}, \alpha \in \mathbb{R}, \qquad \alpha \nabla h_i(x)^*[v] = \langle \nabla h_i(x)^*[v], \alpha \rangle_{\mathbb{R}} = \langle v, \alpha \nabla h_i(x) \rangle = \alpha \mathrm{D} h_i(x)[v].$$

By identification, we deduce that $\nabla h_i(x)^* = \mathrm{D} h_i(x)$*, that is,*

$$\nabla h_i(x)^*[v] = \mathrm{D} h_i(x)[v] = \langle \nabla h_i(x), v \rangle . \tag{10.19}$$

Thus, it makes sense to think of $\mathrm{D} h_i(x)$ *as the* adjoint *of* $\nabla h_i(x)$ *(and vice versa). With this notation, we can write the linear map* A *from above as:*

$$A = \nabla h_i(x) \nabla h_i(x)^*.$$

Indeed, we then have

$$A(v) = \nabla h_i(x) \nabla h_i(x)^*[v] = \mathrm{D} h_i(x)[v] \nabla h_i(x) = \langle \nabla h_i(x), v \rangle \nabla h_i(x).$$

The notation $A = \nabla h_i(x) \nabla h_i(x)^*$ *has the advantage that it makes it visually clear that* A *is a symmetric linear map. Check the following claim:* A *is positive semidefinite and it has rank at most one.*

Example 10.15. *Consider the important special case* $\mathcal{E} = \mathbb{R}^n$ *with the usual inner product* $\langle u, v \rangle = u^\top v$*. Then, given a function* $h_i \colon \mathbb{R}^n \to \mathbb{R}$*, the gradient* $\nabla h_i(x)$ *at a point* $x \in \mathbb{R}^n$ *is a vector in* $\mathbb{R}^n$*: a matrix with a single column. Taking the adjoint of* $\nabla h_i(x)$ *comes down to taking its transpose as a matrix, that is:* $\nabla h_i(x)^* = \nabla h_i(x)^\top$*. Then, the linear map* $v \mapsto \langle \nabla h_i(x), v \rangle \nabla h_i(x)$ *is exactly the symmetric matrix* $\nabla h_i(x) \nabla h_i(x)^\top$*.*

Equipped with new notation, we further rewrite the Hessian of F_β as follows, with $\nabla h_i(x)^* \triangleq \mathrm{D} h_i(x)$:

$$\nabla^2 F_\beta(x) = \nabla^2 f(x) + \beta \sum_{i=1}^{p} h_i(x) \nabla^2 h_i(x) + \beta \sum_{i=1}^{p} \nabla h_i(x) \nabla h_i(x)^*. \tag{10.20}$$

Consider this expression within the context of Theorem 10.10 with large β_k. Then, with x_k a global minimizer of F_{β_k} we may expect $x_k \approx x^*$ and $\beta_k h_i(x_k) \approx \mu_i^*$ for some vector of Lagrange multipliers μ^* at x^*. This justifies the following claim:

$$\nabla^2 F_{\beta_k}(x_k) \approx \nabla^2 f(x^*) + \sum_{i=1}^{p} \mu_i^* \nabla^2 h_i(x^*) + \beta_k \sum_{i=1}^{p} \nabla h_i(x^*) \nabla h_i(x^*)^*.$$

The first two terms form some matrix independent of β_k, whereas the third term is a positive semidefinite matrix of rank at most p and which scales with β_k. Therefore, unless the last term is zero or full rank (which would be unusual), we find that $\nabla^2 F_{\beta_k}(x_k)$ has some of its eigenvalues which converge to constants while other eigenvalues scale with β_k.[3] It follows that the condition number of $\nabla^2 F_{\beta_k}(x_k)$ diverges to infinity with $\beta_k \to \infty$.

In the next section, we discuss a different algorithm which avoids the necessity to increase β to infinity. It does so by adding a penalty term which scales linearly with constraint violation.

Example 10.16. *With $\mathcal{E} = \mathbb{R}^2$, let $f(x) = x_1 + x_2$ and $h(x) = x_1^2 + x_2^2 - 1$. The quadratic penalty formulation of $\min_{x\in\mathbb{R}^2} f(x)$ s.t. $h(x) = 0$ involves*

$$F_\beta(x) = x_1 + x_2 + \frac{\beta}{2}(x_1^2 + x_2^2 - 1)^2.$$

It is easy to compute the following derivatives:

$$\begin{aligned}
\nabla F_\beta(x) &= \begin{bmatrix}1\\1\end{bmatrix} + \beta(x_1^2 + x_2^2 - 1)\begin{bmatrix}2x_1\\2x_2\end{bmatrix}, \\
\nabla^2 F_\beta(x) &= 2\beta\begin{bmatrix}x_1^2 + x_2^2 - 1 + 2x_1^2 & 2x_1x_2 \\ 2x_1x_2 & x_1^2 + x_2^2 - 1 + 2x_2^2\end{bmatrix} \\
&= 2\beta h(x) I_2 + 4\beta x x^\top.
\end{aligned}$$

If x is a global minimizer for F_β then we expect $\beta h(x) \approx \mu$ where μ is the Lagrange multiplier at the global minimizer of the constrained problem. Then, the eigenvalues of $\nabla^2 F_\beta(x)$ are roughly 2μ and $2\mu+4\beta$ (since $\|x\| \approx 1$). Thus, the condition number of $\nabla^2 F_\beta(x)$ ought to behave like $\frac{2\mu+4\beta}{2\mu} \approx \frac{2}{\mu}\beta$ for large β: this grows to infinity with β.

Let us make the above more precise. Notice that if x is a critical point ($\nabla F_\beta(x) = 0$) then $x_1 = x_2$. This makes it possible to find all critical points: they are of the form $[t, t]^\top$ where t is a root of the following real function:

$$t \mapsto 1 + 2\beta(2t^2 - 1)t.$$

This is just a polynomial of degree 3. It takes a bit of calculus[4] to determine that its three roots are real when $\beta \geq \sqrt{\frac{27}{8}}$. We can compute the roots numerically to find the global minimizer of F_β: there are only three candidates to check. Let $t(\beta)$ denote the correct root, so that the global minimizer of F_β

[3] That last statement relies on your intuition in linear algebra / matrix analysis. We could make it precise with some work.

[4] https://en.wikipedia.org/wiki/Discriminant#Degree_3

is $[t(\beta), t(\beta)]^\top$. *(Playing around suggests that* $t(\beta)$ *is the smallest root of the cubic polynomial.) The code below shows numerically that the condition number of* $\nabla^2 F_{\beta_k}(x_k)$ *with* x_k *the global minimizer of* F_{β_k} *indeed deterioriates as* β_k *grows.*

```
for beta = [10, 100, 1000, 10000]
    f = @(x) x(1) + x(2) + (beta/2)*(x(1)^2 + x(2)^2
        - 1)^2;
    g = @(x) [1 ; 1] + 2*beta*(x(1)^2 + x(2)^2 - 1)*
       x;
    H = @(x) 2*beta*[x(1)^2 + x(2)^2 - 1 + 2*x(1)^2,
        2*x(1)*x(2) ; 2*x(1)*x(2), x(1)^2 + x(2)^2 -
        1 + 2*x(2)^2];
    t = min(roots([4*beta, 0, -2*beta, 1]));
    x = [t; t];
    disp(cond(H(x)));
end
% This code outputs:
%    32.2357   286.8375   2.8324e+03   2.8288e+04
```

10.4 Augmented Lagrangian methods, equalities

We aim to "fix" some of the issues that arise with the simple quadratic penalty methods. To ease the discussion, let us first concentrate on minimization problems with only equality constraints:

$$\min_{x \in \mathcal{E}} f(x) \text{ subject to } h(x) = 0, \tag{10.21}$$

where $f \colon \mathcal{E} \to \mathbb{R}$ and $h \colon \mathcal{E} \to \mathbb{R}^p$ are twice continuously differentiable.

Quadratic penalty methods introduce a penalized cost function

$$F_\beta(x) = f(x) + \beta P(x). \tag{10.22}$$

We argued earlier that solutions of the constrained problem are never minimizers of the penalized problem, regardless of how large β is. Indeed, let x^* be a minimizer of the constrained problem. In particular, $P(x^*) = 0$. Since $P(x) \geq 0$ for all x, we deduce that x^* is a global minimizer for P. Since P is differentiable, it follows that $\nabla P(x^*) = 0$. Thus,

$$\nabla F_\beta(x^*) = \nabla f(x^*).$$

In general, this is a nonzero vector. Assume we move away from x^* along a direction v for a small distance ϵ. Taylor expansions of f and P reveal that

$$\begin{aligned} f(x^* + \epsilon v) &= f(x^*) + \epsilon \langle \nabla f(x^*), v \rangle + O(\epsilon^2), \\ P(x^* + \epsilon v) &= O(\epsilon^2), \\ F_\beta(x^* + \epsilon v) &= f(x^*) + \epsilon \langle \nabla f(x^*), v \rangle + O(\beta \epsilon^2). \end{aligned}$$

Therefore, so long as we choose v such that $\langle \nabla f(x^*), v \rangle < 0$, we always have an incentive to move at least a little bit away from x^*. Indeed, we can reduce F_β by a term proportional to ϵ owing to the change of value in f, whereas the penalty term only scales with $O(\beta\epsilon^2)$.

To correct this behavior, a natural idea is to add a penalty term to F_β that scales linearly with constraint violation, that is, to add a term of the form

$$\sum_{i=1}^{p} \mu_i h_i(x) = \mu^\top h(x),$$

with some coefficients $\mu \in \mathbb{R}^p$—we will see in a moment that the choice of notation is not fortuitous.

We define the *Augmented Lagrangian function*:

$$L_\beta(x, \mu) = f(x) + \sum_{i=1}^{p} \mu_i h_i(x) + \frac{\beta}{2} \sum_{i=1}^{p} (h_i(x))^2. \tag{10.23}$$

The name comes from the fact that $L_\beta(x, \mu)$ is nothing but the Lagrangian function $L(x, \mu) = L_0(x, \mu)$ (8.14) "augmented" with a quadratic term. Here, we arrived at L_β differently: by adding a linear term to the quadratic penalty function F_β so that $F_\beta(x) = L_\beta(x, 0)$.

How should we pick the weights μ? Ideally, we would like to pick them such that some minimizer of the constrained problem is a minimizer of $x \mapsto L_\beta(x, \mu)$, for finite values of β. Let us investigate what it takes for this to happen.

We can easily compute the gradient of L_β with respect to x:

$$\nabla_x L_\beta(x, \mu) = \nabla f(x) + \sum_{i=1}^{p} (\mu_i + \beta h_i(x)) \nabla h_i(x). \tag{10.24}$$

Let x^* be stationary for the constrained problem. Assume LICQ holds at x^*: There exists a (unique) vector of Lagrange multipliers $\mu^* \in \mathbb{R}^p$ such that

$$-\nabla f(x^*) = \sum_{i=1}^{p} \mu_i^* \nabla h_i(x^*).$$

Combining this with (10.24) reveals that, at $x = x^*$,

$$\nabla_x L_\beta(x^*, \mu) = \sum_{i=1}^{p} (\mu_i - \mu_i^*) \nabla h_i(x^*). \tag{10.25}$$

LICQ at x^* means that $\nabla h_1(x^*), \ldots, \nabla h_p(x^*)$ are linearly independent. Thus,

$$\nabla_x L_\beta(x^*, \mu) = 0 \qquad \iff \qquad \mu = \mu^*.$$

The above discussion warrants the following claim.

Theorem 10.17. *Consider an equality constrained problem*

$$\min_{x \in \mathcal{E}} f(x) \text{ subject to } h(x) = 0$$

with f and h continuously differentiable. If x^* is a stationary point for the constrained problem and LICQ holds at x^*, then there exists a unique vector of Lagrange multipliers μ^* for x^*, and x^* is a stationary point for the augmented Lagrangian function $x \mapsto L_\beta(x, \mu)$ for any *$\beta \in \mathbb{R}$, if and only if $\mu = \mu^*$.*

This strongly suggests that we should aim to set $\mu = \mu^*$, with μ^* the Lagrange multipliers of a global minimizer x^* of the constrained problem. Of course, we do not know μ^* in advance: that is something for us to estimate algorithmically. Even before we get to that topic, there is another concern we should clear up: it is not sufficient for x^* to be a stationary point of $x \mapsto L_\beta(x, \mu^*)$ for us to be able to find it. Indeed, x^* could be a saddle point, or even a maximizer of $L_\beta(\cdot, \mu^*)$. We need to investigate whether or not x^* can be made to be (at least) a local minimizer of the augmented Lagrangian function. For this to happen, we shall need to take β large enough.

Let us compute the Hessian of L_β with respect to x. To this end, we start from (10.24) and compute a directional derivative along a direction $v \in \mathcal{E}$:

$$\nabla^2_{xx} L_\beta(x, \mu)[v] = \nabla^2 f(x)[v] + \sum_{i=1}^{p} (\mu_i + \beta h_i(x)) \nabla^2 h_i(x)[v] + \beta \sum_{i=1}^{p} \mathrm{D} h_i(x)[v] \cdot \nabla h_i(x). \tag{10.26}$$

With the notation $\nabla h_i(x)^* = \mathrm{D} h_i(x) \colon \mathcal{E} \to \mathbb{R}$ as in (10.20), we can rewrite the above as:

$$\nabla^2_{xx} L_\beta(x, \mu) = \nabla^2 f(x) + \sum_{i=1}^{p} (\mu_i + \beta h_i(x)) \nabla^2 h_i(x) + \beta \sum_{i=1}^{p} \nabla h_i(x) \nabla h_i(x)^*. \tag{10.27}$$

In particular, let x^* be a stationary point of the constrained problem where LICQ holds, and let μ^* be the Lagrange multipliers at x^*. Then,

$$\nabla^2_{xx} L_\beta(x^*, \mu^*) = \nabla^2 f(x^*) + \sum_{i=1}^{p} \mu_i^* \nabla^2 h_i(x^*) + \beta \sum_{i=1}^{p} \nabla h_i(x^*) \nabla h_i(x^*)^*. \tag{10.28}$$

This Hessian is a sum of two parts:

$$\nabla^2_{xx} L_\beta(x^*, \mu^*) = H + \beta M, \tag{10.29}$$

where

$$H \triangleq \nabla^2_{xx} L_0(x^*, \mu^*) = \nabla^2 f(x^*) + \sum_{i=1}^{p} \mu_i^* \nabla^2 h_i(x^*)$$

is the Hessian of the (non-augmented) Lagrangian function and

$$M \triangleq \sum_{i=1}^{p} \nabla h_i(x^*) \nabla h_i(x^*)^* \tag{10.30}$$

is a positive semidefinite map of rank at most p. We already know from Theorem 10.17 that $\nabla_x L_\beta(x^*, \mu^*) = 0$ for all β. Recall from Theorem 2.31 that if it also holds that $\nabla^2_{xx} L_\beta(x^*, \mu^*) \succ 0$, then x^* is a strict local minimizer of $x \mapsto L_\beta(x, \mu^*)$. It can be shown that this indeed happens for β large enough, under some additional conditions on x^*. Intuitively, this happens because (with the additional assumptions) H is positive definite when applied to vectors in the kernel of M, and M is positive definite when applied to vectors orthogonal to the kernel of M.

Theorem 10.18 ([NW06, Thm. 17.5])**.** *Consider the constrained problem* $\min_{x \in \mathcal{E}} f(x)$ *s.t.* $h(x) = 0$ *with* f *and* h *twice continuously differentiable. Assume LICQ holds at a local minimizer* x^* *of this problem and let* μ^* *be the Lagrange multipliers at* x^**. If* x^* *satisfies second-order sufficient conditions as specified in [NW06, Thm. 12.6], there exists a value* $\bar{\beta} > 0$ *such that* x^* *is a strict local minimizer of* $x \mapsto L_\beta(x, \mu^*)$ *for all* $\beta \geq \bar{\beta}$*.*

Proof. We already know that $\nabla_x L_\beta(x^*, \mu^*) = 0$ for all β: this follows from Theorem 10.17. It remains to show that $\nabla^2_{xx} L_\beta(x^*, \mu^*)$ is positive definite when β is larger than some threshold $\bar{\beta}$. For contradiction, assume this is not the case, that is: assume for all $k = 1, 2, \ldots$, with $\beta_k = k$, the map

$\nabla^2_{xx} L_{\beta_k}(x^*, \mu^*)$ is not positive definite. Thus, for each k we can find a vector $v_k \in \mathcal{E}$ with $\|v_k\| = 1$ such that

$$\begin{aligned} 0 \geq \left\langle v_k, \nabla^2_{xx} L_{\beta_k}(x^*, \mu^*)[v_k] \right\rangle &= \langle v_k, H(v_k) \rangle + \beta_k \langle v_k, M(v_k) \rangle \\ &= \langle v_k, H(v_k) \rangle + k \sum_{i=1}^{p} (\langle \nabla h_i(x^*), v_k \rangle)^2. \end{aligned}$$

Stated differently, for all $k = 1, 2, \ldots$ we have

$$0 \leq \sum_{i=1}^{p} (\langle \nabla h_i(x^*), v_k \rangle)^2 \leq -\frac{1}{k} \langle v_k, H(v_k) \rangle .$$

Note that $\langle v_k, H(v_k) \rangle$ is bounded by the smallest and largest eigenvalues of H, which are independent of k. Therefore, taking the limit of the above inequalities for $k \to \infty$ we find that

$$\lim_{k \to \infty} \sum_{i=1}^{p} (\langle \nabla h_i(x^*), v_k \rangle)^2 = 0.$$

Since the vectors v_k have unit norm, the sequence $(v_k)_{k \geq 1}$ is bounded. Hence, it has an accumulation point $\bar{v}$ (also of unit norm). The above limit then reveals that

$$\sum_{i=1}^{p} (\langle \nabla h_i(x^*), \bar{v} \rangle)^2 = 0.$$

In other words:

$$\langle \nabla h_i(x^*), \bar{v} \rangle = 0 \quad \text{for } i = 1, \ldots, p.$$

As a result,

$$0 \geq \left\langle \bar{v}, \nabla^2_{xx} L_{\beta_k}(x^*, \mu^*)[\bar{v}] \right\rangle = \langle \bar{v}, H(\bar{v}) \rangle .$$

Yet, the second-order sufficient condition at x^* states that $\langle v, H(v) \rangle > 0$ for all v such that $\langle \nabla h_i(x^*), v \rangle = 0$ with $i = 1, \ldots, p$. Thus, we have reached a contradiction: the proof is complete. □

The above results suggest that if we can guess μ^* and if we set β sufficiently large (yet finite), then minimizing $L_\beta(\cdot, \mu^*)$ could reveal the sought minimizer x^*. How should we go about finding (a good approximation for) μ^*? The Augmented Lagrangian method (ALM) described in Algorithm 10.2

Algorithm 10.2 Augmented Lagrangian method (equality constraints)

Pick an initial guess $x_0 \in \mathcal{E}$ and initial multipliers $\mu^1 \in \mathbb{R}^p$.
Pick an initial weight $\beta_1 > 0$.
for k in $1, 2, 3 \ldots$ **do**
    Compute x_k by applying a minimization algorithm to $L_{\beta_k}(\cdot, \mu^k)$ initialized at x_{k-1} (a *warm start*).
    Set $\mu^{k+1} = \mu^k + \beta_k h(x_k)$.
    Set $\beta_{k+1} \geq \beta_k$ (e.g.: $\beta_{k+1} = 2\beta_k$).
end for

provides a general framework to (try to) do so.[5] Many refinements are possible: we only consider a simple version.

The main idea is to modify the quadratic penalty method as follows:

1. We minimize $L_\beta(\cdot, \mu)$ instead of F_β.

2. We update an estimate for μ at each iteration.

Say at iteration $k \geq 1$ our current penalty weight is β_k and our current estimate for the Lagrange multipliers is μ^k. We run a minimization algorithm on $L_{\beta_k}(\cdot, \mu^k)$ to find what we hope to be a minimizer x_k. If it is at least a critical point, then we have:

$$0 = \nabla_x L_{\beta_k}(x_k, \mu^k) = \nabla f(x_k) + \sum_{i=1}^{p} (\mu_i^k + \beta_k h_i(x_k)) \nabla h_i(x_k). \tag{10.31}$$

Notice that if x_k were feasible, then the above equation would tell us that it is a KKT point with Lagrange multipliers $\mu^k + \beta_k h(x_k)$. This is one way (among many others)[6] to motivate the update

$$\mu^{k+1} = \mu^k + \beta_k h(x_k) \tag{10.32}$$

in Algorithm 10.2. As a sanity check, observe that if x_k is indeed feasible, then $\mu^{k+1} = \mu^k$. If all goes well, μ^{k+1} is close to μ^*, in which case we have:

$$h(x_k) \approx \frac{1}{\beta_k}(\mu^* - \mu^k). \tag{10.33}$$

[5] Originally, ALM for equality constraints was called the "method of multipliers." It was invented by Hestenes (1969) and Powell (1969). Powell's viewpoint was that the ALM subproblem is a shifted quadratic penalty problem: we discuss this later.

[6] Another example: we can think of $\mu^{k+1} = \mu^k + \beta_k h(x_k)$ as a gradient ascent step on the Lagrangian with respect to the variable μ and with step-size β_k. This leads to an interpretation of ALM as a primal-dual method, where we aim to minimize the (augmented) Lagrangian in the primal variable x and to maximize it in the dual variable μ.

Thus, we see that if μ^k is already quite close to μ^*, then x_k is nearly feasible. In particular, we can hope that points x_k generated by ALM may be much closer to being feasible than points generated by the quadratic penalty method with the same penalty weight β_k (which corresponds to $\mu^k = 0$).

Remark 10.19. *We can also connect ALM with Theorem 10.10. Indeed, notice that we can re-arrange the augmented Lagrangian as follows:*

$$L_\beta(x, \mu) = f(x) + \frac{\beta}{2} \sum_{i=1}^{p} (\mu_i/\beta + h_i(x))^2 - \frac{1}{2\beta} \sum_{i=1}^{p} \mu_i^2. \tag{10.34}$$

The term $-\frac{1}{2\beta} \sum_{i=1}^{p} \mu_i^2$ *is constant with respect to* x*. Thus, minimizing* $L_\beta(x, \mu)$ *with respect to* x *is* equivalent *to minimizing the quadratic penalty function* $\tilde{F}_\beta$ *for the* shifted *constrained problem*

$$\min_{x \in \mathcal{E}} f(x) \text{ subject to } \tilde{h}(x) \triangleq h(x) + \frac{1}{\beta}\mu = 0. \tag{10.35}$$

We know from our discussion of quadratic penalty methods that at a minimizer $\tilde{x}$ *of* $\tilde{F}_\beta$ *we can expect the following, provided* β *is large enough:*

$$\tilde{h}(\tilde{x}) \approx \frac{1}{\beta}\mu^*,$$

where μ^* *are the Lagrange multipliers of the constrained problem (either the original one or the shifted one: they are almost the same for large* β*). By definition of* $\tilde{h}$*, this is equivalent to the statement*

$$h(\tilde{x}) \approx \frac{1}{\beta}(\mu^* - \mu).$$

This recovers the approximate feasibility estimate (10.33). By the same logic, we also obtain

$$\mu^* \approx \mu + \beta h(\tilde{x}),$$

which recovers the update equation for μ *(10.32).*

Remark 10.20. *This is another side note about notation. The constraint map* $h \colon \mathcal{E} \to \mathbb{R}^p$ *goes from one Euclidean space to another. We use* $\langle \cdot, \cdot \rangle$ *and* $\| \cdot \|$ *to denote the inner product and associated norm on* $\mathcal{E}$*. Without fear of ambiguity, we can use the same notation for the (usual) inner product and norm on* $\mathbb{R}^p$*: context always indicates clearly which one we mean. This allows us to use concise notation as follows:*

$$L_\beta(x, \mu) = f(x) + \langle \mu, h(x) \rangle + \frac{\beta}{2} \|h(x)\|^2. \tag{10.36}$$

The directional derivatives with respect to x (with μ fixed) are given by

$$\mathrm{D}(x \mapsto L_\beta(x,\mu))(x)[v] = \mathrm{D}f(x)[v] + \langle \mu, \mathrm{D}h(x)[v] \rangle + \beta \langle \mathrm{D}h(x)[v], h(x) \rangle .$$

The adjoint of $\mathrm{D}h(x)\colon \mathcal{E} \to \mathbb{R}^p$ is the linear map $\mathrm{D}h(x)^*\colon \mathbb{R}^p \to \mathcal{E}$ using which we can rewrite the above as:

$$\mathrm{D}(x \mapsto L_\beta(x,\mu))(x)[v] = \langle \nabla f(x), v \rangle + \langle \mathrm{D}h(x)^*[\mu], v \rangle + \beta \langle v, \mathrm{D}h(x)^*[h(x)] \rangle .$$

By identification, it follows that the gradient of L_β with respect to x is simply:

$$\nabla_x L_\beta(x,\mu) = \nabla f(x) + \mathrm{D}h(x)^*[\mu + \beta h(x)]. \tag{10.37}$$

This connects directly with the usual formulas. Indeed,

$$\mathrm{D}h(x)[v] = \begin{bmatrix} \mathrm{D}h_1(x)[v] \\ \vdots \\ \mathrm{D}h_p(x)[v] \end{bmatrix} = \begin{bmatrix} \langle \nabla h_1(x), v \rangle \\ \vdots \\ \langle \nabla h_p(x), v \rangle \end{bmatrix},$$

so that, for all $v \in \mathcal{E}$ and $z \in \mathbb{R}^p$,

$$\langle \mathrm{D}h(x)^*[z], v \rangle = \langle z, \mathrm{D}h(x)[v] \rangle = \sum_{i=1}^p z_i \langle \nabla h_i(x), v \rangle = \left\langle \sum_{i=1}^p z_i \nabla h_i(x), v \right\rangle .$$

In other words,

$$\mathrm{D}h(x)^*[z] = \sum_{i=1}^p z_i \nabla h_i(x).$$

Notation as above is sometimes convenient as it avoids tedious sum notation.

10.5 Augmented Lagrangian methods, general case

Consider again the more general case of a constrained optimization problem which includes both equality and inequality constraints:

$$\min_{x \in \mathcal{E}} f(x) \text{ subject to } h(x) = 0 \text{ and } g(x) \leq 0.$$

There are several ways to modify ALM as presented above to handle the inequality constraints as well. We discuss two possibilities.

Bound-constrained formulation via slack variables

A simple idea to handle general inequality constraints is to replace them with simple bound constraints via the introduction of slack variables. Explicitly, we consider the following bound-constrained optimization problem:

$$\min_{x\in\mathcal{E},y\in\mathbb{R}^m} f(x) \text{ subject to } h(x)=0, y+g(x)=0 \text{ and } y\geq 0. \tag{10.38}$$

The idea is then to use ALM on this problem but keeping the bound constraints on y "as is," that is: with given μ,λ, we minimize

$$L_\beta(x,y,\mu,\lambda) = f(x)+\mu^\top h(x)+\lambda^\top(y+g(x)) \\ +\frac{\beta}{2}\left(\sum_{i=1}^p h_i(x)^2+\sum_{i=1}^m (y_i+g_i(x))^2\right) \tag{10.39}$$

with respect to $x\in\mathcal{E}$ and $y\in\mathbb{R}^m_+$.

To minimize $(x,y)\mapsto L_\beta(x,y,\mu,\lambda)$ under the constraints $y\geq 0$, one possibility is to use the projected gradient descent method from Section 10.1 with a line-search procedure. This is relatively straightforward to do because projecting to $\mathbb{R}^m_+$ is trivial. More sophisticated algorithms can offer better performance [NW06, §16.7, §18.6].

In the outer iteration of ALM, multipliers can be updated as usual with

$$\mu\leftarrow\mu+\beta h(x),\\ \lambda\leftarrow\lambda+\beta(y+g(x)).$$

See [NW06, Thm 17.4] for details and refinements. These ideas form the basis for successful optimization software used in industrial settings.

Augmented Lagrangian as a shifted penalty

Recall from (10.34) that we could have motivated the Augmented Lagrangian function as a quadratic penalty function for an optimization problem whose constraints are *shifted* according to the conclusions of Theorem 10.10:

$$L_\beta(x,\mu)=f(x)+\frac{\beta}{2}\sum_{i=1}^p(\mu_i/\beta+h_i(x))^2-\frac{1}{2\beta}\sum_{i=1}^p\mu_i^2.$$

Since Theorem 10.10 also applies for inequality constraints, we can use the same idea to introduce the augmented Lagrangian function for the general optimization problem involving both equality and inequality constraints. This

Algorithm 10.3 Augmented Lagrangian method (general constraints)

Pick an initial guess $x_0 \in \mathcal{E}$ and initial multipliers $\mu^1 \in \mathbb{R}^p$, $\lambda^1 \in \mathbb{R}^m_+$.
Pick an initial weight $\beta_1 > 0$.
for k in $1, 2, 3 \ldots$ **do**
    Compute x_k by applying a minimization algorithm to $L_{\beta_k}(\cdot, \mu^k, \lambda^k)$ as in (10.40), initialized at x_{k-1} (a *warm start*).
    Set $\mu^{k+1} = \mu^k + \beta_k h(x_k)$.
    Set $\lambda^{k+1} = \max(0, \lambda^k + \beta_k g(x_k))$.
    Set $\beta_{k+1} \geq \beta_k$ (e.g.: $\beta_{k+1} = 2\beta_k$).
end for

leads to a shifted penalty function inspired by Theorem 10.10:

$$L_\beta(x, \mu, \lambda) = f(x) + \frac{\beta}{2} \sum_{i=1}^{p} \left(\frac{\mu_i}{\beta} + h_i(x) \right)^2 + \frac{\beta}{2} \sum_{i=1}^{m} \max\left(0, \frac{\lambda_i}{\beta} + g_i(x)\right)^2 - \frac{1}{2\beta} \sum_{i=1}^{p} \mu_i^2 - \frac{1}{2\beta} \sum_{i=1}^{m} \lambda_i^2. \quad (10.40)$$

By arguments similar to the ones we had for equality constrained problems, we can justify the following update rules for μ and λ:

$$\mu^{k+1} = \mu^k + \beta_k h(x^k), \quad (10.41)$$

$$\lambda^{k+1} = \max(0, \lambda^k + \beta_k g(x_k)). \quad (10.42)$$

The resulting algorithm is listed as Algorithm 10.3.

Bibliography

[Ber95] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.

[BV04] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[CGT00] A.R. Conn, N.I.M. Gould, and P.L. Toint. *Trust-region methods*. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2000.

[CGT12] C. Cartis, N.I.M. Gould, and P.L. Toint. Complexity bounds for second-order optimality in unconstrained optimization. *Journal of Complexity*, 28(1):93–108, 2012.

[NW06] J. Nocedal and S. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer Science & Business Media, 2 edition, 2006.

[Roc70] R.T. Rockafellar. *Convex analysis*. Princeton University Press, Princeton, NJ, 1970.

[Rus06] A.P. Ruszczyński. *Nonlinear optimization*. Princeton University Press, Princeton, NJ, 2006.

[RW98] R.T. Rockafellar and R.J.B. Wets. *Variational Analysis*. Springer Berlin Heidelberg, 1998.