Introductory Nonlinear Optimization:

Lecture Notes for Math/CS 435/535, WWU

Richard C Barnard

Winter 2019

Contents

Chapter 1

Introduction

1.1 Examples

Constrained Optimization

- Suppose we can purchase a combination of two risky assets (say non-dividend paying stocks) with expected monthly returns r_A and r_B. If x_A and x_B are the amount of each asset we purchase, the expected return is $x_A r_A + x_B r_B$. Naturally we want to maximize the return but without making things too risky. One way is to require that the variance of the return

  $$\sigma_A x_A^2 + \sigma_B^2 x_B^2 + 2\sigma_{AB} x_A x_B$$

  stays below a personal tolerance we pick (here $\sigma_A, \sigma_B, \sigma_{AB}$ are given parameters, namely the variances of the stocks and joint variance).

- A shopper has I money to spend on 3 items; a possible utility (informally the "happiness" function) they receive from their purchase is given by $U(x_1, x_2, x_3) = x_1x_2 + x_2x_3 + x_1x_2x_3$. They then want to maximize this subject to $p_1x_1 + p_2x_2 + p_3x_3 = I$ where p_i is the price of the i-th good.

Numerical Optimization Commonly, practical optimization problems require us to solve problems where closed-form or exact solutions are unavailable or not readily obtained.

- Suppose we want to minimize the function

$$f(x_1, x_2) = x_1^8 + x_2^6 + x_1^4 x_2^2 + 4x_1 x_2 + 1.$$

- The radiotherapy treatment planning problem involves finding a combination of beams of radiative particles which gives a dose of radiation as close to a desired dose profile D as possible without damaging healthy tissue. One simplified version of this problem is as follows. After dividing the patient's body into voxels, we are given:

  1. the desired dose D_{ijk} on each voxel;
  2. H, the set of (i, j, k) where the tissue is healthy;
  3. the finite set of beam forms q_n;
  4. the matrices M_n such that $(M_n q_n)_{ijk}$ is the dose the n-th beam gives to the (i, j, k)-th voxel;
  5. the maximum dose $U_{i,j,k}$ which kills tissue on each voxel.

  Then we want to find numbers $c_1, \ldots, c_N$ which minimizes

$$\begin{aligned} J(c_1, \ldots, c_N) = &\sum_{i,j,k} \Big(\Big[\sum_{n=1}^{N} c_n M_n q_n \Big]_{ijk} - D_{ijk} \Big)^2 \\ &+ \sum_{(i,j,k) \in H} \log(U - \Big[\sum_{n=1}^{N} c_n M_n q_n \Big]_{ijk}) \\ &+ \sum_{n=1}^{N} (\log c_n). \end{aligned}$$

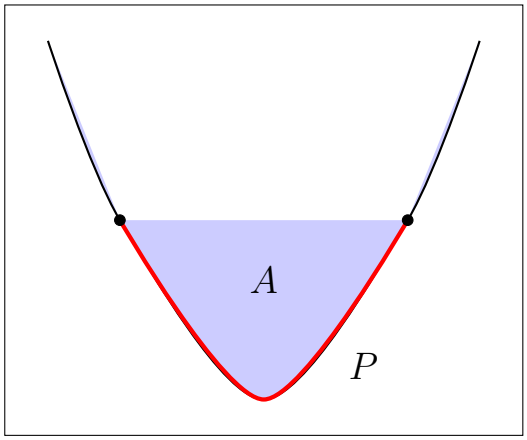

Figure 1.1: Channel design problem

Variational Problems Variational problems involve finding, rather than an optimal vector $\vec{x}^* \in \mathbb{R}^n$, an optimal function withing a space of allowed functions.

- Given a starting point $(0, A)$ and ending point $(B, 0)$, and assuming that gravity is the only active force (that is we ignore friction), what is the track $T : [0, 1] \to \mathbb{R}^2$ between the starting and ending points (i.e. $T(0) = (0, A)$ and $T(1) = (B, 0)$) along which a ball can slide in the shortest time possible?

- We want to design a channel which minimizes the friction between the fluid and the walls of the channel, while maintaining a cross section with area A. Assuming the friction is proportional to the amount of contact between the fluid and the wall, we want to minimize P as shown in section 1.1

Convex Analysis Problems may arise where we want to minimize a *nondifferentiable* function which is *convex* (intuitively, every secant line lies above the graph of the function):

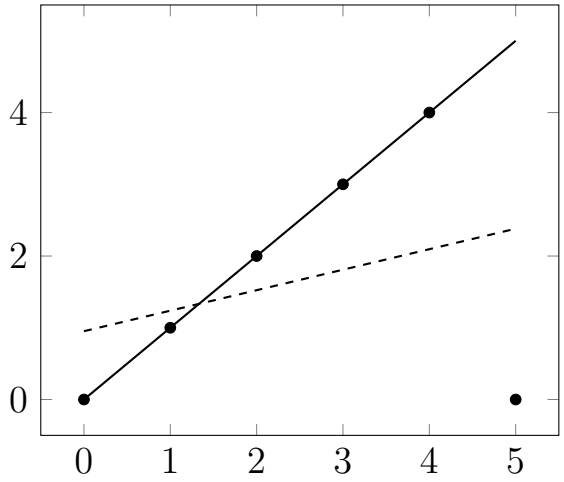

Figure 1.2: Fitting lines to data set

- Suppose that we want to fit a line to a data set. That is, we are given a list of measurements $(x_0, y_0), \ldots (x_N, y_N)$ where x_i is the value of the independent variable and y_i is the measured quantity. We want to find the "best" line $mx+b$ which minimizes the errors $e_i = |mx_i + b - y_i|$. One notion of "best" would come from picking m and b which minimizes

$$J(m, b) = \sum_{i=1}^{N} e_i,$$

which is a nondifferentiable function. This is in contrast to least-squares fitting, which would look to minimize $\sum_{i=1}^{N}(e_i)^2$. However, in the case of section 1.1, we show the data points, the dashed line from using least-squares , and the solid line $y = mx + b$ where m and b come from minimizing $J(m, b)$. As we can see, minimizing J gives a very different result which may (may not) be better for our purposes.

1.2 Notation

We shall use the following notation throughout the course:

- A vector $\vec{x} \in \mathbb{R}^n$ is always viewed to be a column vector. We write $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ or,equivalently, $\vec{x} = [x_1, x_2, \ldots, x_n]^T$.

- The derivative of a function of one variable $f : \mathbb{R} \to \mathbb{R}^n$ shall be denoted by $\dot{f}$.

- The gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ at the point $\vec{x}$ is denoted $\nabla f(\vec{x})$ and is a column vector.

- The differential of a function $h : \mathbb{R}^n \to \mathbb{R}^m$ is denoted $Dh(\vec{x})$. It is an $m \times n$ matrix given by

$$Dh(\vec{x}) = \begin{bmatrix} \nabla h_1(\vec{x})^T \\ \nabla h_2(\vec{x})^T \\ \vdots \\ \nabla h_m(\vec{x})^T \end{bmatrix} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h_m}{\partial x_1} & \frac{\partial h_m}{\partial x_2} & \cdots & \frac{\partial h_m}{\partial x_n} \end{bmatrix}$$

- If $F : \mathbb{R}^n \to \mathbb{R}^n$ is a vector-valued function, its divergence is given as $\text{div}F = \nabla \cdot F = \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2} + \cdots + \frac{\partial F_n}{\partial x_n}$.

- The open ball of radius $\varepsilon \geq 0$ centered at $\vec{x}$ is denoted by $B_\varepsilon(\vec{x})$ and is defined as

$$B_\varepsilon(\vec{x}) = \{\vec{y} : \|\vec{x} - \vec{y}\| < \varepsilon\}.$$

  If the ball is centered at the origin, $\vec{x} = \vec{0}$, we simply write B_ε.

- For a surface S in $\mathbb{R}^n$ and $\vec{x} \in S$, the tangent space to S at $\vec{x}$ is denoted $T_{\vec{x}}S$.

- For a linear operator $A : V \to W$ between vector spaces, its null-space is denoted by $\mathcal{N}(A)$ and its range is denoted by $\mathcal{R}(A)$.

- A function f which has its first k derivatives continuous at every point $x \in \Omega$ for some set Ω is denoted as $f \in C^k(\Omega)$. If k can be arbitrarily large and we still have $f \in C^k(\Omega)$, then $f \in c^\infty(\Omega)$.

Until the last chapter, any function, unless otherwise stated, is assumed to be differentiable.

When discussing iterative methods, a function evaluated at the k-th iterate x_k will be notated by $f(x_k) = f_k$. Finally, parts of the text which are highlighted such as this sentence contain statements which you should check and verify for yourself as exercises.

Chapter 2

Constrained Optimization Theory

Constrained optimization problems are, very generally, problems of the form

$$\min_{x \in V} f(x) \text{ such that } x \in C \subseteq V.$$

Here V is some vector space which contains *decision variables* denoted as x, $f : V \to \mathbb{R}$ is the *objective function* which we want to minimize by making an optimal choice of decision variables from the *feasible set* C. We generally will use x^* to denote **a**n *minimizer* or *optimum* to this optimization problem; note we are not assured that this is unique (or even that it exists!) for any particular problem. Concretely, a *global optimum* x^* is such that $f(x^*) \leq f(y)$ for any other feasible $y \in C$. A *local optimum* x^* is such that $f(x^*) \leq f(y)$ for any $y \in C \cap B_r(x^*)$ for some $r > 0$. For convenience, in this chapter we will use "optimum" and "minimizer" to refer to local optima only. Global optima satisfy the same necessary conditions, but we typically can only give local criteria for optimality; we will discuss the major exception to this later. Typically we form C to restrict our available choices to x which are "reasonable" in some sense (e.g. x_i represents mass and so should not be a negative quantity). Notice that even though we phrase the optimization problem as a minimization problem, problems involving maximization are covered since maximizing f is equivalent

to minimizing $-f$.

The above optimization problem is very general and too broad to get very many practical results. To make things more concrete, in the remainder of this chapter:

- we will assume $V = \mathbb{R}^n$, and so we will refer to a selection of decision variables as a "point";
- the feasible set will be formed from *constraint functions*:

  $$C = \{\vec{x} : h(\vec{x}) = 0 \text{ and } g(\vec{x}) \leq 0\}$$

  where $h : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^p$. Note this means we really have m equality constraints of the form $h_j(\vec{x}) = 0$ and p inequality constraints of the form $g_j(\vec{x}) \leq 0$.

2.1 Problems with equality constraints

First, let's consider problems without inequality constraints, i.e.

$$\min_{\vec{x} \in \mathbb{R}^n} f(\vec{x}) : \mathbb{R}^n \to \mathbb{R} \text{ such that } h(\vec{x}) = 0. \tag{2.1}$$

Note that when $n = 2$, it usually only makes sense to consider a single constraint (i.e. h is scalar-valued). This is because zero-sets of functions are curves which, if they intersect, either intersect at finitely many points or coincide. In the former case, we just have to check f at those points and in the latter, the constraints collapse to a single constraint. But in higher dimensions where $n > 2$, the zero-sets can intersect in a curve, or surface, or hyper-surface.

Let's recall the following geometric property of gradients, which is relevant to much of this course. For any differentiable function h, and one of its level sets $S = \{\vec{x} : h(\vec{x}) = c\}$, if $\vec{y} \in S$ then $\nabla h(\vec{y})$ is perpendicular to the tangent line/ plane/ hyperplane (depending on the dimension) to S at $\vec{y}$. More precise is the following result.

Lemma 2.1.1. *Suppose we have an arbitrary smooth curve $x(t) : [-\varepsilon, \varepsilon] \to \mathbb{R}^n$ which is such that $x(t) \in S$ for any t and $x(0) = \vec{y}$. Then $\frac{d}{dt}x(0)$ is perpendicular to $\nabla h(\vec{y})$.*

Proof. If $x(t)$ is such a curve, since $h(x(t)) = c$ for all t, we have $\frac{d}{dt}h(x(t)) = 0$. But the chain rule says that for any $t \in [-\varepsilon, \varepsilon]$

$$\frac{d}{dt}h(x(t)) = \nabla h(x(t))^T x(t)$$

meaning that $\nabla h(\vec{y})$ is perpendicular to $x'(0)$. □

Example 2.1.1. *Consider $h(x) = x_1^2 + x_2^2 - 1$ with $S = \{h(x) = 0\}$ (i.e. S is the unit circle). Then the family of curves given by $x_\alpha(t) := [\cos(\alpha t + \frac{\pi}{6}), \sin(\alpha t + \frac{\pi}{6})]^T$ lies in S and $x_\alpha(0) = (\frac{\sqrt{3}}{2}, \frac{1}{2})^T$. This means $\dot{x}_\alpha(0)$ is tangent to S at $x_\alpha(0)$. Now,*

$$\dot{x}_\alpha(0) = \alpha \begin{pmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{pmatrix}, \quad \nabla h(\frac{\sqrt{3}}{2}, \frac{1}{2}) = \begin{pmatrix} \sqrt{3} \\ 1 \end{pmatrix}.$$

Note for any α that $\nabla h((\frac{\sqrt{3}}{2}, \frac{1}{2}))^T \dot{x}(0) = 0$. However this is only one family of curves, rather than all curves remaining in S.

The Lagrange Multiplier Theorem

Let's focus for the moment on the case where there is only one equality constraint; that is, we have $m = 1$ and so $h(x) = 0$ is a single equation, rather than a system. Suppose we know that $\vec{x}^*$ is an optimum for (2.1) and we again have a curve $x(t) : [-\varepsilon, \varepsilon] \to \mathbb{R}^n$ such that $x(t)$ remains in $\{h(\vec{x}) = 0\}$ and $x(0) = \vec{x}^*$. Then as a function of t, we know that $f(x(t))$ has a minimum at $t = 0$. So, by the chain rule

$$0 = \frac{d}{dt}f(x(t))|_{t=0} = \nabla f(x(0))^T \dot{x}(0) = \nabla f(\vec{x}^*)^T \dot{x}(0).$$

So now we know that $\nabla f(\vec{x}^*)$ and $\nabla h(\vec{x}^*)$ are both orthogonal to the same vector. If the *codimension* of $\{h(x) = 0\}$ is 1 (e.g. the set is a curve when $n = 2$, the set is a surface when $n = 3$,etc.), then they are parallel to each other, meaning that there is some $\lambda \in \mathbb{R}$ such that

$$\nabla f(\vec{x}^*) + \lambda \nabla(\vec{x}^*) = 0.$$

The constant λ is called a *Lagrange multiplier*; note that we have a necessary condition for optimality via its existence.

Example 2.1.2. *Consider the problem where we want to minimize* $f(x) = x_1 - x_2$ *over the set* $h(x) = \{h(x) = 0\}$ *where* $h(x) = x_1^2 + x_2^2 - 1$. *Then at an optimizer we have*

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} + \lambda \begin{bmatrix} 2x \\ 2y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

So we we need to find a point where $2\lambda(x + y) = 0$. *If* $\lambda = 0$, *we get a contradiction, so we have* $x_1 = x_2$. *Since we need the point to be feasible, we have* $2x_1^2 = 1$ *and so we have the points* $[1/\sqrt{2}, -1/\sqrt{2}]^T$ *and* $[-1/\sqrt{2}, 1/\sqrt{2}]^T$ *as possible optimizers. Evaluating* f *at these points shows that the latter is the optimizer.*

Example 2.1.3. *We want to minimize* $f(x) = x_1^2 - 2x_1x_2 + x_2^2$ *subject to* $x_1^2 + x_2^2 = 16$. *Then we get* f *at the candidate points and associated* λ *as*

$$\begin{aligned} f(-2\sqrt{2}, -2\sqrt{2}) = 0, &\lambda = 0, \\ f(2\sqrt{2}, 2\sqrt{2}) = 0, &\lambda = 0, \\ f(2\sqrt{2}, -2\sqrt{2}) = 32, &\lambda = -2, \\ f(-2\sqrt{2}, 2\sqrt{2}) = 32, &\lambda = -2. \end{aligned}$$

Before moving to the general form of the Lagrange multiplier theorem, it is useful to get a sense of what a multiplier tells us. Consider a general, parameterized version of (2.1):

$$\min_{\vec{x}\in\mathbb{R}^n} f(\vec{x}) \text{ such that } h(\vec{x}) = c$$

where now c is a parameter we can adjust, but we still only have a single equality constraint. As c changes the minimizer will also change. If we pick one minimizer for each value of c (since we might have multiple minimizers, this may be necessary), then we have a function $\vec{x}^*(c)$ where $\vec{x}^* : \mathbb{R} \to \mathbb{R}^n$ and $\vec{x}^*(0) = \vec{x}^*$ is the minimizer for the original problem we started with. The question we next ask is: near $c = 0$, how does the minimum value of the objective change as we change the constraint set? That is, we look at

$$\frac{d}{dc} f(\vec{x}^*(c))|_{c=0} = Df(\vec{x}^*(0))D\vec{x}^*(0) = -\lambda Dh(\vec{x}^*(0))D\vec{x}^*(0)$$

where we should emphasize that $Df(\vec{x}^*(0))$ is the differential with respect to the variable $\vec{x}$ evaluated at $\vec{x}^*(0)$ and $D\vec{x}^*(0)$ is the differential with respect to c evaluated at $c = 0$.

Next, since $\vec{x}^*(c)$ satisfies the constraint, $h(\vec{x}^*(c)) = c$. The chain rule applied to this equation with respect to c gives us

$$1 = Dh(\vec{x}^*(0))D\vec{x}^*(0)$$

which means that $\frac{d}{dc} f(\vec{x}^*(c))|_{c=0} = -\lambda$. In other words, the Lagrange multiplier tells us locally how much the minimal value of the objective changes as we adjust the feasible set.

Now we're ready to state the Lagrange Multiplier Theorem.

Theorem 2.1.4. *Let $\vec{x}^*$ be a local minimizer of (2.1) where $m \leq n$. Assume that the vectors $\nabla h_i(\vec{x}^*)$ are linearly independent (i.e. $Dh(\vec{x}^*)$ has row rank m). Then there exists a $\vec{\lambda} \in \mathbb{R}^m$ such that*

$$Df(\vec{x}^*) + \vec{\lambda}^T Dh(\vec{x}^*) = \vec{0}. \tag{2.2}$$

Remark 1. *Note that another way of writing* (2.2) *is*

$$\nabla f(\vec{x}^*) = -\sum_{i=1}^{m} \lambda_i \nabla h_i(\vec{x}^*)$$

which means that at the optimal choice of $\vec{x}^*$*, we have that the gradient of* f *is in the span of the gradients of the* h_i*. Intuitively, informally, and roughly speaking, this boils down to: the independent gradients of* h_i *span a hyperplane perpendicular to the the intersection of the constraints; if* ∇f *is not in this hyperplane, then we could move along the component of* ∇f *not in the span, which also moves along the intersection of the constraints, violating the optimality we've assumed.*

Tangent Spaces

Now that we are looking at constraint sets which are m-dimensional, rather than just a curve, we need to introduce the tangent space to a set at a given point.

Definition 2.1.5. *Given a set* $S \subset \mathbb{R}^m$ *and* $\vec{x} \in S$*, a vector* $\vec{v}$ *is a* tangent *vector to* S *at* $\vec{x}$ *if and only if there exists a curve* $\vec{x}(t) : (-\varepsilon, \varepsilon) \to S$ *for some* $\varepsilon > 0$ *such that* $\vec{x}(0) = \vec{x}$ *and* $\vec{v} = \vec{x}'(0)$*. The* tangent space $T_{\vec{x}}S$ *to* S *at* x *is the set of all vectors which are tangent to* S *at* $\vec{x}$*.*

The tangent space $T_{\vec{x}}S$ is a vector space which can be and often should be viewed as the tangent hyperplane to S at $\vec{x}$ with a coordinate system that places $\vec{x}$ as its origin.

In the special case where $S = \{h(\vec{x}) = c\}$ for some $c \in \mathbb{R}$, we have that $T_{\vec{x}^*}S \subset \mathcal{N}(Dh(\vec{x}^*))$. Recall that $v \in \mathcal{N}(Dh(\vec{x}^*))$ if $Dh(\vec{x}^*)v = 0$. So $\mathcal{N}(Dh(\vec{x}^*))$ is the space of vectors orthogonal to each $\nabla h_j(\vec{x}^*)$. In the case where $Dh(\vec{x}^*)$ is full rank, then

$$T_{\vec{x}^*}S = \mathcal{N}(Dh(\vec{x}^*)).$$

The assumption that Dh is full rank is simply that S will change if we delete any h_j from the definition; that is, we are assuming that no constraints are redundant.

A useful and needed geometric fact in the proof of Theorem 2.1.4 is: for any matrix A, $\mathcal{R}(A^T) = \mathcal{N}(A)^\perp$. Here, for a subspace V, $V^\perp = \{w : \langle w, v\rangle = 0, \text{ for all } v \in V\}$.

Proof of Theorem 2.1.4: In order to prove (2.2), we need to show that for some $\vec{\lambda}$ that $\nabla f(\vec{x}^*) = -Dh(\vec{x}^*)^T\vec{\lambda}$. This says that $\nabla f(\vec{x}^*) \in \mathcal{R}(Dh(\vec{x}^*)^T)$. Note that for $Dh(\vec{x}^*)^T : \mathbb{R}^m \to \mathbb{R}^n$, if $S = \{\vec{x} : h(\vec{x}) = 0\}$, we have

$$\mathcal{R}(Dh(\vec{x}^*)^T) = \mathcal{N}(Dh(\vec{x}^*)) = (T_{\vec{x}^*}S)^\perp.$$

Let $v \in (T_{\vec{x}^*}S)$ and $x(t) : (-\varepsilon, \varepsilon) \to S$ be a curve such that $x(0) = \vec{x}^*$ and $\dot{x}(0) = v$. Then, since $\vec{x}^*$ is a optimum on S

$$0 = \frac{d}{dt} f(x(t))|_{t=0} = Df(\vec{x}^*)Dx(0) = \nabla f(\vec{x}^*)^T v.$$

This means that $\nabla f(\vec{x}^*)$ is orthogonal to v; since this holds for any $v \in T_{\vec{x}^*}S$, we have that $\nabla f(\vec{x}^*) \in (T_{\vec{x}^*}S)^\perp$. □

Remark 2. *If we don't have the full-rank assumption on the constraint set, there are ready examples to show that* (2.2) *fails. For instance, suppose we have* $f(x) = x_1 + x_3$ *with constraints* $h_1 = x_3$ *and* $h_2 = x_1^3 + x_3$. *Then the feasible set is the* x_2 *axis along which* f *is constant, meaning every feasible point is a minimizer. But at* $x_1 = x_2 = x_3 = 0$, $\nabla f^T = [1, 0, 1]$ *while for any* $\lambda \in \mathbb{R}^2$, $\lambda^T Dh(0) = [0, 0, \lambda_1 + \lambda_2]$ *meaning no* λ *exists which satisfies* (2.2).

Example 2.1.6. *Suppose* $f(x) = x_1^3 + x_2^3$ *and* $h(x) = x_1^2 + x_2^2 - 1$. *Then*

at an optimum x^*, there is a $\lambda \in \mathbb{R}$ such that

$$\begin{aligned} 0 &= 3x_1^2 + 2\lambda x_1 && \Rightarrow \quad 0 = x_1(3x_1 + 2\lambda) \\ 0 &= 3x_2^2 + 2\lambda x_2 && \Rightarrow \quad 0 = x_2(3x_2 + 2\lambda) \\ 4 &= x_1^2 + x_2^2 \end{aligned}$$

Working through the cases from first two equations:

- *Suppose $x_1 = 0$. Then if $x_2 = 0$, the constraint is not satisfied. So the second equation means $x_2 = -\frac{2\lambda}{3}$; this means $\lambda = \pm 3$ and so $x_2 = \mp 2$.*

- *Suppose $x_1 = -\frac{2\lambda}{3}$. Then*

  - *if $x_2 = 0$, $\lambda = \pm 3$ from the constraint, which means $x_1 = \mp 2$;*
  - *if $x_2 = -\frac{2\lambda}{3}$, the constraint means $\lambda = \pm\frac{3}{\sqrt{2}}$ and so $x_1 = x_2 = \mp\sqrt{2}$.*

  So (2.2) gives us the following candidates for optima: $(0, 2)$, $(2, 0)$, $(\sqrt{2}, \sqrt{2})$, $(-\sqrt{2}, -\sqrt{2})$, $(0, 2)$, and $(0, -2)$. Evaluating f at these points shows that $f(0, -2) = (-2, 0) = -8$ is the smallest value of f among these points and so we have the two minima.

Example 2.1.7. *Let $f(x) = 3x_1x_3 + 4x_2x_3$ and the constraints be given by $h_1(x) = x_2^2 + x_3^2 - 4$ and $h_2(x) = x_1x_3 - 3$. The constraint set is shown in Figure 2.1. We need*

$$\begin{pmatrix} 3x_3 & 4x_3 & 3x_1 + 4x_2 \end{pmatrix} + \begin{pmatrix} \lambda_1 & \lambda_2 \end{pmatrix} \begin{pmatrix} 0 & 2x_2 & 2x_3 \\ x_3 & 0 & x_1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}.$$

Unpacking this, we get the equations (including the constraints)

$$\begin{aligned} 3x_3 + x_3\lambda_2 &= 0 \\ 4x_3 + 2x_2\lambda_1 &= 0 \\ 3x_1 + 4x_2 + 2x_3\lambda_1 + x_1\lambda_2 &= 0 \\ x_2^2 + x_3^2 &= 4 \\ x_1x_3 &= 3 \end{aligned}$$

The first equation implies either $x_3 = 0$ (which results in a violation of $h_2 = 0$) or $\lambda_2 = -3$. So the second and third equations become, respectively, $4x_3 + 2x_2\lambda_1 = 0$ and $4x_2 + 2x_3\lambda_1 = 0$. This results in $x_2 = \pm x_3$. If $x_2 = x_3$, the h_1 constraint gives $x_2 = x_3 = \pm\sqrt{2}$ and the h_2 constraint gives $x_1 = \pm\frac{3}{\sqrt{2}}$. If, on the other hand, $x_2 = -x_3$, we get $x_2 = \pm\sqrt{2}$, $x_1 = \mp\sqrt{2}$ and $x_1 = \mp\frac{3}{\sqrt{2}}$. Checking, we get

$$f(\frac{3}{\sqrt{2}}, \sqrt{2}, \sqrt{2}) = 17, \qquad f(-\frac{3}{\sqrt{2}}, -\sqrt{2}, -\sqrt{2}) = 17,$$
$$f(-\frac{3}{\sqrt{2}}, \sqrt{2}, -\sqrt{2}) = 1, \qquad f(\frac{3}{\sqrt{2}}, -\sqrt{2}, \sqrt{2}) = 1.$$

We can't conclude yet that we have the optima (the latter two points evaluated above). Why? Note that, unlike in the previous example, the constraint set is not compact *(in $\mathbb{R}^n$ this means closed and bounded); if it were compact, we'd be able to conclude that the optima were found. Note that the constraint set doesn't result in bounded values for x_1. So, we check what happens as $x_1 \to \pm\infty$. Since $x_1x_3 = 3$ at every feasible point, we can rewrite f on at any feasible point as $f(x) = 9 + x_2x_3$; as $x \to \pm\infty$, we have $x_3 \to 0$, meaning $f \to 9$. Since this limit is above the lower values we found, we can conclude that $(-\frac{3}{\sqrt{2}}, \sqrt{2}, -\sqrt{2})^T$ and $(\frac{3}{\sqrt{2}}, -\sqrt{2}, \sqrt{2})^T$ are minimizers.*

It is important for the next remark that we recall the chain rule for vector-valued functions. Let $f(x) : \mathbb{R}^n \to \mathbb{R}$ and $x(c) : \mathbb{R}^m \to \mathbb{R}^n$. If

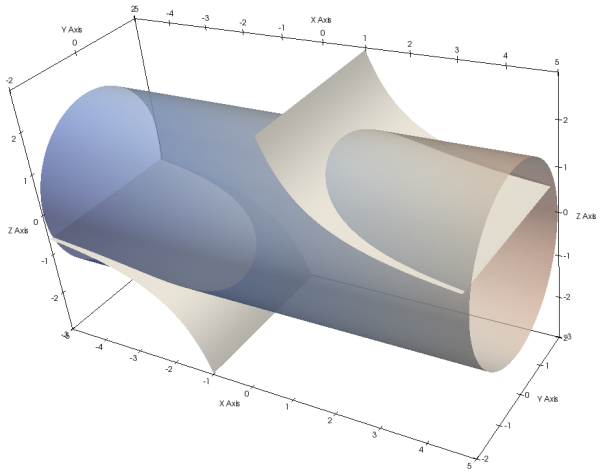

Figure 2.1: The sets $h_1 = 0$ and $h_2 = 0$ from Example 2.1.7, coloring is only for visualization purposes

we set $g(c) := f(x(c)) : \mathbb{R}^m \to \mathbb{R}$, what is $D_c g$? Since $c \in \mathbb{R}^m$, we have that

$$D_c g = \begin{bmatrix} \frac{\partial g}{\partial c_1} & \dots & \frac{\partial g}{\partial c_m} \end{bmatrix}. \tag{2.3}$$

Now, for $1 \leq i \leq m$ we know

$$\frac{\partial g}{\partial c_i} = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial c_i} + \dots + \frac{\partial f}{\partial x_n}\frac{\partial x_n}{\partial c_i} = D_x f(x(c) \begin{bmatrix} \frac{\partial x_1}{\partial c_i} \\ \vdots \\ \frac{\partial x_n}{\partial c_i} \end{bmatrix}.$$

Combining this, we can conclude

$$D_c\big(f(x(c))\big) = D_c g = D_x f(x(c)) \begin{bmatrix} \frac{\partial x_1}{\partial c_1} & \frac{\partial x_1}{\partial c_2} & \dots & \frac{\partial x_1}{\partial c_m} \\ \frac{\partial x_2}{\partial c_1} & \frac{\partial x_2}{\partial c_2} & \dots & \frac{\partial x_2}{\partial c_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_n}{\partial c_1} & \frac{\partial x_n}{\partial c_2} & \dots & \frac{\partial x_n}{\partial c_m} \end{bmatrix} = D_x f(x(c)) D_c x(c).$$

That is, the chain rule for the differential in higher dimensions works just as it does in single variable calculus.

With this chain rule, let us look again at how modifying the constraint set affects the optimal value of f. Suppose, for a given $c \in \mathbb{R}^m$, that $\vec{x}^*(c)$ is a function which gives the optimizer of Equation (2.1) when the constraint is rewritten as $h(x) = c$. Further, let $M : \mathbb{R}^m \to \mathbb{R}$ be defined as $M(c) = f(\vec{x}^*(c))$; the chain rule gives the derivative of M at $c = 0$:

$$DM(0) = Df(\vec{x}^*(0))D\vec{x}^*(0) = -\lambda^T Dh(\vec{x}^*(0))D\vec{x}^*(0).$$

Since $h(x(c)) = c$ for all c, we have

$$Dh(\vec{x}^*(c))D\vec{x}^*(c) = I_{m\times m}.$$

This means that $DM(0) = -\lambda^T$. Since the gradient always points in the direction of maximum increase, increasing M as rapidly as possible would entail changing the constraint in (2.1) to $h(x) = c$ where c is small and in the direction of $-\lambda$. This means that if instead c is perpendicular to λ, the instantaneous change in the minimum value at the new optimizer would be 0.

Example 2.1.8. *Consider the problem of minimizing* $f(x) = x_1^3 + x_2^3 + x_3^3$ *subject to* $x_1^2 + x_2^2 + x_3^2 = 4$, *and* $x_1 + x_2 + x_3 = 1$. *The feasible set is the intersection of the sphere with a plane, i.e. a circle. The Lagrange Multiplier theorem says we need at a minimizer that*

$$\begin{aligned} 3x_1^2 + 2\lambda_1 x_1 + \lambda_2 &= 0 \\ 3x_2^2 + 2\lambda_1 x_2 + \lambda_2 &= 0 \\ 3x_3^2 + 2\lambda_1 x_3 + \lambda_2 &= 0. \end{aligned}$$

This means that if such λ_1, λ_2 *exist, that*

$$A = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix}$$

has zero determinant; otherwise

$$A \begin{bmatrix} 3 \\ 2\lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

would have to have a solution, which can't be $[0,0,0]^T$*. Luckily, because* A *is a Vandermonde matrix,* $|A| = (x_1 - x_2)(x_1 - x_3)(x_2 - x_3)$*. If* $x_1 = x_2$*, then* $x_3 = 1 - 2x_1$ *and* $2x_1^2 + (1 - 2x_1)^2 = 4$ *(by feasibility). This would give* $x_1 = (2 \pm \sqrt{22})/6$ *and the resulting set of solutions to the Lagrange multiplier system (which are feasible) is*

$$[\tfrac{1}{6}(2-\sqrt{22}), \tfrac{1}{6}(2-\sqrt{22}), \tfrac{1}{3}(1+\sqrt{22})]^T \quad [\tfrac{1}{6}(2-\sqrt{22}), \tfrac{1}{3}(1+\sqrt{22}), \tfrac{1}{6}(2-\sqrt{22})]^T$$
$$[\tfrac{1}{6}(2+\sqrt{22}), \tfrac{1}{6}(2+\sqrt{22}), \tfrac{1}{3}(1-\sqrt{22})]^T \quad [\tfrac{1}{6}(2+\sqrt{22}), \tfrac{1}{3}(1-\sqrt{22}), \tfrac{1}{6}(2+\sqrt{22})]^T.$$

The resulting (numerical) λ*'s are*

$$\big[-2.17, -2.55\big]^T, \quad \big[-2.15, -2.55\big]^T, \quad \big[0.17, -.44\big]^T, \quad \big[0.17, -4.11\big]^T.$$

The values of f*, respectively, at these points are* $6.64,\ 6.64,\ 0.91,\ 0.91.$

So, if we now have to minimize, instead f *over the set* $h(x) = [c_1, x_2]^T$*, the sphere has changed radius, and the plane has shifted. Then, if we have, say* $c = [0.5, 0.6]$*, (which is perhaps large but useful for the purposes of illustration), we roughly expect that the minimum value of* f *on the set* $\{h(x) = c\}$ *to be something like*

$$0.91 - \lambda \cdot c = -[0.17, -4.11] \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix} = 3.291$$

while the new minimum value is actually approximately 3.45, which is not a bad estimate when using such a large c*!*

2.2 Problems with inequality constraints

We now turn to expanding the class of problems to those with both equality constraints and inequality constraints. The general form of

such a problem is

$$\min_{\vec{x}\in\mathbb{R}^n} f(\vec{x}) : \mathbb{R}^n \to \mathbb{R} \text{ such that } h(\vec{x}) = 0 \text{ and } g(\vec{x}) \leq 0 \tag{2.4}$$

where the $h : \mathbb{R}^n \to \mathbb{R}^m$ (as before) and $g : \mathbb{R}^n \to \mathbb{R}^p$. Just like in the previous section, $g(\vec{x}) \leq 0$ should be read as a list of p inequalities of the form $g_j(\vec{x}) \leq 0$. As expected, a point $\vec{x} \in \mathbb{R}^n$ is *feasible* for (2.4) if $h(\vec{x}) = 0$ and $g(\vec{x}) \leq 0$.

Definition 2.2.1. *At the feasible point $\vec{x} \in \mathbb{R}^n$, an inequality constraint $g_j(\vec{x}) \leq 0$ is* active *if $g_j(\vec{x}) = 0$ and* inactive *if $g_j(\vec{x}) < 0$. The active set of constraints at this point is the list of indices j for which g_j is active at $\vec{x}$; we will denote this set of indices with $J(\vec{x})$:*

$$J(\vec{x}) = \{j : g_j(\vec{x}) = 0\}.$$

The Karush-Kuhn-Tucker Theorem

As we've generalized the form that the constraint set might take, we will have to replace the Lagrange multiplier theorem with a more general theorem: the Karush-Kuhn-Tucker Theorem (KKT).

Theorem 2.2.2. *Let $\vec{x}^*$ be a local optimum for the problem (2.4) such that $\{\nabla h_i(\vec{x}^*), \nabla g_j(\vec{x}^*) : 1 \leq i \leq m, j \in J^*\vec{x}^*)\}$ is linearly independent. Then there exists $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^p$ such that*

1. *$\mu \geq 0$,*
2. *$Df(\vec{x}^*) + \lambda^T Dh(\vec{x}^*) + \mu^T Dg(\vec{x}^*) = 0$, and*
3. *$\mu_j g_j(x) = 0$ for all $1 \leq j \leq p$.*

Remark 3. *Regarding the KKT conditions:*

- *If we wanted to replace the minimization of f with maximization, the result would read the same except we would require $\mu \leq 0$ instead.*

- *If $j \notin J(\vec{x}^*)$, the only way for the third condition to hold is if $\mu_j = 0$. Similarly, if $\mu_j > 0$, then for the third condition to hold, we need $j \in J(\vec{x}^*)$.*

Let's expand a bit on the role in Theorem 2.2.2 of the multipliers μ. First, the above remark should not lead you to assume that if $\mu_j = 0$ then g_j is inactive; all we are saying is that the only μ_j which might be nonzero are those for which $j \in J(\vec{x}^*)$. Second, if g_j is inactive at the minimizer, informally, it's not playing (locally) a role in the constrained optimization. That is, we could modify $\vec{x}^*$ very slightly and still satisfy that particular constraint.

So at a minimizer, the KKT conditions say we only need to focus on the active constraints h_i and g_j for $j \in J(vx^*)$. The assumption that $\{\nabla h_i(\vec{x}^*), \nabla g_j(\vec{x}^*) : 1 \leq i \leq m, j \in J * \vec{x}^*)\}$ is linearly independent is essentially the full-rank condition we required in Lagrange's theorem; this time we are requiring it on the set of active constraints. Moreover, the second KKT condition contains the same type of statement that we made before: that $\nabla f(\vec{x}^*)$ is a linear combination of the gradients of the active inequality constraint functions and the equality constraint functions.

Proof of the KKT theorem. Given the feasible set C, consider the set

$$\tilde{C} = \{\vec{x} \in C : g_j(\vec{x}) = 0, j \in J(\vec{x}^*); g_k(\vec{x}) < 0, k \notin J(\vec{x}^*)\}$$

which is the feasible set in which the active constraints are considered equality constraints. Clearly, $\vec{x}^* \in \tilde{C}$. We can pick an $\varepsilon > 0$ such that $g_k(\vec{x}) < 0$ for $\vec{x} \in B_\varepsilon(\vec{x}^*)$ for any $k \notin J(\vec{x}^*)$ and $f(\vec{x}^*)$ is minimal on $\tilde{U} \cap B_\varepsilon(\vec{x}^*)$. It should be clear that $\vec{x}^*$ is a local minimizer on $\tilde{U} \cap B_\varepsilon(\vec{x}^*)$.

By the Lagrange multiplier theorem, we can immediately conclude that there exists $(\lambda, \mu) \in \mathbb{R}^m \times \mathbb{R}^p$ such that

$$Df(\vec{x}^*) + \lambda^T Dh(\vec{x}^*) + \mu^T Dg(\vec{x}^*) = 0.$$

where we set get $\mu_k = 0$ for $k \notin J(\vec{x}^*)$, meaning we have found multipliers satisfying the second and third conditions.

So the last item is to establish that $\mu_j \geq 0$ for $j \in J(\vec{x}^*)$. We will prove the first KKT condition via contradiction: assume that there is some j such that $\mu_j < 0$. Let

$$\tilde{S}_j = \{\vec{x} : h(\vec{x}) = 0, g_k(\vec{x}) = 0, k \in J(\vec{x}^*), k \neq j\} \cap C$$

be the surface where the active constraints for $\vec{x}^*$ are still active except for the j constraint.

First, we establish the following claim: *There exists* $v \in T_{\vec{x}^*}\tilde{S}_j$ *such that* $Dg_j(\vec{x}^*)v \neq 0$. Recall that

$$T_{\tilde{S}_j}(\vec{x}^*) = \{v : Dh(\vec{x}^*)v = 0, Dg_k(\vec{x}^*)v = 0, k \in J(\vec{x}^*), k \neq j\}$$

To prove the claim, suppose that $Dg_j(\vec{x}^*)v = 0$ for all $v \in T_{\tilde{S}_j}(\vec{x}^*)$. Then $\nabla g_j(\vec{x}^*) \in (T_{\tilde{S}_j}(\vec{x}^*))^\perp$. But this in turn means that $\nabla g_j(\vec{x}^*)$ is in the span of ∇h_i and $\nabla g_k(\vec{x}^*)$ for $k \in J(\vec{x}^*)$ and $k \neq j$. That contradicts the linear independence assumption, meaning that the claim is true. Without loss of generality, we take v such that $Dg_j(\vec{x}^*)v < 0$. Note that this v is a direction along which g_j decreases, and so moving along v from $\vec{x}^*$ keeps us locally in C.

To obtain the contradiction, we multiply the second KKT condition by v:

$$Df(\vec{x}^*)v + \lambda^T Dh(\vec{x}^*)v + \mu^T Dg(\vec{x}^*)v = 0.$$

Since $v \in T_{\tilde{S}_j}(\vec{x}^*)$, this equation becomes

$$Df(\vec{x}^*)v = -\mu_j Dg_j(\vec{x}^*)v < 0$$

since $\mu_j < 0$. But this inequality says that there is a direction along which f is decreasing and which keeps us locally feasible. Thus $\vec{x}^*$ is not an optimum, and we obtain the desired contradiction. □

Example 2.2.3. *Consider the problem of optimizing* $f(x) = (x_1+2)^2 + (x_2 + 2.5)^2$ *subject to* $x_1^2 + 4x_2^2 \leq 16$ *and* $x_1 + x_2 \leq 2$. *The objective is a parabolic bowl, shifted, and the constraint set is the intersection of a half-plane and an ellipse; these are shown in Figure 2.2. If we ignore the feasibility of the points and don't worry about the sign of* μ_j, *we want to solve*

$$\begin{aligned} 2(x_1 + 2) + 2x_1\mu_1 + \mu_2 &= 0 \\ 2(x_2 + 2.5) + 8x_2\mu_1 + \mu_2 &= 0 \\ \mu_1(x_1^2 + 4x_2^2 - 16) &= 0 \\ \mu_2(x_1 + x_2 - 2) &= 0. \end{aligned}$$

Which gives the points and mu:

$x = [0, 2]^T, \mu = [-0.31, -4]^T$ $\qquad$ $x = [3.2, -1.2]^T, \mu = [-0.49, -7.28]^T$

$x = [-1.81, -1.78]^T, \mu = [0.1, 0]^T$ $\qquad$ $x = [3.88, 0.49]^T, \mu = [-1.52, 0]^T$

$x = [-2, -2.5]^T, \mu = [0, 0]^T$ $\qquad$ $x = [1.25, 0.75]^T, \mu = [0, -6.5]^T$

The points are drawn in Figure 2.2. If I just look at the points and the constraint set (as well as the level sets of f*), can we predict the signs of* μ_1, μ_2 *or if they are equal to* 0.

- *At* $[0, 2]^T$ *and* $[3.2, -1.2]^T$, *both constrains are active. Since* ∇f *is not parallel to* ∇g_1 *or* ∇g_2, *we know that the* μ_i *cannot be zero. Since* μ *is negative at these points, we know that the points might be maximizers of* f *but are not minimizers.*

- *At* $[-1.81, -1.78]^T$*, we know that the linear inequality is inactive, so* $\mu_2 = 0$*. Since moving off the edge of the ellipse increases the value of* f*, we expect* $\mu_1 > 0$*. So this is a viable candidate for a minimizer.*

- *At* $[3.88, 0.49]^T$*, again the elliptical constraint is active, but moving into the ellipse would decrease* f*, we expect* μ *to be negative. If it were feasible, this would be a candidate for a minimizer.*

- *At* $[-2, -2.5]^T$*,* f *has a critical point, so the second KKT condition means* $\mu = 0$*; however it is not feasible.*

- *At* $[1.25, 0.75]^T$*, the linear constraint is active and the elliptical constraint is inactive. The sign of* μ *suggests this is possibly a maximizer of* f*.*

Example 2.2.4. *Suppose we want to minimize* $f(x) = (x_1-1)^2+x_2-2$ *subject to* $x_2 - x_1 - 1 = 0$ *and* $x_1 + x_2 - 2 \leq 0$*. Note that* $Dh(x) = [-1, 1]$ *and* $Dg(x) = [1, 1]$ *are always linearly independent. The KKT conditions and feasibility conditions require*

$$\begin{aligned} 2x_1 - 2 - \lambda + \mu &= 0 \\ 1 + \lambda + \mu &= 0 \\ \mu(x_1 + x_2 - 2) &= 0 \\ \mu &\geq 0 \\ x_2 - x_1 - 1 &= 0 \\ x_1 + x_2 - 2 &\leq 0. \end{aligned}$$

Let's for the moment ignore the inequality constraint (we will check if out any candidates for minimizers satisfy the inequality). Then either$\mu = 0$ *or* $g(x) = 0$*. If* $g(x) = 0$*, then we can solve the remaining*

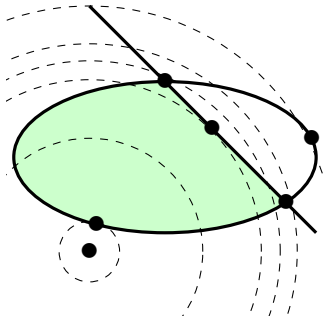

Figure 2.2: The constraint set, level curves (dashed) of f and candidate points discussed in Example 2.2.3

equalities to get $\mu = 0$. *This in turn means*

$$\begin{aligned} 2x_1 - 2 - \lambda &= 0 \\ 1 + \lambda &= 0 \\ x_2 - x_1 - 1 &= 0. \end{aligned}$$

That gives us $x = [.5, 1.5]^T$ *and* $\lambda = 1$. *This point is feasible (Check!). Note that if we assume* $\mu = 0$, *we get* $g(x) = 0$ *by the same logic and the resulting minimizer is unchanged. Since the constraint set is unbounded, we substitute* $x_2 = x_1 + 1$ *from the constraint into the objective to get* $(x_1 - 1)^2 + (x_1 - 1) = x_1(x_1 - 1)$ *which clearly goes to* ∞ *as* $x_1 \to \infty$. *So the point from the KKT theory is a minimizer.*

2.3 Exercises

1. Prove that if A is a matrix, then $\mathcal{R}(A^T) = \mathcal{N}(A)^{\perp}$.

2. Minimize $f(x) = (x_1 - x_2)^3 + (x_1 + x_2)^3$ subject to the following constraints (as separate problems with $m = 1$):

   - $2x_1^2 + x_2^2 = 1$,

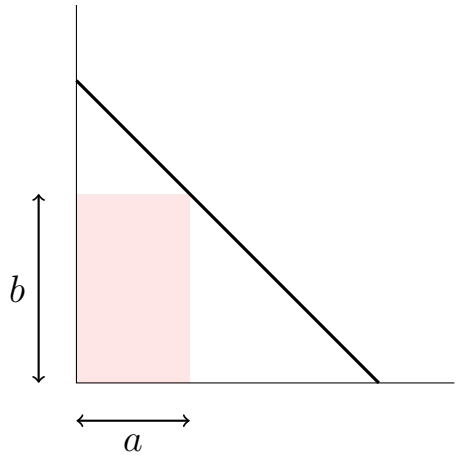

Figure 2.3: The shortest ladder problem

- $x_1^2 + 2x_2^2 = 1$.

Finally, minimize $f(x) = (x_1 - x_2)^3 + (x_1 + x_2)^3 + x_3^2$ subject to $2x_1^2 + x_2^2 + x_3^2 = 1$.

3. Prove that the tangent space to the unit sphere (in $\mathbb{R}^3$) at $p = [0, 0, 1]$ is the set $\{[a, b, 0]^T : a, b \in \mathbb{R}\}$.

4. What is the shortest ladder which can be leaned against a wall that can have a box which is a meters wide and b meters tall fit under the ladder? A schematic is shown in item 4

5. Where is the surface $x_1x_2 + x_2x_3 + x_1x_3 = 12$ closest to the origin? Approximately how much closer to/further from the origin is $x_1x_2 + x_2x_3 + x_1x_3 = 11.9$?

6. Minimize $f(x) = x_1+x_2-x_3$ over points in the plane $x_1+x_2+x_3 = 2$ and on the unit sphere.

7. Find $a, b \in \mathbb{R}$ which minimizes $f(a, b) = \int_a^b \frac{1}{1+t^4} dt$ subject to $a^2b^2 = 1$. Note that since the feasible set is unbounded, you also need to check the behavior as $\|(a, b)^T\| \to \infty$.

8. Minimize $f(x) = x_1 + x_2$ over points on the unit circle by eliminating x_2. How does choosing the sign for the square root affect the answer; how does the "wrong" choice affect the answer?

9. Consider the problem of finding the point on the parabola $x_2 = \frac{1}{5}(x_1 - 1)^2$ closest (in the Euclidean norm sense) to $x = (1, 2)^T$.

   a) Find all candidates for optimizers and verify whether the rank condition holds there.

   b) Which candidates are solutions?

   c) Substitute the constraint into the objective and eliminate the variable x_1, giving an unconstrained problem. Show the solutions of this problem are not solutions of the original.

10. Minimize $f(x) = (x_1 - 10)^2 + (x_2 - 10)^2 + (x_3 - 10)^2$ subject to $x^T x \leq 12$ and $-x_1 - x_2 + 2x_3 \leq 0$. If the a point is optimal which constraints are active at x^*? Is x^* a global minimizer? Is the global minimizer unique?

11. Maximize $f(x) = x_1^2 - x_2^2$ subject to $(x_1 - 2)^2 + x_2^2 - 4 \leq 0$.

12. Solve the stock picking problem from section 1.1.

13. Solve the utility function problem from section 1.1.

Chapter 3

Numerical Optimization

Numerical Optimization, is concerned with finding approximate solutions to optimization problems where often some of the following hold true:

1. the function f is relatively expensive to evaluate at any point;
2. the gradient ∇f is unavailable in closed form;
3. solving explicitly the necessary conditions for optimality (such as the KKT conditions or finding multipliers) is impractical or impossible.

In the face of practical difficulties like these, numerical algorithms have been developed to approximately solve optimization problems. We will focus on problems *without* constraints:

$$\min_{x \in \mathbb{R}^n} f(x) \tag{3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a (sufficiently) smooth function.

Optimization algorithms can usually broken into two main components in each iteration: computing update directions and performing a line search along this update direction. We will focus on the latter first before looking at various methods of computing update directions.

3.1 Line searches

A line search method is a means of deciding the size of a step in the direction of some search direction. Concretely, we are looking for an $\alpha_k > 0$ which is called the *step length* which gives an effective update rule

$$x_{k+1} = x_k + \alpha_k p_k$$

where the search direction p_k is given.

Definition 3.1.1. *A* descent direction *for f at $\vec{x}$ is a p such that* $\nabla f(\vec{x})^T p < 0$.

Typically we expect at each k that p_k is a descent direction. As we will see in the next section(s), often $p_k = -B_k^{-1}\nabla f_k$ where B_k is some symmetric and nonsingular matrix.

In the remainder of this section we turn to selecting $\alpha_k > 0$. This task is complicated by two competing objectives: we want to give a large reduction of f at x_{k+1} but we don't want to spend a large effort. Ideally, we would want to minimize the one dimensional function

$$\psi(\alpha) = f(x_k + \alpha p_k)$$

and then pick α_k as the minimizer of ψ. But this usually is too difficult a problem (at each iteration!) to properly solve while maintaining efficiency of the overall optimization method. Instead, we will look at methods which try out candidates for α_k until some conditions are satisfied.

Armijo and Wolfe Conditions

Let's look at what such conditions might be. Recalling that $f_{k+1} = f(x_k + \alpha_k p_k)$, a first guess might be is that we just want our α_k to satisfy

$$f_{k+1} < f_k.$$

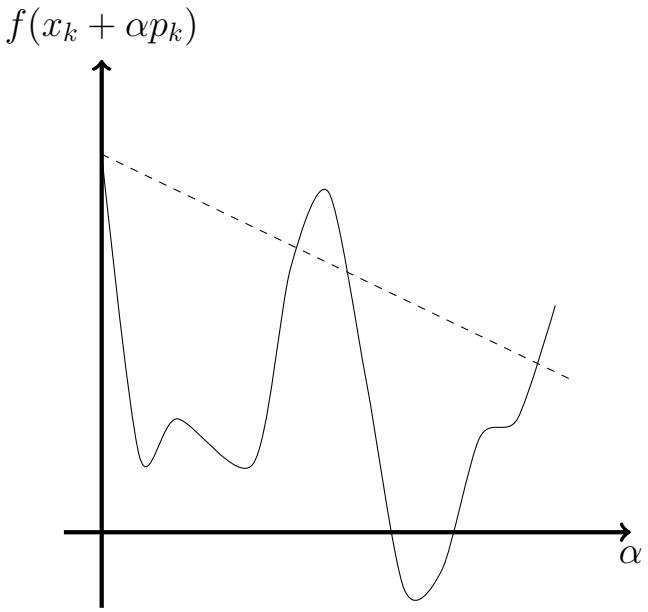

Figure 3.1: Comparison between $f(x_k + \alpha p_k)$ and the dashed line $f_k + c\alpha \nabla f_k^T p_k$

But this condition is not sufficient to prevent bad behavior. Consider minimizing $f(x) = (x-1)^2 - 1$. If we set $x_1 = 1 + \sqrt{2}$, $p_k = -\nabla f_k$, and α_k is chosen such that $x_k = 1 + (-1^{k+1})\sqrt{\frac{k+1}{x_k}}$ then the updates will suffer one major issue and are also relatively inefficient, even though $f(k+1) < f_k$ at each iteration.

A popular modification to this required condition on α is the *Armijo condition*: for some $c \in (0, 1)$, we require α_k to be such that

$$f(x_k + \alpha_k p_k) = f_{k+1} \leq f_k + c\alpha_k \nabla f_k^T p_k. \tag{3.2}$$

What does this condition require? It means α_k should reduce f by an amount proportional to the directional derivative $\nabla f_k^T p_k$. If we look at section 3.1, the α_k we pick should be at a point where the graph of the linear function (with respect to α) $f_k + c\alpha \nabla f_k^T p_k$ (the right hand side of the Armijo condition) lies above the graph of $f(x_k + \alpha p_k)$. While

we are free to pick c in the Armijo condition, we typically pick it to be fairly small; choosing c as something like 10^{-4} typically works quite well.

The Armijo condition doesn't assures us all of the behavior we might want in an line search. In section 3.1, the Armijo condition is satisfied by very small α. In other words, the Armijo condition doesn't assure that we don't simply take (needlessly) small steps; such steps would lead to a very inefficient algorithm. To avoid needlessly small steps a typical second requirement on α_k is the *curvature condition* which requires for some $\tilde{c} \in (c, 1)$ that

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq \tilde{c} \nabla f_k^T p_k. \tag{3.3}$$

Note that the left hand side of the inequality is $\psi'(\alpha_k)$. Since we are assuming that p_k is a descent direction, $\psi'(0)$ is negative: if $\psi'(\alpha)$ is only a bit negative or positive, we might expect that pushing further in the direction p_k will not give further decreases in f. In that case, we will terminate the line search at such an α; otherwise, the curvature condition suggests we still have more improvement in f possible in the direction p_k. Graphically, we can readily see this by looking at the previous figure: the curvature condition ensures we don't pick too small an α.

The Armijo and curvature conditions, in conjunction, are typically called the (weak) *Wolfe conditions*; modifying the curvature condition gives the *strong Wolfe conditions*:

$$f(x_k + \alpha_k p_k) \leq f_k + c\alpha_k \nabla f_k^T p_k \tag{3.4}$$

$$|\nabla f(x_k + \alpha_k p_k^T) p_k| \leq \tilde{c} |\nabla^T f_k p_k|. \tag{3.5}$$

This modified curvature condition ensures that $\psi'(\alpha_k)$ is not too positive, meaning the α_k we pick is not too far from a point where ψ' is close to 0; this is typically the best we can expect in our goal in finding an α close to the minimizer of $\psi(\alpha)$. The following result tells us that

the (strong and weak) Wolfe conditions are not too strict and thus we can expect to find a good step length in our line search.

Lemma 3.1.1. *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, p_k is a descent direction at x_k and that f is bounded below along the direction p_k for any step size. Then if $0 < c < \tilde{c} < 1$, there exists an interval of step length satisfying the weak and strong Wolfe conditions.*

Proof. Define $\psi(\alpha) = f(x_k + \alpha p_k)$ and the linear function $\ell(\alpha) = f_k + \alpha c \nabla f_k^T p_k$. Then since $\phi(\alpha)$ is bounded below for any α and $c \in (0, 1)$, the graph of ψ intersects the graph of ℓ at least once. Let $\bar{\alpha} > 0$ be the smallest α where the intersection occurs:

$$f(x_k + \bar{\alpha} p_k) = f_k + \bar{\alpha} c \nabla f_k^T p_k.$$

Then for any step size less than $\bar{\alpha}$, the Armijo condition will hold.

By the mean value theorem, there exists an $\tilde{\alpha} \in (0, \bar{\alpha})$ such that

$$\psi(\bar{\alpha}) - f_k = \bar{\alpha} \nabla f(x_k + \tilde{\alpha} p_k)^T p_k.$$

Combining the previous two equalities, we get

$$\nabla f(x_k + \tilde{\alpha} p_k)^T p_k = c \nabla f_k^T p_k > \tilde{c} \nabla f_k^T p_k$$

since $c < \tilde{c}$ and $\nabla f_k^T p_k < 0$. So $\tilde{\alpha}$ the second weak Wolfe condition; since f is continuously differentiable, there must be an interval around $\tilde{\alpha}$ where it also holds. We can conclude the strong Wolfe conditions hold in the same interval. □

In practice, these conditions are the standard ones we use when deciding if a given step length in our line search is good enough to terminate the search. Others have been found which also give good behavior, such as the Goldstein conditions. However, the (strong) Wolfe conditions will be enough for our purposes.

Backtracking line search

Now that we know how to determine whether a given step size is a good one to use for an update, we turn to actually finding such step sizes. In the situation where we know little about the function, a standard and simple line search algorithm is the backtracking or Armijo algorithm. First we pick a tolerance $\varepsilon > 0$ to correspond to how "close" to being a minimizer of ψ we will allow α_k. Then we need to pick an initial $\alpha_i > 0$, and a "search ratio" $r \in (0, 1)$. The algorithm is given Algorithm 1. One might object that we aren't satisfying the curvature condition,

Data: ψ, $r \in (0, 1)$, $\alpha > 0$, ∇f_k, $c \in (0, 1)$, p_k,f_k
if α *satisfies Equation* (3.2) **then**
  Set $\alpha_k \leftarrow \alpha$;
  STOP;
end
Set $\alpha \leftarrow r\alpha$;
Repeat;

Algorithm 1: The Armijo line search

which is true. However, we are looking for the *first* α_k which gives a sufficient decrease, working our way from the largest allowed step size (which, granted, we must predfine). This means that, while we might not find the step size giving the largest possible difference between the left and right hand sides of Equation (3.5) (which we might consider a part of the goal in finding the minimizer of ψ,), we are in spirit avoiding the concern that led us to look at the curvature condition. Additionally, the method is easy to implement and does not require evaluating $\nabla f(x_k + \alpha p_k)$ for each checked step size α. This is useful when the derivative of f is expensive to evaluate.

Newton line search

The next method, the *Newton's line search* is more involved than the backtracking line search, but may involve looking at fewer candidates for α_k. We define q to be the second-order Taylor series expansion of ψ around a given α_i

$$q_i(\alpha) = \psi(\alpha_i) + \psi'(\alpha_i)(\alpha - \alpha_i) + \frac{1}{2}\psi''(\alpha_i)(\alpha - \alpha_i)^2.$$

Then the line search is performed by using Newton's method; that is we iteratively set α_{i+1} to be the minimizer of q_i:

$$\alpha_{i+1} = \alpha_i - \frac{\psi'(\alpha_i)}{\psi''(\alpha_i)}.$$

This is repeated until α_{i+1} and α_i are close enough together (say $|\alpha_{i+1} - \alpha_i| < 10^{-8}$). The final α_{i+1} is then used for our α_k. While this appears more "clever" than the Armijo line search, we have to consider two major caveats. First, the Newton line search might not work very well if our first α_i is not near a minimizer of ψ. Second, the line search requires (repeated) evaluation of ψ'', which may prove computationally expensive. Contrast that with with the Armijo which involves in a single line search only one evaluation of ∇f_k and evaluations of ψ only (i.e. no derivatives).

Other line search methods

These are far from the only line search methods that have been developed. For instance, there are the golden section and fibonacci methods, where we repeatedly winnow down the interval where the minimizer of ψ might be.

3.2 Gradient-Based Methods

Now that we know how to compute an update via line search once an update direction has been chosen, we now turn to the task of actually choosing said direction. First, we recall that the gradient $\nabla f(\vec{x})$ points in the direction in which f increases the most rapidly from $\vec{x}$.

Steepest Descent Method

Using this property of the gradient, the *steepest descent* algorithm is conceptually very straightforward: at each iteration, move along the direction which locally gives the most improvement in our goal to minimize f: A (usually unfortunate) property of the steepest descent

Data: $\vec{x}_k$,$\varepsilon > 0$,$f : \mathbb{R}^n \to \mathbb{R}$
Set $p_k = -\nabla f(\vec{x}_k)$;
if $\|p_k\| \leq \varepsilon$ **then**
  $\vec{x}_k \approx \vec{x}^*$;
  STOP;
end
Set $\psi(\alpha) = f(\vec{x}_k + \alpha p_k)$;
Perform a line search using ψ, resulting in α_k;
Set $\vec{x}_{k+1} \leftarrow \vec{x}_k + \alpha_k p_k$;
Repeat;

Algorithm 2: The Steepest Descent Method

method is that the iterations follow a pattern:

Proposition 3.2.1. *Suppose $\{\vec{x}_k\}$ are obtained via the steepest descent method with an exact line search; that is, α_k is the exact minimizer of ψ at each iteration. Then $\vec{x}_{k+1} - \vec{x}_k$ is orthogonal to $\vec{x}_{k+2} - \vec{x}_{k+1}$.*

Proof. By definition,

$$\langle \vec{x}_{k+1} - \vec{x}_k, \vec{x}_{k+2} - \vec{x}_{k+1} \rangle = \alpha_k \alpha_{k+1} \langle \nabla f(\vec{x}_k), \nabla f(\vec{x}_{k+1}) \rangle.$$

Next, since we have assumed that the line search is perfect,

$$0 = \psi'(\alpha_k) = \nabla f(\vec{x}_k - \alpha_k \nabla f_k)^T (-\nabla f_k) = -\langle \nabla f(\vec{x}_{k+1}), \nabla f(\vec{x}_k) \rangle.$$

□

This shows that the steepest descent method results in a very inefficient zig-zag behavior.

Newton and Quasi-Newton Methods

Recall that Newton's method in 1D is an iterative procedure where at each iterate, a second-order Taylor approximation to f is minimized and the new point is chosen as this minimizer. If our starting point is close enough to the minimizer, this has a few desirable properties: namely the descent direction is chosen and then the step-size is determined immediately without the need for a line search. In higher dimensions, Newtons method proceeds similarly as in the 1D case. Given a point $\vec{x}_k \in \mathbb{R}^n$ and a function $f : \mathbb{R}^n \to \mathbb{R}$, we define the quadratic approximation of f at this point:

$$q_k(\vec{x}) := f_k + (\vec{x} - \vec{x}_k)^T \nabla f_k + \frac{1}{2} (\vec{x} - \vec{x}_k)^T D^2 f_k (\vec{x} - \vec{x}_k).$$

Next, we note that if q has a minimizer $\vec{z}$, then

$$0 = \nabla q(\vec{z}) = \nabla f_k + D^2 f_k (\vec{z} - \vec{x}_k)$$

so if $D^2 f_k$ is positive definite we set $\vec{x}_{k+1}$ to be the minimizer of q:

$$\vec{x}_{k+1} = \vec{x}_k - (D^2 f_k)^{-1} \nabla f_k.$$

For high-dimensional problems, we wouldn't compute and store the inverse of the Hessian $(D^2 f_k)^{-1}$, rather we would use a method such as Gaussian elimination to solve $(D^2 f_k)p_k = -\nabla f_k$. Secondly, note that just as in 1D, we are not performing a line search. Rather, we are just using $\alpha_k = 1$.

There are several issues with Newton's method. First, we need the Hessian to be positive definite at each iteration. However, this is not a problem when we are close to a minimum, where this will hold in general for smooth f. More significantly, we are not assured that $f_{k+1} < f_k$ even if $(D^2 f_k)$ is positive definite. If we step too far in p_k, while p_k is a descent direction, we might not have a decreasing sequence $\{f_k\}$. This is not a problem if we are willing to modify Newton's method, as shown by the following theorem.

Theorem 3.2.1. *Let $\{x_k\}$ be a sequence generated by Newton's method. If $D^2 f_k > 0$ and $\nabla f_k \neq 0$, then*

$$p_k = -(D^2 f_k)^{-1} \nabla f_k = \vec{x}_{k+1} - \vec{x}_k$$

is a descent direction. That is, there exists a $\hat{\alpha} > 0$ such that for all $\alpha \in (0, \hat{\alpha})$,

$$f(\vec{x}_k + \alpha p_k) < f_k.$$

Proof. Let $\psi(\alpha) = f(\vec{x}_k + \alpha p_k)$ as before. Then by the chain rule

$$\psi'(0) = -\nabla f_k^T (D^2 f_k)^{-1} \nabla f_k.$$

Since the Hessian is positive definite, its eigenvalues $\lambda_1, \ldots, \lambda_n$ are all positive. This means that the eigenvalues $1/\lambda_\ell$ of $(D^2 f_k)^{-1}$ are all positive and so the inverse of the Hessian is positive definite also. That means $\psi'(0) < 0$ and so there is a $\hat{\alpha} > 0$ for the one-dimensional function ψ such that $\psi(\alpha) < \psi(0)$ for $\alpha \in (0, \hat{\alpha})$. □

Data: $\vec{x}_k$,$\varepsilon > 0$,$f : \mathbb{R}^n \to \mathbb{R}$
if $\|\nabla f(\vec{x}_k)\| \leq \varepsilon$ **then**
  $\vec{x}_k \approx \vec{x}^*$;
  STOP;
end
Solve $(D^2 f_k)p_k = -\nabla f(\vec{x}_k)$;
Set $\psi(\alpha) = f(\vec{x}_k + \alpha p_k)$;
Perform a line search using ψ, resulting in α_k;
Set $\vec{x}_{k+1} \leftarrow \vec{x}_k + \alpha_k p_k$;
Repeat;

Algorithm 3: The modified Newton's Method

This means that, as long as we are in a region where $D^2 f_k > 0$, we could modify Newton's method to ensure a descent sequence; this is summarized in Algorithm 3. While we have now saddled our algorithm with the added expense of performing a line search, near minima the modified Newton's method has better convergence than the steepest descent method.

One obvious drawback to the modified Newton's method which we haven't addressed is needing to compute the Hessian $D^2 f_k$; additionally, for large n computing p_k can be computationally expensive also. Quasi-Newton methods overcome this drawback by noting that we don't need the Hessian, but rather a good approximation H_k of the inverse Hessian $H_k \approx (D^2 f_k)^{-1}$. Their general form is shown in Algorithm 4.

How we choose H_k depends on the method we are using. We want to design for such functions an H_k such that $x_{k+1} - x_k$ is a descent direction at each k. To do this we need (to recall) the following definition.

Definition 3.2.2. *A function $f(\vec{x}) = o(\|\vec{x}\|^r)$ (read as f is "little-o" of $\|\vec{x}\|^r$) if $\lim_{\vec{x}\to 0} \frac{\|f(\vec{x})\|}{\|\vec{x}\|^r} = 0$.*

Data: $\vec{x}_k$,$\varepsilon > 0$,$f : \mathbb{R}^n \to \mathbb{R}$
if $\|\nabla f(\vec{x}_k)\| \leq \varepsilon$ **then**
  $\vec{x}_k \approx \vec{x}^*$;
  STOP;
end
Compute H_k;
Solve $p_k = -H_k \nabla f(\vec{x}_k)$;
Set $\psi(\alpha) = f(\vec{x}_k + \alpha p_k)$;
Perform a line search using ψ, resulting in α_k;
Set $\vec{x}_{k+1} \leftarrow \vec{x}_k + \alpha_k p_k$;
Repeat;

Algorithm 4: The general form of a Quasi-Newton Method

Taylor's theorem says that, since $\vec{x}_{k+1} - \vec{x}_k = -\alpha_k H_k \nabla f_k$, we have

$$\begin{aligned} f_{k+1} &= f_k + \nabla f_k^T (\vec{x}_{k+1} - \vec{x}_k) + o(\|\vec{x}_{k+1} - \vec{x}_k\|) \\ &= f_k - \alpha_k \nabla f_k^T H_k \nabla f_k + o(\|H_k \nabla f_k\| \alpha_k). \end{aligned}$$

If $\nabla f_k^T H_k \nabla f_k > 0$ holds (which is true if H_k is positive definite), then we have $f_{k+1} < f_k$ as long as the error term is smaller than $\alpha_k \nabla f_k^T H_k \nabla f_k$.

These methods start by looking at what happens if $D^2 f_k = Q$ for each Q; that is, when f is a quadratic function $f(\vec{x}) = \frac{1}{2}\vec{x}^T Q \vec{x} - b^T \vec{x}$ where Q is symmetric. In this case, since $\nabla f(\vec{x}) = Q\vec{x} - b$, we have

$$\nabla f_{k+1} - \nabla f_k = Q(\vec{x}_{k+1} - \vec{x}_k).$$

Define, for notational simplicity, $\Delta g_k := \nabla f_{k+1} - \nabla f_k$ and $\Delta \vec{x}_k := \vec{x}_{k+1} - \vec{x}_k$. Then for any k,

$$Q^{-1} \Delta g_i = \Delta \vec{x}_i, \qquad \text{for all } 0 \leq i \leq k. \tag{3.6}$$

So we expect H_{k+1} to satisfy the same:

$$H_{k+1} \nabla g_i = \Delta \vec{x}_i \qquad \text{for all } 0 \leq i \leq k.$$

In other words, we expect that at the n-th iteration,

$$H_n\big[\Delta g_0, \ldots, \Delta g_{n-1}\big] = \big[\Delta \vec{x}_0, \ldots \Delta \vec{x}_{n-1}\big].$$

So as long as the matrix $G := \big[\Delta g_0, \ldots, \Delta g_{n-1}\big]$ is nonsingular, we can write this condition as $H_n = XG^{-1} = Q^{-1}$. Note that Newton's method minimizes quadratic functions in one step; the above shows that applying a quasi-Newton algorithm at the $(n+1)$-th step on a quadratic function is the same as implementing a single step of Newton's method. So, after at most $n+1$ steps, any properly designed quasi-Newton method with exact line search will minimize a quadratic function. We will hold off on establishing the condition that G is nonsingular for quadratic f until the next section where we discuss conjugate gradient methods.

Since, Taylor's theorem says any f is locally approximated by a quadratic function, the above conditions motivate the design of $H_k \approx (D^2 f_k)^{-1}$ for general f. Typically, we formulate such designs as H_{k+1} comes from adding a correction term to H_k; since any matrix can be described in this form, this is not a restriction but rather a convention.

The first quasi-Newton algorithm we consider is the DFP algorithm (named for its originator Davidon and Fletcher and Powell who implemented and popularized it). This involves generating a rank-two correction term to H_k at each step to generate H_{k+1} and is summarized in Algorithm 5. Note that we only need some positive definite symmetric H_0 to initialize the algorithm; typically we set $H_0 = i$ for convenience.

Theorem 3.2.3. *Suppose $\nabla f_k \neq 0$ and H_k is positive definite. Then the H_{k+1} coming from the DFP algorithm is also positive definite.*

We will not prove this result in this text.

The DFP algorithm can occasionally get stuck if H_k becomes singular, which means $(D^2 f)^{-1} \approx 0$. In that situation it might be more useful to look directly at the Hessian. The BFGS (named for its creators

Data: $\vec{x}_0$,$\varepsilon > 0$,$f : \mathbb{R}^n \to \mathbb{R}$,symmetric positive definite H_0
if $\|\nabla f(\vec{x}_k)\| \leq \varepsilon$ **then**
  $\vec{x}_k \approx \vec{x}^*$;
  STOP;
end
Set $p_k = -H_k \nabla f(\vec{x}_k)$;
Set $\psi(\alpha) = f(\vec{x}_k + \alpha p_k)$;
Perform a line search using ψ, resulting in α_k;
Set $\vec{x}_{k+1} \leftarrow \vec{x}_k + \alpha_k p_k$;
Set $\Delta\vec{x}_k = \vec{x}_{k+1} - \vec{x}_k$ and $\Delta g_k = \nabla f_{k+1} - \nabla f_k$ and

$$H_{k+1} = H_k + \frac{\Delta\vec{x}_k \Delta\vec{x}_k^T}{\Delta\vec{x}_k^T \Delta g_k} - \frac{(H_k \Delta g_k)(H_k \Delta g_k)^T}{\Delta g_k^T H_k \Delta g_k};$$

Repeat with $k \leftarrow k + 1$;

Algorithm 5: The DFP algorithm

Broyden, Fletcher, Goldfarb, and Shanno) algorithm has become the most popular quasi-Newton method. Recall we wanted for quadratic functions

$$H_{k+1}\Delta g_i = \Delta\vec{x}_i \qquad \text{for all } 0 \leq i \leq k$$

which came from wanting $\Delta g_i = Q\Delta\vec{x}_i$. The BFGS algorithm looks to approximate Q rather Q^{-1} by a sequence of matrices B_k while requiring

$$\Delta g_i = B_{k+1}\Delta\vec{x}_i \qquad \text{for all } 0 \leq i \leq k.$$

How do we create B_k? If we interchange B_k and H_k as well as interchanging Δg_k and $\vec{x}_k$ from the DFP , we get a scheme to update B_k:

$$B_{k+1} = B_k + \frac{\Delta g_k \Delta g_k^T}{\Delta g_k^T \Delta\vec{x}_k} - \frac{(B_k \Delta\vec{x}_k)(B_k \Delta\vec{x}_k)^T}{\Delta\vec{x}_k^T B_k \vec{x}_k};$$

But how does this give us H_{k+1}? It turns out *rank-one perturbations* of a matrix can be inverted analytically:

$$(A + \vec{u}\vec{v}^T)^{-1} = A^{-1} - \frac{(A^{-1}\vec{u})(\vec{v}^T A^{-1})}{1 + \vec{v}^T A^{-1}\vec{u}}.$$

Applying this identity twice, we can write B_{k+1}^{-1} in terms of $B_k^{-1} = H_k$. This gives the BFGS algorithm, summarized in Algorithm 7, which proves in most applications to be robust when the line search is rather inexact (meaning the we can set the line search tolerance to being rather not too strict and thus making it less costly).

Data: $\vec{x}_0$,$\varepsilon > 0$,$f : \mathbb{R}^n \to \mathbb{R}$,symmetric positive definite H_0
if $\|\nabla f(\vec{x}_k)\| \leq \varepsilon$ **then**
  $\vec{x}_k \approx \vec{x}^*$;
  STOP;
end
Set $p_k = -H_k \nabla f(\vec{x}_k)$;
Set $\psi(\alpha) = f(\vec{x}_k + \alpha p_k)$;
Perform a line search using ψ, resulting in α_k;
Set $\vec{x}_{k+1} \leftarrow \vec{x}_k + \alpha_k p_k$;
Set $\Delta\vec{x}_k = \vec{x}_{k+1} - \vec{x}_k$ and $\Delta g_k = \nabla f_{k+1} - \nabla f_k$ and

$$H_{k+1} = H_k + \left(1 + \frac{\Delta g_k^T H_k \Delta g_k}{\Delta g^T \Delta\vec{x}_k}\right)\frac{\Delta\vec{x}_k \Delta\vec{x}_k^T}{\Delta\vec{x}_k^T \Delta g_k} - \frac{(H_k \Delta g_k \vec{x}_k^T) + (H_k \Delta g_k \Delta\vec{x}_k^T)^T}{\Delta g_k^T \vec{x}_k};$$

Repeat with $k \leftarrow k + 1$;

Algorithm 6: The BFGS algorithm

Limited-Memory methods The DFP and BFGS methods are appealing due to not needing to perform any matrix inversions. However, the H_k will be an $n \times n$ matrix which in general need to be stored and

updated at each iteration. This may become prohibitive to store for large n: since H_k is an approximation of the inverse of $D^2 f_k$, we have no expectation that after many iterations it won't have many nonzero entries which need to be stored.

In light of this concern, limited memory methods have been developed. While they are not as fast (with respect to iterations needed) as the above methods, they benefit from not needing to store H_k. The first insight needed to develop such a method is to note that we don't in fact need the entries of H_k. Instead, we need only to know the result of H_k being multiplied with $-\nabla f_k$. Next, we note that we could rewrite H_{k+1} as

$$H_{k+1} = V_k^T h_k V_k + \rho_k \Delta x_k \Delta x_k^T$$

where $\rho_k = (\Delta g_k^T \Delta x_k)^{-1}$ and $V_k = I - \rho_k \Delta g_k \Delta x_k$. But H_k used the same update formula, so we know, by recursively looking at the update that if we started with H_k^0 which we allow to vary with k (in BFGS we used the same H_0 for this matrix), then

$$\begin{aligned} H_k = &(V_{k-1}^T \dots V_{k-k}^T) H_k^0 (V_{k-k} \dots V_{k-1}) \\ &+ \rho_{k-k} (V_{k-1}^T \dots V_{k-k+1}^T) \Delta x_{k-k} \Delta x_{k-k}^T (V_{k-k+1} \dots V_{k-1}) \\ &+ \rho_{k-k+1} (V_{k-1}^T \dots V_{k-k+2}^T) \Delta x_{k-k+1} \Delta x_{k-k+1}^T (V_{k-k+2} \dots V_{k-1}) \\ &\vdots \\ &+ \rho_{k-1} \Delta x_{k-1} \Delta x_{k-1}^T. \end{aligned}$$

The L-BFGS method takes this and truncates the memory by setting some m and only using the most recent m terms in this expanded

update formula. So we instead use

$$\begin{aligned} H_k = &(V_{k-1}^T \dots V_{k-m}^T) H_k^0 (V_{k-m} \dots V_{k-1}) \\ &+ \rho_{k-m}(V_{k-1}^T \dots V_{k-m+1}^T)\Delta x_{k-m}\Delta x_{k-m}^T (V_{k-m+1} \dots V_{k-1}) \\ &+ \rho_{k-m+1}(V_{k-1}^T \dots V_{k-m+2}^T)\Delta x_{k-m+1}\Delta x_{k-m+1}^T (V_{k-m+2} \dots V_{k-1}) \\ &\vdots \\ &+ \rho_{k-1}\Delta x_{k-1}\Delta x_{k-1}^T. \end{aligned}$$

This expression in turn allows us to write a two-loop recursion formula for computing $H_k \nabla f_k$ without storing H_k.

$q \leftarrow \nabla f_k$;
for $i = k-1, k-2, \dots, k-m$ **do**
  $\alpha_i \leftarrow \rho_i \Delta x_i q_i$;
  $q \leftarrow q - \alpha_i \Delta g_i$;
end
$r \leftarrow H_k^0 q$;
for $i = k-m, k-m+1, \dots, k-1$ **do**
  $\beta \leftarrow \rho_i \Delta g_i r$;
  $r \leftarrow r + \Delta x_i(\alpha_i - \beta)$;
end
Set $H_k \nabla f_k = r$;

Algorithm 7: The L-BFGS two loop recursion

Notice that if H_k^0 is diagonal, then we need a total of $4mn + n$ multiplications (if not, the second term becomes larger). A good practical choice for this matrix is

$$\frac{\Delta x_{k-1}^T \Delta g_{k-1}}{\Delta g_{k-1}^T \Delta g_{k-1}} I$$

which tries to estimate a good scale for the current Hessian matrix along the most recent update direction. Typically, this also helps ensure that p_k is of the proper scale, which usually means we can take

$\alpha_k = 1$. The algorithm's performance then can be shown to depend on balancing the needs of picking m not too large to involve lots of storage with the needs of needing m to be big enough to maintain an accurate approximation of the inverse Hessian.

3.3 Conjugate Gradient Methods

A drawback to Newton's method is the need to compute Hessian matrices; conjugate gradient methods, like quasi-Newton methods (which can be written as conjugate gradient methods) do not need Hessian computations and can solve quadratic minimization problems in n steps.

As before, we first look at quadratic functions $f(\vec{x}) = \frac{1}{2}\vec{x}^T Q\vec{x} - \vec{x}^T b$.

Definition 3.3.1. *Let $Q \in \mathbb{R}^{n\times n}$ be symmetric. Two directions d_1, d_2 are* Q-conjugate *if $d_1^T Q d_2 = 0$. A set of vectors $\{d_j\}_{j=1}^m$ is Q-conjugate if for all $i \neq j$, $d_i^T Q d_j = 0$.*

Note, that if $Q = I$ then two directions are Q-conjugate if they are orthogonal.

Lemma 3.3.1. *Let $Q \in \mathbb{R}^{n\times n}$ be symmetric positive definite. If $\{d_j\}_{j=1}^k$, where $k \leq n$, are non-zero and Q-conjugate, then they are linearly independent.*

The proof is an exercise.

Q-Gram-Schmidt If we have a linearly independent set of vectors and apply the Gram-Schmidt process with an inner product in terms of Q, i.e. $\langle \vec{x}, \vec{y} \rangle = \vec{x}^T Q \vec{y}$, the resulting vectors form a Q-conjugate set. We will briefly describe such a process, which call a *Q-Gram-Schmidt process*, and prove as an exercise that the vectors are indeed Q-conjugate. The regular Gram-Schmidt process starts with $\{\vec{v}_1, \ldots, \vec{v}_n\}$

which are linearly independent, sets $u_1 = \vec{v}_1$ and

$$u_k = v_k - \sum_{j=1}^{k-1} \frac{\vec{v}_k^T u_j}{u_j^T u_j} u_j.$$

Replacing each u_j with $u_j/\|u_j\|$ results in an orthonormal set. The Q-Gram-Schmidt process simply takes $u_1 = \vec{v}_1$ as before, and

$$u_k = v_k - \sum_{j=1}^{k-1} \frac{\vec{v}_k^T Q u_j}{u_j^T Q u_j} u_j.$$

The Conjugate Direction algorithm

Suppose we have a set of Q-conjugate directions p_k and have quadratic f, with $Q = Q^T > 0$. As in the previous section regarding quasi-Newton methods, we will later extend the algorithm we develop to non-quadratic functions and expect that the algorithm works well since smooth functions are locally like quadratics via Taylor's theorem.

Note that if $f(\vec{x}) = \frac{1}{2}\vec{x}^T Q\vec{x} + \vec{x}^T b$, that $\nabla f_k = Q\vec{x}_k - b$. Also, note that there can only be at most n non-zero conjugate directions.

Data: $\vec{x}_0$,$\varepsilon > 0$,$f : \mathbb{R}^n \to \mathbb{R}$,quadratic, and $p_0, \ldots, p_{n-1}$ non-zero conjugate directions
if $\|\nabla f(\vec{x}_k)\| \leq \varepsilon$ **then**
  $\vec{x}_k \approx \vec{x}^*$;
  STOP;
end
Let $\alpha_k = -\frac{\nabla f_k^T p_k}{p_k^T Q p_k}$;
Set $\vec{x}_{k+1} \leftarrow \vec{x}_k + \alpha_k p_k$;
Repeat with $k \leftarrow k + 1$;

Algorithm 8: The basic Conjugate Direction algorithm

Lemma 3.3.2. *Suppose $p_0, \ldots, p_{n-1}$ are non-zero Q-conjugate and $\vec{x}_k$ is defined from Algorithm 8. Then if f is quadratic,*

$$\nabla f_k^T p_j = 0, \qquad \text{for all } 0 \leq j \leq k-1.$$

Proof. We will prove this by induction, if $\psi(\alpha) = f(\vec{x}_0 + \alpha p_0)$, then since α_0 minimizes ψ, $0 = \psi(\alpha_0) = \nabla f_1^T d_0$. So the result is true for $k = 1$. Next assume it is true for k and we want to prove it for $k+1$. For $0 \leq j \leq k-1$ we have

$$\begin{aligned} \nabla f_{k+1}^T p_j &= \vec{x}_{k+1}^T Q d_j - b^T p_j \\ &= (\vec{x}_k^T + \alpha_k p_k^T) Q p_j - b^T p_j \\ &= \vec{x}_k^T Q p_j - b^T p_j \\ &= \nabla f_k^T p_j = 0. \end{aligned}$$

The only task left is to prove $\nabla f_{k+1}^T p_k = 0$, but this is obtained from an argument identical to the $k = 1$ case. □

Note that this means that at $\vec{x}_n$, $\nabla f_n^T p_j = 0$ for all $j = 0, \ldots, n-1$. But since this set of conjugate directions form a basis for $\mathbb{R}^n$, $\nabla f_n = 0$, meaning that $\vec{x}_n$ is the unique minimizer for f since it is quadratic.

Conjugate Gradient for quadratic functions

Next, we look at how to construct the directions in Algorithm 8. The conjugate gradient algorithm does so by, at each point, looking at the gradient of f, and selecting a new direction so that the set of selected directions (so far) is Q-conjugate. We are still assuming $f = \frac{1}{2}\vec{x}^T Q \vec{x} - b^T \vec{x}$ with $Q = Q^T > 0$.

- Let $\vec{x}_0$ and $p_0 = -\nabla f_0$.

- The next point will be $\vec{x}_1 = \vec{x}_0 + \alpha_0 p_0$ where

$$\alpha_0 = -\frac{\nabla f_0^T Q p_0}{p_0^T Q p_0}.$$

- We will later show that p_1 given by

$$p_1 = -\nabla f_1 + \beta_0 p_0, \qquad \beta_0 = \frac{\nabla f_1^T Q p_0}{p_0^T Q p_0}$$

  is Q-conjugate to p_0. In general,

$$p_{k+1} = -\nabla f_{k+1} + \beta_k p_k, \qquad \beta_k = \frac{\nabla f_{k+1}^T Q p_k}{p_k^T Q p_k}.$$

To show that β_k is not coming from fiat, note that we know by Lemma 3.3.2 that ∇f_1 is orthogonal to p_0 and so it is linearly independent. Applying Q-Gram-Schmidt to $\{p_0, -\nabla f_1\}$ gives

$$p_1 = -\nabla f_1 - \frac{-\nabla f_1^T Q p_0}{p_0^T Q p_0} p_0 = -\nabla f_1 + \beta_0 p_0$$

For general k, though, Q-Gram-Schmidt to the set $\{-\nabla f_k, p_{k-1}, \ldots, p_0\}$ gives us

$$p_{k+1} = -\nabla f_{k+1} - \frac{-\nabla f_{k+1}^T Q p_k}{p_k^T Q p_k} p_k - \cdots - \frac{-\nabla f_{k+1}^T Q p_0}{p_0^T Q p_0} p_0$$

. To finish the justification of the definition of β_k (which is the coefficient in front of p_k), we need to show at each k that the coefficient for p_j, $0 \leq j \leq k-1$ is 0 in this general formula.

Lemma 3.3.3. *With p_j defined as above, for $0 \leq j \leq k-1$*

$$\nabla f_{k+1}^T Q p_j = 0.$$

Proof. Let $j \leq k-1$ and note that for quadratic f, that $\vec{x}_{j+1} = \vec{x}_j + \alpha_j p_j$ and $Q\vec{x} = \nabla f(\vec{x}) + b$. Then

$$Qp_j = \frac{1}{\alpha_j} Q(\vec{x}_{j+1} - \vec{x}_j) = \frac{1}{\alpha_j}(\nabla f_{j+1} - \nabla f_j).$$

By construction, ∇f_{j+1} and ∇f_j are in the span of $\{p_0, \ldots, p_{d+1}\}$ and thus by Lemma 3.3.2, ∇f_{k+1} is perpendicular to $p_0, \ldots, p_{j+1}$ since $j + 1 \leq k < k + 1$ and so $\nabla f_{k+1}^T Q p_j = 0$. □

The directions $p_0, \ldots, p_{n-1}$ are Q-conjugate since they come from a Q Gram-Schmidt process.

We finish by noting that in the case of a quadratic f, quasi-Newton methods are conjugate gradient methods.

Theorem 3.3.2. *Let f be quadratic and H_{k+1} symmetric matrixs with $H_{k+1}\Delta g_i = \Delta \vec{x}_i$ for $0 \leq k < n-1$, $0 \leq i \leq k$ (using the notation for Δg_i and $\Delta \vec{x}_j$ from section 3.2). Then with $p_j = -H_j \nabla f_j$ and $\vec{x}_{j+1} = \vec{x}_+ \alpha_j p_j$, the set $p_0, \ldots, p_{k+1}$ is Q-conjugate.*

Proof. This is left as an exercise. □

Conjugate Gradient algorithm for non-quadratic functions

To extend the CG algorithm to non-quadratic functions, we first rely on the claim that near a minimizer, we can use a quadratic approximation to f. If we were willing to compute the Hessian at each step, we could use it as Q in the above above formulas to get α_k and β_k. Since we do not want to, we will use a line search for α_k and rely on a choice of β_k which does a "good enough" job mimicking the behavior of the quadratic case. We close with a few schemes for choosing the β_k without computing a Hessian.

Data: $\vec{x}_0$,$\varepsilon > 0$,$f : \mathbb{R}^n \to \mathbb{R}$, $p_0 = -\nabla f_0$)
if $\|\nabla f(\vec{x}_k)\| \leq \varepsilon$ **then**
  $\vec{x}_k \approx \vec{x}^*$;
  STOP;
end
Set $\psi(\alpha) = f(\vec{x}_k + \alpha p_k)$;
Perform a line search using ψ, resulting in α_k;
Set $\vec{x}_{k+1} \leftarrow \vec{x}_k + \alpha_k p_k$;
Set β_k (according to choice of scheme);
Set $p_{k+1} = -\nabla f_{k+1} + \beta k p_k$;
Repeat with $k \leftarrow k + 1$;

Algorithm 9: The CG algorithm for general functions

Hestenes-Steifel Mimicking the quadratic case, we set $Qp_k = \frac{1}{\alpha_k}(\nabla f_{k+1} - \nabla f_k)$ and plug this into the previous formula to get

$$\beta_k = \frac{\nabla f_{k+1}^T(\nabla f_{k+1} - \nabla f_k)}{p_k^T(\nabla f_{k+1} - \nabla f_k)}.$$

Polak-Ribiere We note that in the quadratic case, $p_k^T \nabla f_{k+1} = 0$,meaning we might rely on $\nabla f_k^T p_k = -\nabla f_k^T \nabla f_k$ to rewrite the denominator in the Hestenes-Steifel to get

$$\beta_k = \frac{\nabla f_{k+1}^T(\nabla f_{k+1} - \nabla f_k)}{\nabla f_k^T \nabla f_k}.$$

Fletcher-Reeves

Lemma 3.3.4. *Let $\vec{x}_j$ be a sequence of points generated by the conjugate gradient method. Then for $0 \leq k \leq n-1$, $0 \leq j \leq k$, $\nabla f_{k+1}^T \nabla f_j = 0$.*

Proof. From Lemma 3.3.2,

$$0 = \nabla f_{k+1}^T p_j = -\nabla f_{k+1}^T \nabla f_j + \beta_{j-1} \nabla f_{k+1}^T p_{j-1}$$

which means that$\nabla f_{k+1}^T \nabla f_j = 0$. □

Applying this to the Polak-Ribiere formula, we get

$$\beta_k = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}.$$

Though less robust to errors in the line search than the Polak-Ribiere algorithm, the Fletcher-Reeves algorithm also works quite well. Indeed, Polak-Ribere can be seen as a refinement of Fletcher-Reeves to handle such errors.

3.4 Test Functions

Now that we've found a reasonable collection of algorithms and line search procedures, we might look at generating test problems which sheds some light on the relative strengths and weaknesses of each method. Note: the below functions do not necessarily have $D^2 f(x) > 0$ for all x.

- The **Beale Function**: $f(x) = (1.5 - x_1 - x_1 x_2)^2 + (2.5 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$ has a minimum near $(2.6, -0.35)^T$. There are a total of four points where $\nabla f = 0$.
- The **Booth Function**: $f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$ is quadratic so Newton's method will find a minimum but other methods may have trouble finding it.
- **Hump Function**: $f(x) = 4x_1^2 - 2.1x_1^4 + \frac{x_1^6}{3} + x_1 x_2 - 4x_2^2 + 4x_2^4$ has 15 stationary points where $\nabla f = 0$, but the true minimum is near $\pm(0.1, -0.7)^T$.

- The **Rosenbrock Function**: $f(x) = (a - x_1)^2 + b(x_2 - x_1^2)$ has a global minimum at $(a, a^2)^T$. Good tests come up when $b \geq 100a > 0$.

- The **2D Perm Function I** $f(x) = \sum_{k=1}^{2} \sum_{j=1}^{2} (j^k + 10)(\frac{x_j^k}{j^k} - 1)$ has two stationary points which are global minima.

- The **2D Perm Function II** $f(x) = \sum_{k=1}^{2} \sum_{j=1}^{2} (j + 10)(x_j^k - \frac{1}{j^k})$ has two stationary points which are global minima.

3.5 Exercises

1. Let $f(t) = \log t + 6t$, which has a zero on $(0, 1)$ and is increasing on the same interval. For any $a \in \mathbb{R}$, define $F(t) = \int_a^t f(s)ds$; the zero of f, then, coincides with the minimum of F. If we pick a appropriately, we can compute $F(t)$ by hand; below we look to minimize it.

    - Perform several iterations of an Armijo line search (what is the derivative of F?) by hand starting at $t = 0.2$ and using an initial step length of 0.2.
    - Perform several iterations of Newton's Method by hand with $t_0 = 1$.

2. If $\tilde{c} < c$ and $c, \tilde{c} \in (0, 1)$, can we always find a step size satisfying the Wolfe conditions?

3. Define $f(t) = \int_0^t e^{s^2 \sin t} ds$. Compute $f'(t)$; can we find a minimizer t analytically ? Now, use an Armijo line search to find a t such that $|f'(t)| < 10^{-2}$. Next, use Newton's method to find such a point (you need to recall that we need to start near the minimizer).

4. Let $f(t) = .01(t^6 - 30t^4 + 192t^2 + 7t^4)$. If you use Newton's method, what are the results when $t_0 = -8$ and when $t_0 = -3$. What is the source of the discrepancy?

5. Prove that if $\{x_k\}$ is a steepest descent sequence and $\nabla f(x_k) \neq 0$, then $f(x_{k+1}) < f(x_k)$.

6. Let $f(x) = \frac{1}{2}x^T Q x - b^T x$ where Q is symmetric, positive definite.

   - What is $\nabla f(x)$?
   - What is the minimizer of f, analytically.
   - Analytically find the step size in the steepest descent method which minimizes $\psi(\alpha)$.
   - What is the explicit formula for the steepest descent update x_{k+1}?
   - If we can take an exact line search, and the search direction is $-\nabla f_k$, prove that if $x_0 - x^*$ (where x^* is a minimizer of f) is parallel to an eigenvector of Q, then we only need one line search to find a minimum of f.

7. Let $f(x) = (x_1 + 2x_2 - 2)^2 + (2x_1 + x_2 - 1)^2$ and $x_0^T = [0, 0]$.

   - What is the first search direction using steepest descent?
   - Analytically find the best step size and use this to find $\vec{x}_1$.
   - Find $\vec{x}_1$ by analytically performing Newton's method. What is $\nabla f(\vec{x}_1)$?
   - If we were using a quasi-Newton method with $H_0 = I$, what would x_1 be?
   - Update H_1 using the DFP algorithm.
   - What is the next search direction using the DFP algorithm?

8. Prove the following:

   Theorem 3.5.1. *Let* $f(x) = \frac{1}{2}x^T Q x - b^T x$ *where* $Q = Q^T$. *Assume that for* $0 \leq k < n-1$, H_k *are symmetric matrices satisfying*

   $$H_{k+1}\big(\nabla f(x_{j+1}) - \nabla f(x_j)\big) = x_{j+1} - x_j$$

   for $0 \leq j \leq k$. *With* $d_0 = -\nabla f(x_0)$, $d_j = -H_j \nabla f(x_j)$ *for* $j > 0$, *and* $x_{j+1} = x_j + t_j d_j$, *(where* $t_j > 0$*), the set* $\{d_0, \ldots, d_{k+1}\}$ *is* Q*-conjugate.*

9. Let $f(t) = .01(t^6 - 30t^4 + 192t^2 + 7t^3)$. If you use Newton's method, what are the results when $t_0 = -7$ and when $t_0 = -3$. What is the source of the discrepancy?

Chapter 4

Variational Problems

Consider the following set of questions:

1. A ball is to slide along a frictionless track from the point $(0, Y)$ to $(X, 0)$, with gravity the only active force. What shape should the track be so as to minimize the time it takes for the ball to travel from start to finish? This is the *brachistochrone problem*.

2. Let S be a surface in $\mathbb{R}^3$ and let $x, y \in S$ be two points. What is the *geodesic*—the path of minimial distance connecting x and y while staying within S?

3. What shape will a cable—hung between two supports—take if the only force acting on it is due to its own weight?

4. Given two wire rings, what is the surface of minimal surface area that will meet up with the wire rings?

The common theme is that these questions all are minimization problems where the decision variable we must select is a function which satisfies some restrictions. For instance, in the brachistochrone problem, the minimizer is a function $f : \mathbb{R} \to \mathbb{R}$ which satisfies $f(0) = Y$, $f(X) = 0$ and we might require that $f \in C^k(\mathbb{R})$ for $k \geq 1$. In the

geodesic problem, we want a parameterized curve $r : [0, T] \to \mathbb{R}^3$ such that $r(t) \in S$ for all t and $r(0) = x$ and $r(T) = y$.

Before moving to the theory of solving these problems, we first will look at how to write them as optimization

The Straight line Our first variational problem, which is a particular version of the geodesics problem, is: what path between points $x, y\mathbb{R}^2$ has the shortest length? Let's assume that $x_1 \neq y_1$ and that the path can be written as the graph of a function $y = y(x)$. Since the length of the path is the integral of $\sqrt{1 + (\dot{y})^2}$, the problem is to find

$$\text{minimize} \quad \int_{x_1}^{y_1} \sqrt{1 + (\dot{y})^2} dx, \qquad \text{such that } (y(0), y(L)) = (y_0, y_1).$$

Geodesics on a cylinder Suppose that now we have two points on the cylinder $C = \{x_1^2 + y_1^2 = 1\} \subset \mathbb{R}^3$. For ease, let's assume $x = (1, 0, 0)$ and $y \in C$ is som other point; we will use cylindrical coordinates (r, θ, z). Suppose $y = (1, \tilde{\theta}, \tilde{z})$ where, without loss of generality $\tilde{\theta} \in [0, \pi]$. If we can expect the geodesic to be monotonic in θ, then z can be expressed as $z(\theta)$ and we get something very similar to the previous problem:

$$\text{minimize} \quad \int_0^{\tilde{\theta}} \sqrt{1 + (\dot{z})^2} dx, \qquad \text{such that } (z(0), z(\tilde{\theta})) = (0, \tilde{z}).$$

The Brachistochrone problem We will assume, again, that the curve is monotonic in x, and so the curve the ball will travel along is the graph of a function $y(x)$. At any x, we will denote the velocity of the ball with $v(x)$. Over the interval Δx, the bead will move Δs along the path during the time $\Delta s / v(x)$. But $\Delta s \approx \sqrt{1 + (\dot{y}(x))^2} \Delta x$ and so the time needed to move Δx is approximately $\sqrt{1 + (\dot{y}(x))^2} / v(x) \Delta x$.

Summing over all Δx and taking a limit, the total time the ball needs to travel is

$$\int_0^X \frac{\sqrt{1+(\dot{y}(x))^2}}{v(x)}dx. \tag{4.1}$$

Next, we need to determine $v(x)$; we'll use convervation of energy and assume points on the x-axis have "zero" potential energy. At $(0, Y)$, the ball is at rest (and so with zero kinetic energy) with potential energy mgY where m is its mass and g is the acceleration due to gravity. At any point along the path (x, y), the ball will have kinetic energy $\frac{mv(x)^2}{2}$ and potential energy mgy. Conservation of energy means

$$\frac{mv(x)^2}{2} + mgy = mgY \quad \Rightarrow \quad v = \sqrt{2g(Y-y)}.$$

Plugging this in, we get

$$\text{minimize } \int_0^X \sqrt{\frac{1+(\dot{y}(x))^2}{2g(Y-y)}}dx \qquad \text{such that } (y(0), y(X)) = (Y, 0).$$

Minimal surface of revolution Consider finding a function $u(x)$ such that $u(a) = A$ and $u(b) = B$ and the surface by revolving $u(x)$ about the x-axis, restricted to $x \in [a, b]$, is of minimal surface area. We get

$$\text{minimize } \int_a^b u(x)\sqrt{1+(\dot{u})^2}dx, \qquad \text{such that } (u(a), u(b)) = (A, B).$$

Hanging cable Suppose we hang a uniform cable of length L between supports separated by the distance $2D$. The cable will take the shape minimizing the total potential energy. Assume that this shape is given by $u : [-D, D] \to \mathbb{R}$ and the density of the cable is $\rho\frac{kg}{m}$. Then

the total potential energy is given by

$$\int_{-D}^{D} \rho g u(x) \sqrt{1 + (\dot{u}(x))^2} dx.$$

Recalling we have a fixed length of this cable, we get the minimization problem

$$\text{minimize } \int_{-D}^{D} u(x)\sqrt{1 + (\dot{u}(x))^2}dx \quad \text{s.t. } u(-D) = u(D) = 0, \quad \int_{-D}^{D} \sqrt{1 + (\dot{u}(x))^2}dx = L. \tag{4.2}$$

4.1 Problem formulation

Note that all of the above examples involved finding an optimal function of a single variable. However, while we will continue to focus on examples involving functions of a single variable, we will give the theory for functions of several variables. Given an $\Omega \subset \mathbb{R}^n$, and a *Lagrangian* $F : \Omega \times \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$, we look to minimize the cost functional

$$J(u) = \int_{\Omega} F(x, u(x), \nabla u(x))dx$$

where u is differentiable and satisfies some constraints.

Notationally, we will use the variables $(x, \lambda, \xi) \in \Omega \times \mathbb{R} \times \mathbb{R}^n$ for the Lagrangian $F(x, \lambda, \xi)$. We will assume F is differentiable with respect to x and twice differentiable with respect to λ and ξ.

4.2 The Euler-Lagrange equation

The Euler-Lagrange (E-L) equation gives a necessary condition for optimizers of a variational problem. The form of the equation depends

on the constraints on u.

Theorem 4.2.1. *Let $\Omega \in \mathbb{R}^n$ be a bounded open set with piecewise smooth boundary $\partial\Omega$, and $u_0 : \partial\Omega \to \mathbb{R}$ be given. Consider the variational problem:*

$$\text{minimize } J(u) := \int_\Omega F(x, u(x), \nabla u(x))dx, \quad \text{s.t.} \quad u = u_0 \text{ on } \partial\Omega.$$

If u is an optimal solution, then u is also a solution to (E-L), which takes the form:

$$div\big(F_\xi(x, u(x), \nabla u(x))\big) = F_\lambda\big(x, u(x), \nabla u(x)\big), \qquad x \in \Omega \qquad (4.3)$$

$$u = u_0 \qquad x \in \partial\Omega. \qquad (4.4)$$

Note that the left hand side of Equation (4.3) should be read as taking the gradient of F with respect to ξ, evaluating at $\lambda = u(x), \xi = \nabla u(x)$, and then taking the divergence of the resulting vector-valued function of x.

The proof of Theorem 4.2.1 will require the following lemma:

Lemma 4.2.1. *If $f : (a, b) \to \mathbb{R}$ is continuous and $\int_a^b f(x)\phi(x) = 0$ for all differentiable ϕ with $\phi(a) = \phi(b) = 0$, then $f(x) = 0$ for all $x \in (a, b)$.*

Proof. By continuity, given any $\varepsilon > 0$ and $x_0 \in (a, b)$, there exists $\delta > 0$ such that $|f(x)-f(x_0)| < \varepsilon$ for all $|x-x_0| < \delta$. Suppose that $f(x_0) \neq 0$; without loss of generality, we assume $f(x_0) > 0$ and let $\varepsilon = \frac{f(x_0)}{2}$ and the $\delta > 0$ from the continuity statement. Then, on $(x_0 - \delta, x_0 + \delta)$, $|f(x) - f(x_0)| < \frac{f(x_0)}{2}$which implies $f(x) > f(x_0) - \frac{f(x_0)}{2} = \frac{f(x_0)}{2}$. Let ϕ be differentiable which is 0 outside of $(x_0 - \delta, x_0 + \delta)$ and strictly positive inside of the interval. Then

$$0 = \int_a^b f(x)\phi(x)dx = \int_{x_0-\delta}^{x_0+\delta} f\phi dx \geq \frac{f(x_0)}{2} \int_{x_0-\delta}^{x_0+\delta} \phi dx > 0$$

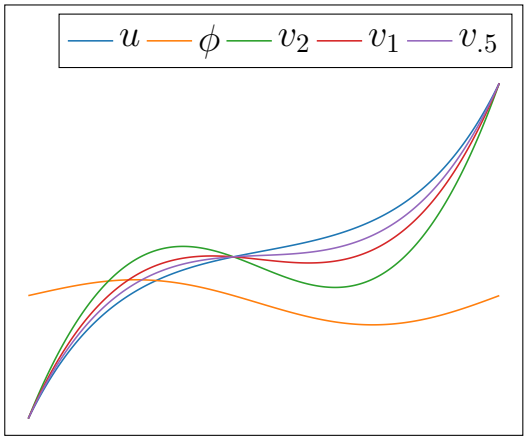

Figure 4.1: A function u and the variations v_t corresponding to direction ϕ

which contradicts the assumption we made on f. This means no such x_0 can exist and so $f(x) = 0$ for all x. □

Proof of Theorem 4.2.1. We first prove Theorem 4.2.1 when $u : \mathbb{R} \to \mathbb{R}$ and $u(a) = A$, $u(b) = B$. Let $\phi : [a, b] \to \mathbb{R}$ be an arbitrary differentiable function with $\phi(a) = \psi(b) = 0$ and $\varepsilon > 0$. Then for any $-\varepsilon < t < \varepsilon$ the function $v_t(x) = u(x) + t\phi(x)$ is admissible. The function v_t is a variation of $u(x) = v_0(x)$; see Figure 4.1. Now, since u is optimal, we have for all t that

$$J(v_0) \leq J(v_t).$$

Defining $g(t) := J(v_t)$, we get that the optimality of u means $g'(0) = 0$. The assumptions on the differentiability of F mean we can move derivatives in and out of the integral as needed:

$$0 = \frac{d}{dt}|_{t=0} J(v_t) = \int_a^b \big(F_\lambda(x, u(x), u'(x))\phi(x) + F_\xi(x, u(x), u'(x))\phi'(x)\big)dx.$$

Integrating by parts on the second part of the above integral, and

recalling $\phi(a) = \phi(b) = 0$, we have

$$0 = \int_a^b \big(F_\lambda(x, u(x), u'(x)) - \frac{d}{dx} F_\xi(x, u(x), u'(x))\big)\phi(x)dx \tag{4.5}$$

which, by Lemma 4.2.1, means Equation (4.3) holds for the one dimensional case.

Next, we look at how to prove Theorem 4.2.1 in the general case. The proof largely remains the same. Again, we have a $\phi : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ where now we need $\phi = 0$ on $\partial\Omega$. The definition of v_t is unchanged, and we define $g(t)$ as above. Again, $g'(0) = 0$ meaning

$$0 = \frac{d}{dt}|_{t=0} J(v_t) = \int_a^b \big(F_\lambda(x, u(x), \nabla u(x))\phi(x) + \nabla_\xi F(x, u(x), \nabla u(x)) \cdot \nabla\phi(x)\big)dx.$$

The main difference is that rather than integration by parts, we now make use of the Divergence Theorem (Gauß' theorem) to the vector field $\phi(x)\nabla_\xi F(x, u(x), \nabla u(x))$:

$$\int_\Omega \nabla_\xi F \cdot \nabla\phi dx = -\int_\Omega \phi \mathrm{div}(\nabla_\xi F)dx + \int_{\partial\Omega} \phi \nabla_\xi F \cdot dA.$$

But since $\phi = 0$ on $\partial\Omega$, inserting this into section 4.2 gives

$$0 = \int_\Omega \Big(F_\lambda(x, u(x), \nabla u(x)) - \mathrm{div}(F_\xi(x, u(x), \nabla u(x)))\Big)\phi dx$$

which implies Equation (4.3) by Lemma 4.2.1. □

Having stated the general form of Theorem 4.2.1, we turn to a couple of special cases in the one-dimensional case, i.e. where $F : (a, b) \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$.

Special Case I: When $F(x, \lambda, \xi) = F(\xi)$, as in the geodesic problem, Equation (4.3) becomes $\frac{d}{dx}F_\xi(\dot{u}(x)) = 0$. Expanding, then we have

$$F_{\xi\xi}(\dot{u}(x))\ddot{u}(x) = 0.$$

Suppose u is optimal and so this equality holds. If $F_{\xi\xi} \equiv 0$, then any admissible u satisfies the Euler-Lagrange equations, meaning I is constant over all admissible functions. If $F_{\xi\xi}(\dot{u}(x_0)) \neq 0$, then it can be shown that $u''(x) = 0$ for all $x \in (a, b)$.

Special Case II: Suppose $F(x, \lambda, \xi) = F(\lambda, \xi)$. Then,

$$\frac{dF}{dx} = \frac{\partial F}{\partial x} + \frac{\partial F}{\partial \lambda}\dot{u} + \frac{\partial F}{\partial \xi}\ddot{u} = \frac{\partial F}{\partial \lambda}\dot{u} + \frac{\partial F}{\partial \xi}\ddot{u}.$$

The Euler-Lagrange equation then says

$$\frac{d}{dx}\Big(\frac{\partial F}{\partial \xi}\Big) = \frac{\partial F}{\partial \lambda}$$

which means

$$\dot{u}\frac{d}{dx}\Big(\frac{\partial F}{\partial \xi}\Big) = u'\frac{\partial F}{\partial \lambda} = \frac{\partial F}{\partial x} - \frac{\partial F}{\partial \xi}\ddot{u}.$$

Rearranging, we get

$$0 = \frac{\partial F}{\partial x} - \dot{u}\frac{d}{dx}\Big(\frac{\partial F}{\partial \xi}\Big) - \ddot{u}\frac{\partial F}{\partial \xi} - u''\frac{\partial F}{\partial \xi} = \frac{d}{dx}\Big(F - u'\frac{\partial F}{\partial \xi}\Big).$$

This gives *Beltrami's identity*

$$F - u'\frac{\partial F}{\partial \xi} = C \tag{4.6}$$

where C is a constant. That is, if u is a solution to Equation (4.3), then it also solves Equation (4.6); we have not verified the converse (which is, in fact, not true in general). However, it can be shown that if u satisfies Equation (4.6) but not Equation (4.3), then u must be a linear function. In some cases, Beltrami's identity gives an easier necessary condition to work with.

4.3 Natural boundary conditions

Next, we consider the problem where $\Omega = (a, b)$ and one end is "free":

$$\text{minimize } J(u) := \int_a^b F(x, u(x), \dot{u}(x))dx, \quad \text{s.t.} \quad u(a) = A.$$

Because we have a less restrictive set of requirements on u, we might expect that Equation (4.3) and $u(a) = A$ do not give enough information to identify the optimal u.

Indeed, when we derived Equation (4.3) in the proof of Theorem 4.2.1, we relied on integration by parts. We have no reason now to require that our $\phi(b) = 0$, meaning we get by integration by parts of $0 = g'(0)$ that

$$\begin{aligned} 0 = \int_a^b \big(F_\lambda(x, u(x), u'(x)) - \frac{d}{dx} F_\xi(x, u(x), u'(x))\big)\phi(x)dx \\ + F_\xi(b, u(b), \dot{u}(b))\phi(b). \end{aligned} \tag{4.7}$$

Now, if $\phi(b) = 0$ *does* have $\phi(b) = 0$ we still have that

$$0 = \int_a^b \big(F_\lambda(x, u(x), u'(x)) - \frac{d}{dx} F_\xi(x, u(x), u'(x))\big)\phi(x)dx$$

which gives us the original Equation (4.3) via Lemma 4.2.1. So let's look at Equation (4.7) with Equation (4.3) also holding. This means that $F_\xi(b, u(b), \dot{u}(b))\phi(b) = 0$. Since $\phi(b)$ is arbitrary, we have that the additional boundary condition which should be satisfied is

$$F_\xi(b, u(b)\dot{u}(b)) = 0;$$

this is the "natural" boundary condition at b. The same procedure could be applied to the endpoint at $x = a$.

Besides getting an optimality condition for free boundary problems, we also get optimality conditions for problems where the constraint is

in the form of an inequality. Suppose that we require $B_1 \leq u(b) \leq B_2$, rather than $u(b) = B$. Now, we could first find the (possibly) optimal solution when the boundary at $x = b$ is free. If that solution u satisfies $B_1 \leq u(b) \leq B_2$, we are done. Otherwise, we have either $u(b) = B_1$ or $u(b) = B_2$, which gives us two variational problems with fixed boundary conditions. We can then solve these two problems and see which problem's minimizer gives the lowest optimal value of J.

4.4 Integral-constrained problems

Finally, we look at variational problems which have not just boundary conditions but integral constraints; such a constraint showed up in the hanging cable problem we introduced at the outset of this chapter. Specifically, for a u to be admissible we will also require that, given $H : (a, b) \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^n$ and $\beta \in \mathbb{R}^n$,

$$I(u) := \int_a^b H(x, u(x), \dot{u}(x))dx = \beta. \tag{4.8}$$

Remark 4. *Note that if we have both integral and fixed endpoint constraints, it may be easier from a notation/book-keeping perspective to rewrite $u(a) = A$ and $u(b) = B$ as*

$$u(a) = A, \qquad \int_a^b \dot{u}(x)dx = B - A.$$

This change makes no difference in the solution, it is only sometimes more convenient.

Integral constraints look a bit more like we are in constrained optimization and might want to use something like Lagrange's theorem. However, proving Lagrange's theorem for infinite dimensional problems (like variational problems) involves using the L^2 inner product and some other notions which are outside the content of this course.

Roughly, the idea is that we require the variation v_t to be admissible for any small enough t. Differentiating $g(t)$ as in the proof of Theorem 4.2.1, we get Equation (4.5); differentiating $I(v_t) = \beta$ and looking at it at $t = 0$ gives for each row in H

$$\int_a^b ((H_i)_\lambda(x, u, \dot{u}) - \frac{d}{dx}(H_i)_\xi(x, u, \dot{u}))\phi dx = 0$$

One can then establish that the integrand in Equation (4.5) is a linear combination of the rows of this integrand. That is,

$$0 = F_\lambda(x, u, \dot{u}) - \frac{d}{dx}F_\xi(x, u, \dot{u}) + c^T\big(H_\lambda(x, u, \dot{u}) - \frac{d}{dx}H_\xi(x, u, \dot{u})\big) \quad (4.9)$$

Typically, we define the *augmented Lagrangian*

$$\tilde{F}(x, \lambda, \xi; c) := F(x, \lambda, \xi) + c^T H(x, \lambda, \xi).$$

Theorem 4.4.1. *Consider the optimization problem of minimizing*

$$J(u) = \int_a^b F(x, u(x), \dot{u}(x))dx$$

subject to $u(a) = A, u(b) = B$ *and* $\int_a^b H(x, u(x), \dot{u}(x))dx = \beta$. *Suppose there exists a vector* c *such that* $\tilde{F}_{\xi\xi} \geq 0$ *for each* $x \in (a, b)$ *and* $\lambda \in \mathbb{R}$. *If* u *uniquely solves*

$$\frac{d}{dx}\big[\tilde{F}_\xi(x, u(x), \dot{u}(x))\big] = \tilde{F}_\lambda(x, u(x), \dot{u}(x))$$

and u *is admissible, then* u *is optimal.*

Note that solving the Euler-Lagrange equation for $\tilde{F}$ will result in having u in terms of c; these parameters are determined by then imposing the integral constraints on the u we have obtained to then ensure u is admissible.

4.5 Example problems

Here we'll look at how the above necessary conditions can be used to obtain solutions to some variational problems.

A simple example Consider the problem of minimizing $\int_0^1 \left[\frac{1}{2}\big(u'(x)\big)^2 + xu(x)\right]dx$ subject to $u(0) = u(1) = 1$. The Lagrangian is $F(x, \lambda, \xi) = \frac{1}{2}\xi^2 + x\lambda$, meaning we can't make use of Beltrami's identity. The Euler-Lagrange equation, using $\partial_\xi F = \xi$, $\partial_\lambda F = x$, and $\frac{d}{dx}(\partial_\xi F(x, u(x), u'(x))) = \frac{d}{dx}u'(x) = u''(x)$, gives

$$u''(x) = x.$$

Integrating twice, this ODE has a solution $u(x) = \frac{x^3}{6} + c_1 x + c_2$. Requiring $u(0) = u(1) = 1$ means $c_2 = 0$ and $c_1 = -1/6$.

Notice that using the Euler-Lagrange equation resulted in a second order ODE to be solved. Typically, closed form solutions are only available if this ODE is linear with constant coefficients. If Beltrami's identity is available, the ODE obtained will be first order; there are many more such ODEs whose solution are explicitly solvable. In face, the ODE is separable and can be solved by substitutions typically.

Geodesic on a cylinder Recall we want to find a path of minimal length joining $(1, 0, 0)$ and $(1, \theta_0, z_0)$ (in cylindrical corrdinates). In Cartesian coordinates, we want a $\sigma(\theta) = (\cos\theta, \sin\theta, z(\theta))$, $\theta \in (0, \theta_0)$. That is, we want $z(\theta)$ which gives minimal path length; the corresponding Lagrangian is $F(\xi) = \sqrt{1 + \xi^2}$. Since this Lagrangian is independent of θ, λ, we have $z(\theta) = \alpha\theta + \beta$, the boundary condtions say we need $z(\theta) = \frac{z_0}{\theta_0}\theta$.

The Brachistochrone Problem Assume that we want the path to start at $(0, 1)$ and end at $(x_1, 0)$ and we write $y(x)$ as the height of the

ramp at each x. Then the Lagrangian is

$$F(x, \lambda, \xi) = \sqrt{\frac{1+\xi^2}{1-\lambda}}$$

which means we may use Beltrami's identity. Applying this, we get

$$\begin{aligned} C = F - y'\partial_\xi F(x, y, y') &= \frac{\sqrt{1+(y')^2}}{\sqrt{1-y}} - \frac{(y')^2}{\sqrt{(1-y)(1+(y')^2)}} \\ &= \frac{1}{\sqrt{(1-y)(1+(y')^2)}}; \end{aligned}$$

relabeling C, we have

$$(1-y)(1+(y')^2) = C \quad \Rightarrow \quad \frac{dy}{dx} = \pm\sqrt{\frac{C}{1-y} - 1}.$$

This is separable, giving

$$\pm\sqrt{\frac{1-y}{C-1+y}}dy = dx.$$

If we use a trigonometric substitution, and set $1 - y = C\sin^2(\phi/2)$, we can use $\frac{dy}{d\phi} = -C\sin(\phi/2)\cos(\phi/2)$ to get

$$\pm\sqrt{\frac{C\sin^2(\phi/2)}{C(1-\sin^2(\phi/2))}}C\sin(\phi/2)\cos(\phi/2)d\phi = dx.$$

This expression can be simiplified to

$$\frac{dx}{d\phi} = \pm\frac{C}{2}(1-\cos(\phi))$$

and so

$$x = \pm\frac{C}{2}(\phi - \sin\phi) + \tilde{C}.$$

Note that since we have the bead dropping down from $y = 1$, we expect $C > 0$. Since $x(\phi)$ is an increasing function, $x'(\phi)$ is positive for $\phi > 0$ and $x(0) = \tilde{C} = 0$, we have

$$x = \frac{C}{2}(\phi - \sin\phi) \qquad y = \frac{C}{2}(\cos\phi - 1) + 1. \tag{4.10}$$

This pair of parametric equations is that of a cycloid. A cycloid is the result of rolling a wheel of radius $C/2$ upside down along the bottom of $y = 1$ with center moving along $1 - C/2$. The wheel rolls counter-clockwise and $(x(\phi), y(\phi))$ is the path traced by the point on the rim of the wheel which starts at $(0, 1)$.

Note that for a given x_1, we need $x(\phi_1) = x_1$. So we need to determine both C and ϕ_1 such that $x(\phi_1) = x_1$ and $y(\phi_1) = 0$. To obtain this we have

$$\cos\phi_1 = 1 - \frac{2}{C}$$

and since $-1 \leq \cos\phi \leq 1$, we have $C \geq 1$. From the definition of arccos, we have to consider both $\phi_1 = \arccos(1 - 2/C)$ and $\phi_1 = 2\pi - \arccos(1-2/C)$. Simplifying the resulting expression after we plug these into x, we have

$$x_1 = \frac{C}{2}\arccos(1 - \frac{2}{C}) - \sqrt{C-1} = g_1(C);$$
$$x_1 = \frac{C}{2}(2\pi - \arccos(1 - \frac{2}{C})) + \sqrt{C-1} = g_2(C).$$

At $C = 1$, $g_1 = g_2 = 1$. For $C \to \infty$, $g_1(C) \to 0$ with $g_1'(C) < 0$ and $g_2(C) \to \infty$ with $g_2'(C) > 0$. Thus if $x_1 \in (0, \pi/2)$, we should solve $x_1 = g_1(C)$ and otherwise we should solve $x_1 = g_2(C)$. Once we have found C, we have ϕ_1 and so we have the cycloid going through the desired points and thus the solution for the Brachistochrone problem.

4.6 Exercises

1. Find a u that minimizes $\int_0^1 (u'(x)^2 + 2u(x)^2)e^x dx$ subject to $u(0) = 1$, $u(1) = 0$.

2. Find a u that minimizes $\int_0^\pi (u'(x)^2 - .5u(x)^2)e^x dx$ subject to $u(0) = 1$, $u(\pi) = 2$.

3. Let $r(t) = \big(x(t), y(t)\big)$ be a curve from (x_0, y_0) to (x_1, y_1) with $x_1 > x_0$ whose trace is not the graph of a function. Prove that this cannot be a geodesic. First, show that if there are $t_1 < t_2$ such that $x(t_1) = x(t_2)$ then the path must include a vertical line. Then show we could reduce the path length by ommitting the vertical segment.

4. Prove that if $F : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ is continuous and $\int_\Omega F(x)\phi(x)dx = 0$ for all continuous $\phi : \Omega \to \mathbb{R}$ with $\phi|_{\partial\Omega} = 0$, then $F = 0$ on Ω.

5. Prove the special case I on for the geodesic problem.

6. Assuming sufficient smoothness of all functions involved, prove that if f minimizes the problem

$$\min \int_a^b F(x, f(x), f'(x), f''(x))dx, \quad \text{s.t. } (f(a), f(b), f'(a), f'(b)) = (A, B, C, D),$$

   then f satisfies the second order Euler-Lagrange Equation

$$F_\lambda(x, f, f', f'') - \frac{d}{dx}F_\xi(x, f, f', f'') + \frac{d^2}{dx^2}F_\mu(x, f, f', f'') = 0$$

   where F uses the variables $F(x, \lambda, \xi, \mu)$.

7. Suppose we have a car with a rocket attached to its front and back. To slow it down, we have to fire the rocket in the opposite direction. So deceleration and acceleration burn fuel in the same

manner. We want to minimize the fuel used to move from $u = 0$ to $u = 1$ in time 1, with the car beginning and ending with zero velocity. Minimize the fuel consumption, and plot the resulting u, u' trajectory. That is minimize:

$$\min \int_0^1 (u'(t)^2)+(u''(t))^2)dt \quad \text{s.t. } u(0) = 0, u(1) = 1, u'(0) = u'(1) = 0.$$

Chapter 5

Convex Optimization

In this chapter, we will look at the role convexity plays in optimization. Specifically, we will focus on two questions. First, in the previous chapters, we obtained necessary conditions, such as the KKT or Euler-Lagrange equations, for an optimizer; now we ask if *sufficient conditions* are possible. Second, we look at how we might handle optimization problems where the objective is not differentiable; are there calculus-like rules available and, if so, can they be used to obtainnecessary/sufficient conditions?

5.1 Convexity with differentiable data

If the feasible set is unbounded and we're restricting ourselves to one-dimensional problems, what is the necessary condition for optimality for Equation (2.1)? It's $f'(x^*) = 0$, the KKT criteria in one diension. But, of course, this is only necessary, what might we expect to give sufficient conditions?

- If $f(x) \to \infty$ as $x \to \pm\infty$, then at least one solution to $f'(x) = 0$ exists and one of those solutions is a global minimum.

- If $f''(x) > 0$ for all x, then f is strictly convex and $f'(x) = 0$ has at most one solution.

So if both properties hold, then solving $f'(x) = 0$ gives the unique global minimum.

Definition 5.1.1. *A set $K \subseteq \mathbb{R}^n$ is convex if for any $x, y \in K$, then $tx + (1-t)y \in K$ for any $t \in [0,1]$.*

In other words, a convex set is one where if x and y are in the set, the entire line segment between them is as well.

Definition 5.1.2. *Given convex $K \subseteq \mathbb{R}^n$, a function $f : K \to \mathbb{R}$ is convex if for any $x, y \in K$,*

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y), \qquad t \in [0,1].$$

It is strictly convex if we can replace in the above inequality $\leq$ with $<$ for any $t \in (0,1)$.

That is, a function is convex if the epigraph—the set of all points lying on or above the graph of the function—is a convex set.

Theorem 5.1.3. *If $f : K \subseteq \mathbb{R}^n \to \mathbb{R}$ is convex, then x^* is a local minimizer for f if and only if it is a global minimizer on K.*

Proof. The "if" is obvious. Now suppose that x^* is a local minimizer. Then there is an $\varepsilon > 0$ such that $f(x^*) \leq f(x)$ for any $x \in B_\varepsilon(x^*) \cap K$. Let $y \in K$ be arbitrary. For $t \in [0, \varepsilon/\|x^* - y\|]$, we know $(1-t)x^* + ty \in B_\varepsilon(x^*)$ and convexity gives

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

This in turn means $tf(x^*) \leq tf(y)$. If $t > 0$, we are done. □

While there is a nice geometric intuition for what makes a function convex, it might be advantageous to develop criteria for checking convexity when the function is differentiable.

Proposition 5.1.1. *Let $f : K \subseteq \mathbb{R}^n \to \mathbb{R}$ with K convex and open.*

- *If $f \in C^1$, then it is convex if and only if $f(y) \geq f(x)+Df(x)(y-x)$ for all $x, y \in K$.*
- *If $f \in C^2$, then it is convex if and only if $D^2f(x)$ is positive semi-definite for all $x \in K$.*

Proof. We will only prove the first part. For any $x, y \in K$, define $v = (y-x)/\|y-x\|$. Then for $t \in (0, \|x-y\|)$, the point $x+tv$ is on the line segment connecting x and y; convexity means $f(x+tv)$ is less than or equal to the value of the linear function which passes through $(x, f(x))^T$ and $(y, f(y))^T$. That is,

$$f\big(x + t\frac{y-x}{\|y-x\|}\big) \leq f(x) + t\frac{f(y)-f(x)}{\|x-y\|}.$$

Next notice that

$$Df(x)\dot{v} = \lim_{t\to 0^+} \frac{f(x+tv)-f(x)}{t} \leq \frac{f(y)-f(x)}{\|y-x\|}$$

which gives the desired inequality in the first part.

For the other direction in part 1, let $x, y \in K$ and $t \in (0,1)$. Then $w = tx + (1-t)y \in K$ and we assume

$$f(x) \geq f(w) + Df(w)(x-w) \qquad f(y) \geq f(w) + Df(w)(y-w).$$

Multiple the first inequality by t and the second inequality by $1-t$ and add the results to get that $tf(x) + (1-t)f(y) \geq f(tx + (1-t)y)$. □

We now turn to sufficient conditions for optimality; primarily we need the feasible set in Equation (2.4) to be convex and f to be convex on the feasible set. One version of this kind of result is given as:

Theorem 5.1.4. *Assume that in Equation* (2.4), $h \equiv 0$ *and* f, g_j *are* $C^1(\mathbb{R})$ *and convex for* $1 \leq j \leq p$. *Suppose there are* x^* *and* μ *satisfying the conditions in Theorem 2.2.2. Then* x^* *is a global minimizer of* f *subject to* $g(x) \leq 0$.

Proof. Let y be such that $g(y) \leq 0$. Then

$$\begin{aligned} f(y) &\geq f(x^*) + Df(x^*)(y - x^*) \\ &= f(x) - \mu^T Dg(x^*)(y - x^*) \\ &\geq f(x^*) + \mu^T(g(x^*) - g(y)) \\ &= f(x^*) - \mu^T g(y) \\ &\geq f(x^*). \end{aligned}$$

Here we relied on, for each respective line: the convexity of f, the second KKT condition, the convexity in each component of g and the first KKT condition, the third KKT condition, and the feasibility of y and $\mu \geq 0$. □

As a brief remark, if we have equality constraints with $h \not\equiv 0$, we need h_i to be linear for each i to obtain a similar statement for the corresponding general result.

Convexity for variational problems This might lead us to ask if a similar type of statement holds for variational problems. Indeed, such a statement holds and is exactly what we might expect:

Theorem 5.1.5. *Assume that the hypotheses of Theorem 4.2.1 hold and that for any* $x \in \Omega$, F *is a convex function of* λ *and* ξ. *Then if* u *satisfies Equations* (4.3) *and* (4.4), u *is an optimal solution to*

the variational problem. If F is strictly convex in λ and ξ for each $x \in \Omega$, and u satisfies Equations (4.3) *and* (4.4)*, it is the unique optimal solution to the variational problem.*

Proof. We first prove the result for the one dimensional case. Suppose for any $x \in (a, b)$ F is convex in λ and ξ and u satisfies (4.3). Then convexity means

$$\begin{aligned} F(x, \tilde{\lambda}, \tilde{\xi}) &\geq f(x, \lambda, \xi) + D_{\lambda,\xi} F(x, \lambda, \xi) \begin{pmatrix} \tilde{\lambda} - \lambda \\ \tilde{\xi} - \xi \end{pmatrix} \\ &= F(x, \lambda, \xi) + F_\lambda(x, \lambda, \xi)(\tilde{\lambda} - \lambda) + F_\xi(x, \lambda, \xi)(\tilde{\xi} - \xi). \end{aligned}$$

Thus, for any v admissible,

$$\begin{aligned} F(x, v(x), v'(x)) \geq F(x, u(x), u'(x)) &+ F_\lambda(x, u(x), u'(x))(v(x) - u(x)) \\ &+ F_\xi(x, u(x), u'(x))(v'(x) - u'(x)). \end{aligned}$$

This means

$$\begin{aligned} J(v) - J(u) &= \int_a^b F(x, v(x), v'(x)) - F(x, u(x), u'(x)) dx \\ &\geq \int_a^b \Big[F_\lambda(x, u(x), u'(x))(v(x) - u(x)) \\ &\quad + F_\xi(x, u(x), u'(x))(v'(x) - u'(x)) \Big] dx. \end{aligned}$$

Integration by parts on the second term gives $v(x) - u(x)$ times the Euler-Lagrange equation; so we have $J(v) \geq J(u)$ and so u is optimal. Strict convexity would imply by the same argument that $\geq$ can be replaced by $>$.

The general case follows using a similar argument, where we, as in the proof of Theorem 4.2.1, replace integration by parts with the Divergence Theorem. □

The following result gives a slightly stronger statement; we will not prove it.

Theorem 5.1.6. *If u is the unique solution to Equation* (4.3) *and for any fixed $x \in \Omega$ and $\lambda \in \mathbb{R}$, F is a convex function with respect to ξ, then u is an optimal solution of the variational problem.*

5.2 Convexity with non-differentiable data

For the following, we will focus on problems involving minimizing $f(x)$ over $x \in \mathbb{R}^n$; constraints will be introduced when needed. Let's consider the following (easy) minimization problem:

$$\min_{x \in \mathbb{R}^n} f(x) = \sum_{i=1}^{n} |x_i|.$$

Clearly, the global optimizer is $x_i = 0$. However, none of the theory in chapter 2 is applicable to this (perhaps) reasonable problem: at the minimizer (among other points), f is nondifferentiable! Convex analysis (and the more general topic of nonsmooth analysis) is the study of, in part, optimization problems where the data—objective and/or constraint functions—is nondifferentiable. Our goal here is to introduce and establish a few basic properties of the primary tool for handling such problems: the subgradient.

Recall that a *differentiable* function g is convex if and only if, at every point, the tangent line/plane/hyperplane lies below (or along) the graph of g; that is, the tangent hyperplane (for the higher dimnsional case) is *majorized* by the graph of g. An alternative way this is phrased in the literature is that the tangent hyperplane is a *supporting* hyperplane. This is the geometric intuition of the first part of Proposition 5.1.1. Let's look a little more at what this means geometrically

in one dimension. At every point $x \neq 0$, $f(x) = |x|$ is differentiable with derivative $Df(x) = x/|x|$ (we phrase it this way as it will be the same for higher dimensions). That means that for each $x < 0$, the line orthogonal to $(-1, -1)$ and going through the point $(x, |x|)$ is tangent to the graph of f and lies below it everywhere. Similarly, for each $x > 0$ the line orthogonal to $(1, -1)$ and going through the point $(x, |x|)$ is tangent to the graph of f and supports f. Moreover, such a line is the *only* tangent line; wherever f is differentiable, we have a unique supporting tangent line (assuming f is convex also.)

However, what about at $x = 0$? Now, clearly, the line orthogonal to $(1, -1)$ and going through $(0, 0)$ is a supporting tangent line. So is the line orthogonal to $(-1, -1)$. But more than just those two satisfy this condition: lines orthogonal to, say: $(0, -1)$, $(0.5, -1)$, $(-0.5, -1)$, $(-0.25, -1)$, $(0.25, -1)$ all would be supporting tangent planes. In fact, this would be true for any line orthogonal to $(\xi, -1)$ going throuhgh $(0, 0)$ for $\xi \in [-1, 1]$. This is illustrated in Figure 5.1.

Let's use the connection between Df (when/where it exists) and supporting tangent lines/planes/hyperplanes to handle cases where Df doesn't exist everywhere.

Definition 5.2.1. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function. A vector ξ is a* subgradient *of f at a point x if*

$$f(z) \geq f(x) + \xi^T(z - x), \quad \forall z.$$

That is, it is a vector defining a nonvertical suporting hyperplane of f at the point $(x, f(x))$.

The set of all subgradients of f at x is called the subdifferential of f at x *and is denoted by $\partial f(x)$. The set-valued mapping $\partial f : x \to \partial f(x)$ is called the* subdifferential of f.

If at x, the set $\partial f(x)$ is non-empty, f is said to be subdifferentiable *at x.*

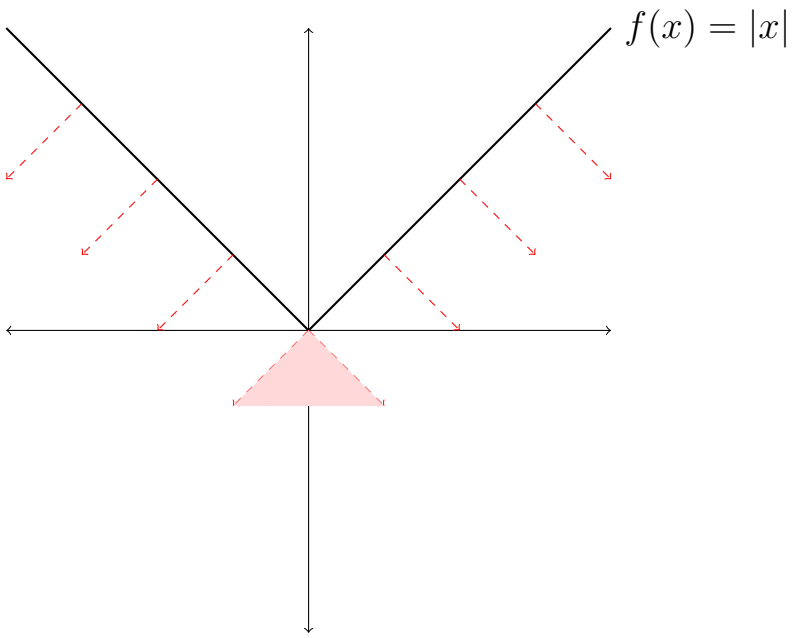

Figure 5.1: Directions orthogonal to supporting tangent lines for $f(x) = |x|$ are in dashed, red.

In other words, the subdifferential of f at x is a set containing all directions which can generate a supporting tangent hyperplane (we will use "hyperplane" rather than continuing to also mention lines and planes).

First, we will make a few comments: the set $\partial f(x)$ is closed and convex; it may be empty, consist of a single vector, or more than one vector.

Example 5.2.2. • *Let $f(x) = \|x\|_2$ be the standard Euclidean norm. It is differentiable at every $x \neq 0$ and $\partial f(x)$ consists of the set $\{x/\|x\|_2\}$ (this can be readily checked by a result to follow). But at $x = 0$ we will have the subgradients are those vectors satisfying*

$$\|z\| \geq \langle \xi, z \rangle$$

for every z. That means the subdifferential at $x = 0$ is the Euclidean unit ball.

- *Let $f(x) = \max_{1 \le j \le n} |x_j$ be the Tchebycheff norm; then if e_j is the jth row of the $n \times n$ identity matrix,*

$$\partial f(0) = \{\sum_{j=1}^{n} a_j e_j - b_j e_j : \sum_{j=1}^{n} a_j + b_j = 1, a_j \ge 0, b_j \ge 0\} \quad (5.1)$$

- *Let*

$$f(x) = \begin{cases} -(1 - \|x\|^2)^{1/2}, & \|x\| \le 1 \\ +\infty & else \end{cases}.$$

Then for x such that $\|x\| < 1$, f is subdifferentiable but not for $\|x\|x \ge 1$, it is not.

- *Let C be a nonempty convex set and $I_C(x)$ be the* indicator *function :*

$$I_C(x) = \begin{cases} 0 & x \in C \\ +\infty & x \notin C. \end{cases}$$

Then $\xi \in I_C(x)$ means that ξ points "away" from every $z - x$ for $z \in C$. So ξ is normal *to C at x.*

A natural question we might first ask is how the *set* $\partial f(x)$ is related to the derivative of f:

Theorem 5.2.3 (Rockafellar)**.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function and x a point such that $f(x) < +\infty$. Then if f is differentiable at x, then $\nabla f(x)$ is the unique subgradient of f. Conversely, if f has a unique subgradient at x, then f is differentiable at x.*

In other words, the subdifferential always coincides with the set containing just the gradient of f when ∇f exists. Moreover, if f is

finite everywhere, we can construct the subdifferential at a point where f is nondifferentiable; the following result can be extended to the cases where f takes nonfinite values or f is not continuous, but the process is a bit more complicated.

Theorem 5.2.4. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuous and convex.Then*

$$\partial f(x) = \{\sum_{\ell \in \Lambda} c_\ell v_\ell : \sum_{\ell \in \Lambda} c_\ell = 1, c_\ell > 0, |\Lambda| < \infty, v_\ell = \lim_{i \to \infty} \nabla f(x_i), x_i \to x\}.$$

This means that we can construct $\partial f(x)$ by:

1. Finding all possible sequences $\{x_i\}$ of points converging to x where $\nabla f(x_i)$ exists;
2. Taking the limits of all resulting $\nabla f(x_i)$ to form a set $S(x)$;
3. Finding the smallest closed convex set containing $S(x)$ by taking all possible weighted averages of the elements of $S(x)$;
4. the resulting closed convex set is $\partial f(x)$.

In our example of $f(x) = |x|$, note that we can find $\partial f(0)$ in this way. Every sequence $x_i > 0$ with $x_i \to 0$ has $\nabla f(x_i) = 1$ and every sequence with $x_i \to 0$ with $x_i < 0$ has $\nabla f(x_i) = -1$. Any other sequence will have (if it converges) $\nabla f(x_i) = \pm 1$. So $S(0) = \{-1, 1\}$ in this example. Taking every weighted average possible of $S(0)$ gives us $\partial f(0) = [-1, 1]$, which coincides with the geometric notions from Figure 5.1.

The third step involves taking a set of vectors, S and making the smallest closed convext set containing S; this procedure of taking all possible weighted sums of the elements of S to generate this convex set is called taking the convex hull of S. We denote for brevity the convex hull of S as:

$$\overline{\text{co}} S := \{\sum_{\ell \in \Lambda} c_\ell : c_\ell \geq 0, \sum_{\ell \in \Lambda} c_\ell = 1, \|\Lambda\| < \infty\}.$$

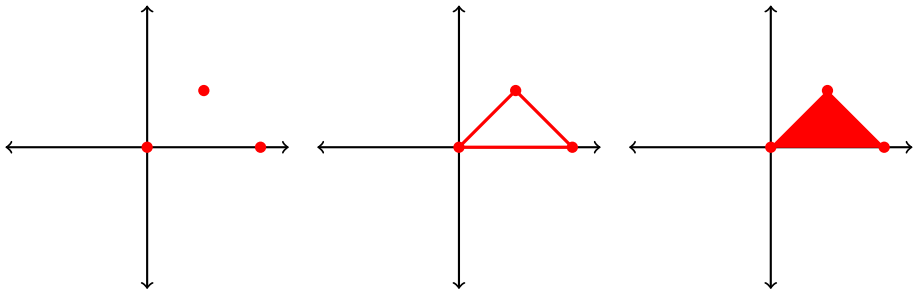

Figure 5.2: Forming the convex hull of a set of three points

Again, we can read this as taking every possible finite collection of vectors in S and taking every possible weighted average of that collection. As an example, consider $S = \{0, 3\} \subset \mathbb{R}$. The smallest closed convex set containing S is the interval $[0, 3]$. We can construct this by taking every finite collection of elements of S and drawing the line segment between them. In one dimension, that generates the interval. Let's consider the set $S = \{(0, 0), (1, 1), (2, 0)\}$. First, let's take every pair of points in S and include the line segments connecting each pair. Then, we need to take any pair of points on two separate line segments and make sure we include the line segment connecting them. We then can note that every point in the resulting set (which includes all of the line segments above) is convex, contains S, and can be re-interpreted as all points formed by taking weighted averages of the original points (which now form the corners of the polygonal region). This is shown in Figure 5.2 with the final $\overline{\text{co}}S$ in red on the right.

Example 5.2.5. *Let's use this to compute the subdifferential of the function*

$$f(x) = \begin{cases} .5x & x \in (-\infty, 1] \\ x - .5 & x \in (1, 2] \\ 3x - 3.5 & x \in (2, +\infty) \end{cases}$$

which is not differentiable everywhere. We know that on $(-\infty, 1)$ f

is differentiable and with derivative identically .5. *And on* $(1,2]$ f *has derivative equal to* 1. *So any sequence of points* $x_i \to x$ *such that* $\nabla f(x_i)$ *converges, must have* $\nabla f(x_i) \to .5$ *or* $\nabla f(x_i) \to 1$. *That means at* $x = 1$, *where* f *is nondifferentiable,* $\partial f(x)$ *contains* .5 *and* 1. *From this, we can construct our subdifferential by taking the hull around these points. That means* $\partial f(1) = [.5, 1]$. *Similarly, we can find* $\partial f(2)$ *and so*

$$\partial f(x) = \begin{cases} \{.5\} & x \in (-\infty, 1) \\ [.5, 1] & x = 1 \\ \{1\} & x \in (1, 2) \\ [1, 3] & x = 2 \\ \{3\} & x \in (3, +\infty) \end{cases}.$$

Note that the subdifferential whenever f *is differentiable is the* set *containing a single vector and* not *the vector itself.*

Now that we have an idea on how to construct $\partial f(x)$, we will ask what sort of calculus rules does the subdifferential satisfy. Since subgradients and gradients coincide where the latter exist, it would be nice if the subdifferential satisfies something like a product rule or chain rule. In fact, a general form of this does hold. We will state without proof a simplified version of these rules; it is relatively straightforward to verify that every operation on convex functions such as taking sums and (certain) compositions is also convex. The assumption that all functions are finite valued and smoothness hypotheses are not necessary for all of the statements, but it simplifies the presentation.

Theorem 5.2.6. *Let* $f_i : \mathbb{R}^n \to \mathbb{R}$ *be convex and finite valued.*

- *If* $s > 0$ *is a scalar,*

$$\partial(sf_i)(x) = s\partial f_i(x) := \{sv : v \in \partial f_i(x)\}.$$

- $$\partial(\sum_i f_i)(x) \subseteq \sum_i \partial f_i(x) := \{\sum v_i : v_i \partial f_i(x)\}.$$

  Equality holds if all but f_1 is C^2.

- *Suppose $f_2 : \mathbb{R} \to \mathbb{R}$ is nondecreasing. Then we have the chain rule*

  $$\partial f_2(f_1(x)) \subseteq \overline{co}\{\alpha\xi : \alpha \in \partial[f_2](f_1(x)), \xi \in \partial f_1(x)\}$$

  Note we take α in this statement is a subgradient of f_2 at $f_1(x)$. Equality holds if f_1 is continuously differentiable.

- *Suppose $A : \mathbb{R}^m \to \mathbb{R}^n$ is linear. Then $g = f_1(Ax)$ is convex and*

  $$\partial g(x) = A^T \partial f_1(Ax) := \{A^T \xi : \xi \in \partial f_1(Ax)\}.$$

- *Let $f = \sup_i\{f_i(x) : i \in \mathcal{N}\}$. Then f is convex and*

  $$\partial f(x) = \overline{co}\{\xi : \xi \in \partial f_i(x) \text{ and } f_i(x) = f(x)\}.$$

  That is, the right hand side is the set containing all subgradients of the functions f_i which give the largest value at x.

Finally, the (probably) most critically useful and easiest proven result about the subdifferential is:

Theorem 5.2.7. *Suppose $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ is convex for some convex Ω. Then x^* is a minimizer of f if and only if $0 \in \partial f(x)$.*

Proof. Since x^* is a minimizer, $f(y) \geq f(x^*)$ for all $y \in \Omega$. But this means $f(y) \geq f(x^*) + [0, \ldots, 0]^T(y - x^*)$ which means $0 \in \partial f(x^*)$. The converse statements also hold. □

These calculus rules and this last result mean we can now solve problems such as minimizing $(x-10)^4 + 15f(x)$ where $f(x)$ is the piecewise constant from the above example. All we need to do is find an x where $0 \in \partial\big[(x-10)^4 + 15f(x)\big]$. But that means we need to find an x where $0 \in 4(x-10)^3 + 15\partial f(x)$. This results in simply checking up to 5 inequalities or equalities (one for each case in $\partial f(x)$).

Bibliography

[1] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.

[2] Steve McDowall, *An Introduction to non-linear optimization*, Lecture Notes, 2018.

[3] Michael Herty, *Constrained Optimization in Finite and Infinite Dimensional Spaces*, Lecture Notes, 2012.

[4] Michael Herty and Sonja Steffensen, *Numerical optimization:Numerische Verfahren der nichtlinearen Optimerung*, Lecture Notes, 2012.

[5] Michael Hinze, Rene Pinnau, Michael Ulbrich, and Stefan Ulbrich, *Optimization with PDE Constraints*, Springer, 2009.

[6] Francis H Clarke, *Optimization and nonsmooth analysis*, SIAM, 1990.

[7] R. Tyrrell Rockafellar, *Convex analysis*, Princeton University Press, 1988.