Convex Optimization and Analysis

Jeffrey Liu

A quick warning...

These notes are incomplete and subject to mass reorganization and editing. However chapters 4-6 are fairly readable in its current state, and they are the most important. When I'm done chapter 7 will probably be three times the length of the other chapters, but by the time I finish it will be the most interesting.

Also, all credit to Fabrizio Conti for the brilliant cover photo, retrieved from Unsplash. Hope it helps the reader visualize descent algorithms :).

Foreword

Hi, Jeffrey here. Today is St Patrick's Day, 2020, but this year I won't be drinking my problems away on Ezra. I'm not sure when in the future that you, if anyone, will be reading this, so I'll forgive you if you don't remember that this was the year of the COVID-19 pandemic. My classes were suspended last Friday, and I've spent the better part of the break so far playing video games and watching YouTube[1].

To keep my sanity over the next couple months, I've also decided to start a collection of notes for some of my more favoured subjects.

Convexity is a beautiful property that lends itself to many powerful results in algebra, analysis, and geometry. Applications have been found in many branches of pure mathematics, engineering, finance, computer science, physics, and of course optimization. I'll be basing these notes off of the extensive literature in the subject, in particular: Boyd and Vandenberghe: *Convex Optimization*; and Wolkowicz's CO 463 course notes[2], among others.

In contrast to these sources, however, I'll be exploring the subject from the perspective of a (not exceptionally bright :p) undergraduate, so I'll try my best to motivate. I'll also look to introduce non-convex optimization, especially towards numerical methods and their applications in machine learning.

I hope that these notes will form a brief yet extensive overview of the subject, perhaps as a companion for a first graduate-level course in convex or nonlinear optimization. Of course, I'm still a student (read: **I'm not an expert**), so expect room for improvement. A quick note: some elementary linear algebra and calculus is expected, although I'll try to have an appendix with some prerequisite theorems.

Enjoy[3]!

[1] If my parents or any future professors or employers ever read this, I was also reading lots of textbooks.

[2] I'll be following the course notes the closest, but with my own commentary and editorials.

[3] I'd like to thank Fabrizio Conti again for the brilliant cover photo, retrieved from Unsplash. Also, special thanks to my CO 463 Professor Henry Wolkowicz, as well as my mom and dad for their continued belief in me.

Contents

Notation

I try to use standard notation; anything that may be controversial will be listed here. If you are unfamiliar with any of these definitions, please read the appendix.

Sets and Vector Spaces

$\mathbb{N}$	The set of non-negative integers $\{0, 1, 2, ...\}$
$[n]$	The set of integers $\{1, 2, ..., n\}$
$\mathbb{Z}, \mathbb{Z}_+, \mathbb{Z}_{++}$	The set of integers, resp., the non-negative and positive integers
$\mathbb{R}, \mathbb{R}_+, \mathbb{R}_{++}$	The set of real numbers, resp., the non-negative and positive reals
$\mathbb{R}^n$	The vector space of column vectors with n real entries, together with the standard inner product
$\mathbb{E}, \mathbb{E}^n$	Any Euclidean vector space[4](finite dimensional real inner product space), resp., of dimension n if specified
$\mathbb{M}^n$	The vector space of $n \times n$ square matrices with real entries, together with the Frobenius inner product ($\langle A, B \rangle = \operatorname{tr}(A^\top B)$)
$\mathbb{S}^n_+, \mathbb{S}^n_{++}$	The subset of $\mathbb{M}^n$ consisting of positive semidefinite (resp. positive definite) matrices.
ETC TODO	

Vectors and Matrices

$x^\top$	The transpose of x
I, I_n	The identity matrix (resp. of dimension n if specified)
x_k	The kth entry of x
$x^{(k)}$	The kth element of a sequence of vectors $\{x^{(k)}\}_{k \geq 0}$.
$\operatorname{diag}(x)$, X	The square matrix with x along the main diagonal and zeros everywhere else
$\|x\|_p$	The ℓ_p-norm of x
$\|x\|$	The induced norm of x (square root of inner product)
e	The vector of all 1's with dimension implied by context
e_i	The ith vector in the standard basis of $\mathbb{R}^n$

Abuse of Notation

$\min_{x \in \Omega} f(x)$	The *infimum* of $f(x)$ as x ranges in Ω
$\max_{x \in \Omega} f(x)$	The *supremum* of $f(x)$ as x ranges in Ω

[4] An astute reader may note that all n dimensional Euclidean vector spaces are isomorphic to $\mathbb{R}^n$, and may ask why we bother distinguishing them. Honestly I think the only reason is so that we can be lazy and always assume $\mathbb{R}^n$ refers to that specific vector space with the standard inner product.

Introduction

Note: I'll put some pictures here eventually.

Historically, mathematicians have preferred structural results over numerical ones. Greats such as Euler, Gauss, Riemann, etc., have been devoted to the creation of beautiful theories in algebra, analysis, geometry, and number theory. It wasn't until the 20th century and the invention and popularization of the computer did the study of algorithms and computation take off.

Combinatorics and Optimization are two fields which blossomed in this new age. Many computational subfields of combinatorics, including those of graph theory, matroid theory, and polyhedral theory, have only been developed recently. The simplex method, published by Dantzig in 1947, was the first of many algorithms with promising practicality, and with it came the field of operations research.

Convexity, as a geometric property, has been known since antiquity. Properties have been investigated by the likes of Euler and Cauchy, however its true potential wasn't realized until the late 1800's when German mathematician Minkowski was able to apply it to number theory. He and fellow German Brunn developed much of the theory in two and three dimensions. Carathèodory, Krein, Milman, and Fenchel, among many others, developed and generalized much of the theory from the turn of the century until about the second world war. By 1970 or so all of the convexity theory we will require has been discovered.

The simplex method was a huge landmark in optimization with remarkable practical efficiency, however no polynomial time variation has ever been found. Collaborations to find a provably polynomial time algorithm lead to the discovery of the ellipsoid method in the 70's. In 1984, Indian Karmarkar proposed the first poly-time interior-point method for linear programming while working for Bell Labs, and non-linear adaptations have been an active field of research since the late 1980's.

Since the 90's, applications have been found in the traditional domains of operations research, and also in engineering (robotics, signal processing, circuit design, . . .); computer science (machine learning), the physical sciences, and finance.

We'll first discuss convex geometry and the natural extension to functions. With this, we'll be able to analyze convex programs and develop optimality constraints. These constraints, when violated, give arise to algorithms. Finally, we'll conclude with as many interesting applications as I could find.

For better flow, I will not prove many of the propositions and theorems unless they are particularly insightful. My primary intention is for these notes to be a reference to myself, but I'll try my best to be a good teacher (plus I enjoy teaching and hope to do it more often). I'll have an appendix on hints and solutions for some the more tricky proofs.

Oh, one last thing: I'll try to maintain a prerequisite DAG. So please don't be discouraged like I was and think that you have to read two chapters of geometry before you even see an optimization problem. In fact, you may be able to understand many of the algorithms by simply reading their description and googling any definitions you haven't heard of before.

But still, it's really important to know the fundamentals. I'm really sorry that sections 1 and 2 may get a little boring, but we need to learn to walk before we can run.

1 Convex Geometry

We shall explore the main geometric structures which arise in convex optimization, beginning with, of course, sets.

1.1 Convex Sets

1.1.1 Definitions

Although some notion of convexity has existed geometrically since at least Archimedes, the modern definition of a convex set is actually an algebraic one:

Definition 1.1. A subset $C \subseteq \mathbb{E}$ is **convex** if

$$x, y \in C, \lambda \in [0,1], \qquad \Longrightarrow \qquad (1-\lambda)x + \lambda y \in C. \tag{1.1}$$

We can see that, for a fixed x and y, the quantity $(1-\lambda)x+\lambda y$ represents the closed line segment between x and y as λ ranges over the unit interval. Then Equation 1.1 is equivalent to the statement that the set S contains all of its line segments. There are a couple alternate equivalent formulations of convexity which may (or may not) help you visualize the definition of convexity:

Proposition 1.2. Let $C \subseteq \mathbb{E}$. Then the following are equivalent:

(i) C is convex,

(ii) C contains all of its convex combinations, that is, if $x^{(1)}, \ldots, x^{(k)} \in C$ and $\lambda_1, \ldots, \lambda_k \in \mathbb{R}_+$ with $\sum_{i=1}^k \lambda_i = 1$, then the **convex combination**[5]

$$\sum_{i=1}^{k} \lambda_i x^{(i)} \tag{1.2}$$

belongs to C as well.

(iii) The intersection of C with any line is either the empty set or a connected interval.

Convexity is an extraordinarily simple condition; many everyday sets can easily be shown to be convex. We'll list a few examples:

Example 1.3. The empty set is convex (vacuously).

Example 1.4. All subspaces S of $\mathbb{E}$ are convex. In addition, translations of subspaces ($S + \overline{x}$ for some $\overline{x} \in \mathbb{E}$) are convex. These translations are called affine sets (more on this later!).

Example 1.5. All polyhedra, sets of the form (where $A \in \mathbb{R}^{n \times m}, b \in \mathbb{R}^m$)

$$P = \{x \in \mathbb{E} \,:\, Ax \leq b\}\,, \tag{1.3}$$

are convex. In particular if $m = 1$ then we conclude the closed halfspace

$$H = \{x \in \mathbb{E} \,:\, \langle \phi, x \rangle \leq \alpha\} \tag{1.4}$$

is convex as well. The open halfspace is also convex, although it is not a polyhedra.

[5] This is also called the weighted mean (from physics), and a generalization of barycentric coordinates. You can also think of the λ_i as being a probability distribution.

1.1.2 Operations Preserving Convexity

There are several common operations which preserve convexity of a set. We'll be able to use these operations to build increasingly sophisticated sets from the basic examples given above. These propositions all follow from definitions, and the reader is encouraged to prove them on their own.

Proposition 1.6. Let $C_i \subseteq \mathbb{E}, i \in I$ be a collection of convex sets, where I is a (potentially uncountably large) index set. Then the intersection

$$\bigcap_{i \in I} C_i \tag{1.5}$$

is convex.

In particular, the *convex hull* of a set S, possibly defined as the intersection of all convex sets containing S, is convex. We'll talk more about this later.

Proposition 1.7. Let $C_1, C_2 \subseteq \mathbb{E}$ be two convex sets, and $\alpha, \beta \in \mathbb{R}_+$. Then the Minkowski sum, defined by

$$\alpha C_1 + \beta C_2 := \{\alpha x + \beta y \, : \, x \in C_1, y \in C_2\} \tag{1.6}$$

is a convex set.

Proposition 1.8. Let $C_i \subseteq \mathbb{E}^{n_i}$ for $i \in [m]$. Then the Cartesian product defined by

$$C_1 \times \cdots \times C_m = \{(x_1, \ldots, x_m) \in \mathbb{E}^{n_1 \cdot \cdots \cdot n_m} \, : \, x_i \in C_i, \forall i \in [m]\} \tag{1.7}$$

is a convex set. Conversely, if $C \subseteq \mathbb{E}^{n_1 \times \cdots \times n_m}$ is a convex set, then each projection

$$\left\{x_i \subseteq \mathbb{E}^{n_i} \, : \, (x_1, \ldots, x_m) \in \mathbb{E}^{n_1 \times \cdots \times n_m}\right\} \tag{1.8}$$

is a convex set as well, for $i \in [m]$.

Proposition 1.9. Let $\mathcal{A} : \mathbb{E}^n \to \mathbb{E}^m$ be an affine mapping. Then if $C \subseteq \mathbb{E}^n$ and $D \subseteq \mathbb{E}^m$ are convex sets, both the image $\mathcal{A}(C)$ and the pre-image $\mathcal{A}^{-1}(D)$ are convex sets.

Proposition 1.10. Let $C \subseteq \mathbb{E}$ be a convex set. Then the interior $\operatorname{int} C$ and closure $\operatorname{cl} C$ are convex sets as well.

Actually, we shall see that we can strengthen Proposition 1.10 when we define the *relative interior*, in case C has a "lower dimension" than $\mathbb{E}$. But this will come later.

1.1.3 Convex Hulls and Carathéodory's Theorem

Many problems encountered in nature will **not** be convex (and usually will be difficult to solve), so we'll formulate strategies to relax these problems so that they are convex. The simplest of which is the convex hull.

Definition 1.11. Let $S \subseteq \mathbb{E}$ be *any* set. Then the **convex hull** of S, denoted $\operatorname{conv} S$, is the *smallest convex set containing* S. That is, if C is any other convex set containing S, then $\operatorname{conv} S \subseteq C$.

It's easy to show that the convex hull of S is the intersection of all convex sets containing S. In fact, this also serves as a decent definition of the convex hull, as by Proposition 1.6 we know that this intersection is convex. Some texts prefer the following equivalent formulation of convex hull:

Proposition 1.12. Let $S \subseteq \mathbb{E}$. Then

$$\operatorname{conv} S = \left\{ \sum_{i=1}^{k} \lambda_i x^{(i)} \,:\, k \in \mathbb{Z}_{++}, \sum_{i=1}^{k} \lambda_i = 1, \lambda \in \mathbb{R}^k_+, x^{(i)} \in S \right\}. \tag{1.9}$$

That is, the convex hull of S is precisely the set of all convex combinations of points from S. Now, despite the simplicity of the definition of convexity, we can build a very rich theory to describe these objects. We can immediately state a beautiful result of the subject.

Theorem 1.13 (Carathéodory, 1911)**.** Let $S \subseteq \mathbb{E}^n$. Then the convex hull of S is

$$\left\{ \sum_{i=1}^{\mathbf{n+1}} \lambda_i x^{(i)} \,:\, \sum_{i=1}^{n+1} \lambda_i = 1, \lambda \in \mathbb{R}^{n+1}_+, x^{(i)} \in S \right\}. \tag{1.10}$$

Proof. We'll show that Carathéodory's Theorem follows from linear independence[6]. Insert proof here. □

In other words, **no matter how misbehaved** S is as a set, any point in the convex hull of S can be written as a convex combination of just $n + 1$ points from S. This is remarkable, as it will often allow us to simplify arbitrarily many variables into just $n + 1$. In fact, if S is sufficiently well behaved, we get an even stronger statement.

Theorem 1.14 (Fenchel and Blunt, YYYY?)**.** Let $S \subseteq \mathbb{E}^n$, and suppose that S has no more than n connected components. Then only n points are needed in Theorem 1.13.

There are several related theorems that are worth knowing, although they don't appear too often[7].

[6] Although there is a very nice proof using linear programming

[7] Some of the proofs of these theorems have appeared as challenge problems on some of my exams, so beware.

Corollary 1.15 (A consequence of Carathéodory's Theorem)**.** The convex hull of a compact set is compact.

Exercise 1.16. Beware! The compactness condition cannot be weakened: the convex hull of a closed set may not be closed. Try to find a counterexample!

Theorem 1.17 (Helly, YYYY)**.** Let $C^{(i)} \subseteq \mathbb{E}^n, i \in I$ be a (potentially uncountable?) collection of compact convex sets. Then if every subcollection of $n+1$ sets have a non-empty intersection, then

$$\bigcap_{i \in I} C^{(i)} \neq \emptyset \tag{1.11}$$

Theorem 1.18 (Radon, YYYY)**.** Let $\{x^{(1)}, ..., x^{(n+2)}\} \subseteq \mathbb{E}^n$. Then there is a partition I_1, I_2 of the indices $[n+2]$ such that the convex hulls

$$C_1 = \operatorname{conv}\left\{x^{(i)} \,:\, i \in I_1\right\}, \qquad C_2 = \operatorname{conv}\left\{x^{(i)} \,:\, i \in I_2\right\} \tag{1.12}$$

intersect $C_1 \cap C_2 \neq \emptyset$.

Theorem 1.19 (Shapley-Folkman, YYYY)**.** GOES HERE

1.2 Affine sets

Earlier, we alluded to the notion of an *affine set*, as well as the *dimension* and *relative interior* of a set in $\mathbb{E}$. We make these definitions now.

If we do not restrict λ to the interval $[0,1]$ in the definition of convexity, then we get another definition:

Definition 1.20. A subset $S \subseteq \mathbb{E}$ is **affine** if

$$x, y \in S, \lambda \in \mathbb{R}, \quad \Longrightarrow \quad (1-\lambda)x + \lambda y \in S. \tag{1.13}$$

Intuitively, just as convex sets contain all of their convex combinations, we'd like an affine set to be one containing all of its affine combinations. However unlike their convex counterparts, affine sets can be characterized extremely simply.

Proposition 1.21 (Different formulations of affine sets)**.** Let $S \subseteq \mathbb{E}$. Then the following are equivalent:

(i) S is an affine set, in the sense of Definition 1.20.

(ii) S is a **linear manifold**, ie., there exists a linear transformation $\mathcal{A} : \mathbb{E} \to \mathbb{F}$ and vector $b \in \mathbb{F}$ so that

$$S = \{x \in \mathbb{E} \,:\, \mathcal{A}x = b\}. \tag{1.14}$$

(iii) There exists some $d \in \mathbb{E}$ and linear transformation $\mathcal{B} : \mathbb{F} \to \mathbb{E}$ so that

$$S = \{x \in \mathbb{E} \,:\, x = \mathcal{B}y + d, y \in \mathbb{F}\}. \tag{1.15}$$

This is sometimes called the **parametric form** or **nullspace representation** of S.

(iv) S is the translation of a subspace, ie., for $\overline{x} \in S$ the set $S - \overline{x}$ is a subspace of $\mathbb{E}$.

Among these formulations, (1.14) is the most useful, ie., affine sets are the solution sets to some system of linear equations, and the terms *affine set* and *linear manifold* will be used interchangeably.

However, (iv) inspires us to adapt linear independence to the affine case.

affinely independence definition goes here, and some basic theorems

These tools form the backbone of what's really happening in our proof of Theorem 1.13.

Dimension, Relative interior

1.3 Geometry with Convex Sets

1.3.1 Extreme Points and Faces

What is this used for? facial reduction and stuff... to be added...

Proposition 1.22. Let $C \subseteq \mathbb{E}$ be a convex set. Then $\operatorname{ext} C$ is non-empty if and only if C is pointed, ie., C does not contain any lines.

We conclude this section with the interior description of a convex set:

Theorem 1.23 (Minkowski)**.** Let $C \subseteq \mathbb{E}$ be a bounded convex set. Then

$$C = \operatorname{conv} \operatorname{ext} C. \tag{1.16}$$

In other words, the set of extreme points of C is the smallest set which is "good enough" to determine C via its convex hull; it is the "shortest worker's instruction for building the set".

1.3.2 Projections

Projections onto convex sets will be our main tools in proving hyperplane separation theorems. They are a natural extension of projections onto subspaces from linear algebra.

Recall that we can define an *orthogonal projection* P onto a subspace $V \subseteq \mathbb{E}$ as a linear transformation satisfying $P^2 = P^\top = P$. AND THEN w=v+v perp, etc

We'd like to define the projection onto a convex set similarly.

Definition 1.24. Let $C \subseteq \mathbb{E}$ be a non-empty closed convex set, and $x \in \mathbb{E}$. Then define the **projection** of x onto C , denoted as $P_C(x)$, as the *unique* solution to

$$P_C(x) = \arg\min_{y \in C}\{\|y - x\|\}. \tag{1.17}$$

It takes a little work to see why $P_C(x)$ always exists, and if it does, why it's unique.

Proposition 1.25 (Kolmogorov's Criterion, YYYY)**.** *Let $C \subseteq \mathbb{E}$ be a non-empty closed convex set, and $x \in \mathbb{E}$. Then $P_C(x)$, as defined in Definition 1.24, is well defined. In particular, $y_x = P_C(x)$ if and only if*

$$\langle x - y_x, y - y_x \rangle \leq 0, \forall y \in C. \tag{1.18}$$

Proof. ?? □

Moreau decomp

Here are a few nice exercises with projections.

Proposition 1.26. *Let $x, y \in \mathbb{E}$ and $C \subseteq \mathbb{E}$ be a non-empty closed convex set. Then*

$$\|P_C(x) - P_C(y)\| \leq \|x - y\| . \tag{1.19}$$

Insert some more...

1.3.3 Separation

Preliminary version:

Theorem 1.27 (Hyperplane Separation)**.** *Let $C \subseteq \mathbb{E}$ be a closed convex set, and suppose that $x \in \mathbb{E} \setminus C$. Then there exists a hyperplane separating x from C, i.e., there exists $\phi \in \mathbb{E}, \alpha \in \mathbb{R}$ such that*

$$\langle c, \phi \rangle \leq \alpha < \langle x, \phi \rangle, \tag{1.20}$$

for all $c \in C$.

We can think of hyperplane separation as another *Theorem of the Alternative*, in the sense that *either* x is contained inside C, or it isn't and we have a hyperplane certificate to prove it. Spoiler alert: This theorem will prove essential to developing strong duality for convex optimization.

We can immediately generalize Theorem 1.27 where we don't just separating a point from a convex set, but instead separate two convex sets from each other:

Corollary 1.28. *Let $C_1, C_2 \subseteq \mathbb{E}$ be two non-empty closed convex sets, where C_2 is compact and $C_1 \cap C_2 = \emptyset$. Then there exists a strictly separating hyperplane between C_1 and C_2, ie., there exists $\phi \in \mathbb{E}, \alpha \in \mathbb{R}$ such that*

$$\langle \phi, c_1 \rangle < \alpha < \langle \phi, c_2 \rangle \tag{1.21}$$

for all $c_1 \in C_1, c_2 \in C_2$.

other stuff outer description

1.4 Cones

1.4.1 Definitions

The cone is another very important geometric object, although as with the convex set, we define it algebraically.

Definition 1.29. A subset $K \subseteq \mathbb{E}$ is called a **cone** if

$$x \in K, \lambda \in \mathbb{R}_+ \quad \implies \quad \lambda x \in K. \tag{1.22}$$

Oftentimes, we may find ourselves focusing on the closed convex cones (c.c.c. for short), which are closer our primary school intuition of a "pylon-shaped" cone. As we shall see, these specific cones have many nice properties, however it is important to note that these c.c.c.s are not the only types of cones. Convex cones have a nice characterization:

Proposition 1.30. Let $K \subseteq \mathbb{E}$ be a cone. Then K is a convex cone if and only if

$$x, y \in K \quad \implies \quad x + y \in K. \tag{1.23}$$

A convex cone contains all of its conic combinations, which are sums of the form

$$\sum_{i=1}^{k} \lambda_i x^{(i)} \quad \text{for } \lambda \in \mathbb{R}^k_+ \text{ and } x^{(i)} \in K. \tag{1.24}$$

Just as with convex sets, the intersection of any family of convex cones is itself a convex cone. Similar to the convex hull, we can define a conical relaxation of a set, or the conic hull:

Definition 1.31. Let $S \subseteq \mathbb{E}$ be any set. Then the **conical hull** of S, denoted $\operatorname{cone} S$, is the smallest *convex* cone containing S.

As with before, we have an elegant theorem for conical hulls:

Theorem 1.32 (Carathéodory, 1911)**.** Let $S \subseteq \mathbb{E}^n$. Then the conic hull of S is

$$\left\{ \sum_{i=1}^{\mathbf{n}} \lambda_i x^{(i)} \,:\, \sum_{i=1}^{n} \lambda_i = 1, \lambda \in \mathbb{R}^n_+, x^{(i)} \in S \right\}. \tag{1.25}$$

The reason why we care about closed convex cones so much (besides the fact that they are just *so cool*) is because they correspond with **partial orders** on $\mathbb{E}$. Understanding cone geometry leads to conic optimization (duh), most notably including second-order cone programming and semidefinite programming. We have efficient algorithms for both these problems.

1.4.2 Partial Orders

Many things cannot be compared to each other[8]. In linear algebra, there really isn't a good way to compare two arbitrary vectors. For example, in $\mathbb{R}^2$, the vectors $(0,1)$ and $(1,0)$ are indistinguishable (especially before the choice of a basis). This appears to be a huge problem in optimization, where questions of the form "minimize …", inherently requires us to be able to compare stuff to each other.

However, this is a non-issue, as we introduce the concept of a *partial order.*

PARTIAL ORDER DEFINITION

Unlike *total orders*, we are allowed to have pairs of objects a, b which are **incomparable**, meaning neither $a \preceq b$ or $b \preceq a$.

Example 1.33. We are already familiar with a partial order— the *less than or equal to* relation $\leq$ over the real numbers. We can generalize this to obtain a natural partial order in $\mathbb{R}^n$, where

$$x \preceq y \quad \iff \quad x_i \leq y_i, \forall i \in [n]. \tag{1.26}$$

Essentially we say $x \preceq y$ if every entry of x is less than the corresponding entry of y. Usually we won't be too picky with the notation and just write $x \leq y$. After messing around with the definition for a bit, a cool alternate formulation is

$$x \leq y \quad \iff \quad y - x \in \mathbb{R}_+. \tag{1.27}$$

As it turns out, this is actually a more natural definition for the $\leq$ relation[9]. In particular, it is (at least on the surface) coordinate free, and it extends easily.

Proposition 1.34. Indeed, if $K \subset \mathbb{E}$ is any pointed convex cone, then the relation $x \preceq y$ if and only if $y - x \in K$ is a partial order. Conversely, if $\preceq$ is any partial order then

$$\{x \in \mathbb{E} \,:\, 0 \preceq x\} \tag{1.28}$$

is a pointed convex cone.

EXAMPLE: SEMI DEFINITE CONE
RECESSION/ASYMPTOTIC CONE ¡- move this smwhere?
TANGENT CONE ¡- and this
moreau decompositions

1.4.3 The Dual Cone

Blah blah

Definition 1.35. Let $S \subseteq \mathbb{E}$. Then we can define the **positive polar cone** of S, denoted as S^+, as

$$S^+ := \{\phi \in \mathbb{E} \,:\, \langle x, \phi \rangle \geq 0, \forall x \in S\}. \tag{1.29}$$

[8] Apples and oranges is a canonical example

[9] and it may be familiar if you've done anything with equivalence relations

We can define the **negative polar cone** S° similarly, and $S^\circ = -S^+$. Note that the positive polar cone is sometimes referred to as the *dual cone*, denoted as S^*, and the negative polar cone is simply called the *polar cone.*

Insert graphic representing dual cone

The dual cone comes up surprisingly often, as it represents I DON'T KNOW... FIGURE THIS OUT. One fact that comes up a lot is the following:

Proposition 1.36. Let $K \subseteq \mathbb{E}$. Then K is a closed convex cone if and only if

$$K = (K^+)^+. \tag{1.30}$$

Proposition 1.36 can be used to provide a geometric interpretation/proof of the famous Farkas' Lemma from linear programming:

Corollary 1.37 (Farkas' Lemma)**.** Let $A \in \mathbb{R}^{n\times m}, b \in \mathbb{R}^n$. Then exactly one of the following is true:

(i) The system $Ax = b, x \geq 0$ has a solution

(ii) The system $A^\top y \geq 0, b^\top y < 0$ has a solution

Proof. Hi □

Farkas' lemma and other such *theorems of the alternative* are the foundation of a beautiful duality theory in linear programming. It is strongly recommended that the reader be familiar with these ideas.

2 Convex Functions

2.1 Preliminary Definitions

We've seen many examples of the importance and elegance of convex sets. As we shall see, there is a very natural correspondence between convex sets and convex functions, which will allow us to transfer much of the theory over. In particular, we will be able to derive several properties which are crucial to the success of convex optimization.

The general definition of a convex function is usually introduced in freshman calculus, and is defined by Jensen's Inequality.

Definition 2.1. Let $f : C \to \mathbb{R}$. Then f is a **convex function** if C is a convex set and

$$x, y \in C, \lambda \in [0,1] \quad \implies \quad f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y). \tag{2.1}$$

We see that it really is necessary for C to be a convex set, as otherwise the LHS of 2.1 may not be defined. Pictorially, we define f to be convex if it lies below all of its secant lines.

MAYBE INSERT PICTURES AND EXAMPLES

If we have a function f such that $-f$ is convex, then we call f **concave**. Concave functions satisfy the inequality

$$x, y \in C, \lambda \in [0,1] \quad \implies \quad f((1-\lambda)x + \lambda y) \geq (1-\lambda)f(x) + \lambda f(y). \tag{2.2}$$

We can immediately uncover the relationship between convex sets and convex functions. Recall the following definition:

Definition 2.2. Let $f : S \subseteq \mathbb{E} \to \mathbb{R}$ be any function. We define the **epigraph** of f, denoted $\operatorname{epi} f$, by

$$\operatorname{epi} f = \{(x, r) \in S \times \mathbb{R} \,:\, f(x) \leq r\}. \tag{2.3}$$

PICTURE? The epigraph represents the region "above" the graph of f, and is a subset of a Euclidean space of one dimension higher. Then we have the following equivalence:

Proposition 2.3. Let $f : S \to \mathbb{R}$. Then the following are equivalent:
(i) f is a convex function
(ii) $\operatorname{epi} f$ is a convex set

Examples: lines, quadratics (introduce hessian is psd iff convex), other examples, norms,

As an aside, we'd also like to note that we can recover the classical Jensen's inequality by induction.

Theorem 2.4 (Jensen's Inequality)**.** Let $f : \mathbb{E} \to \mathbb{R}$ be a convex function, $x^{(1)}, \ldots, x^{(k)} \in \mathbb{E}$ be points in the domain, and $\lambda \in \mathbb{R}^k$ satisfying $\sum_{i=1}^{k} \lambda_i = 1$ be weights. Then

$$f\left(\sum_{i=1}^{k} \lambda_i x^{(i)}\right) \leq \sum_{i=1}^{k} \lambda_i f(x^{(i)}). \tag{2.4}$$

We also need several definitions about functions in general. Depending on your calculus/analysis background, you may have seen many of these definitions before. Personally I knew none of these concepts before I took CO 255/463, and if you're in a similar situation I recommend that you spend some time to draw some examples and get a visual intuition.

Many of these definitions don't show up too often, and a surprising amount of it is unnecessary if you're simply looking to apply algorithms[10]. Nevertheless, they form important tools for us to develop much of the deeper theory in convex analysis.

Definition 2.5. Let $f : \mathbb{E} \to \mathbb{R}$ and $r \in \mathbb{R}$. Then we can define the rth **level set**, denoted $L_r(f)$, as

$$L_r(f) = \{x \in \mathbb{E} \,:\, f(x) = r\}. \tag{2.5}$$

Similarly, we can define the rth **sublevel set** (sometimes called lower level set), denoted as $S_r(f)$, as

$$S_r(f) = \{x \in \mathbb{E} \,:\, f(x) \leq r\}. \tag{2.6}$$

[10] It may even be tempting to skip this section and hope that it never comes up.

continuous + bounded sublevel sets? lower upper semicontinuous to understand what these mean geometrically, perhaps its helpful to picture in terms of epigraph closed function

2.1.1 As a vector space

REDO THIS A BIT Before we begin, we'd like to modify our definition of function to include a few objects which will simplify future propositions. In particular, we extend the real line by considering the addition of a new point which we call positive infinity[11]. This way, the infimum of any subset of the extended real line will be an extended real number.

Definition 2.6. For any function $\tilde{f} : C \subseteq \mathbb{E} \to \mathbb{R}$ we can define the **extended value function** $f : \mathbb{E} \to (-\infty, +\infty]$ by

$$f(x) = \begin{cases} \tilde{f}(x) & x \in C \\ +\infty & x \notin C. \end{cases} \tag{2.7}$$

For our purposes, we shall always assume that the domain C in Definition 2.6 is a convex set. We'll often want to recover this original domain from our extended function, so we'll adopt the following notation:

Definition 2.7. Let $f : \mathbb{E} \to (-\infty, +\infty]$ be an extended value function. Then the **domain** of f, denoted $\operatorname{dom} f$, is the set

$$\operatorname{dom} f := \{x \in \mathbb{E} \, : \, f(x) < \infty\} . \tag{2.8}$$

2.1.2 Elementary Properties

maxes are at extreme points, local mins are global mins, tangent line theorems, three slopes,

2.2 Other Types of Convexity

2.2.1 Quasiconvexity

2.2.2 Strong Convexity

2.2.3 Strict Convexity

2.2.4 K-Convexity

2.3 Calculus with Convex Functions

2.3.1 Derivatives

locally lipschitz, differentiable nearly everywhere

[11] I'll denote this extended real line as $(-\infty, +\infty]$, but other authors may use $\mathbb{R} \sqcup \{+\infty\}$ or other more concise notation. We'll try to avoid doing arithmetic with positive infinity; it's enough to just assume that $c + \infty, \lambda \cdot \infty$ are both equal to ∞ (for $c \in (-\infty, +\infty]$ and $\lambda \in (0, +\infty]$) and all other expressions are undefined.

2.3.2 Subdifferentials

some more I bet

3 Convex Programs

minimum vs minimal

3.1 The Framework

abstract convex program

kkt style convex program

3.2 Optimality Conditions

Picture this: you are at a bar on a Friday night, you're sitting with a few friends who're tryna vibe to crappy mumble rap, while you yourself are working very hard on a convex optimization problem. A hooded stranger[12] notices you, walks up and hands you a napkin with some numbers written on it. He claims that he has found an optimal solution for your problem. How could you quickly verify that he is correct?

Understanding optimality constraints is crucial to designing good algorithms. For one, it is important to know when to stop. Also, by analyzing when and why optimality constraints *fail* to hold, we can discover algorithms. As a stupid example, consider Fermat's Theorem:

Theorem 3.1 (Fermat, 1600's)**.** Let $f : A \to \mathbb{R}$ be some function, and suppose that x^* is a local extremum for f. If f is differentiable at x^*, then

$$\nabla f(x^*) = 0. \tag{3.1}$$

When condition 3.1 is violated, then we have a non-zero gradient and thus a direction for descent/ascent. As we'll see, this leads to gradient descent.

For convex functions we have even stronger conditions. The most important property is this very simple one:

Theorem 3.2 ("Unimodality")**.** Suppose that $f : \mathbb{E} \to \mathbb{R}$ is a convex function, and that $M \subseteq \mathbb{E}$ is a convex set. Then suppose that $x^* \in M \cap \operatorname{dom} f$ is a local minimizer on M, ie., there exists some $\delta > 0$ such that

$$x \in M \cap B(x^*; r) \quad \implies \quad f(x) \geq f(x^*). \tag{3.2}$$

Then x^* is actually a global minimizer of f on M, ie.,

$$x \in M \quad \implies \quad f(x) \geq f(x^*). \tag{3.3}$$

[12] Perhaps a UofT student.

Most algorithms are only capable of finding local extrema, but in convex optimization, this turns out to be equivalent to finding global extrema. We can use this theorem, as well as some facts about convex functions, to strengthen Fermat's theorem:

Theorem 3.3. A point $x^* \in \operatorname{dom} f$ is a global minimizer for f *if and only if* $0 \in \partial f(x^*)$.

Of course, if $x^* \in \operatorname{int}\operatorname{dom} f$ and f is differentiable at x^*, then $\partial f(x^*) = \{\nabla f(x^*)\}$ and we recover Fermat's original theorem. So what happens if $x^* \notin \operatorname{int}\operatorname{dom} f$?

Tangent cones, that one condition from homework? Rockafeller pshenichnyi

Weierstrauss Theorem

Lagrange Multipliers - introduce it math 247 style and say we'll talk more about it with duality

Maximizing convex functions

3.3 Duality

Duality, as a concept, is loosely defined as looking at an object in two ways. For example, we can analyze a signal with respect to either the frequency domain or the time domain. A compact convex set can be regarded by the union of a bunch of points (REFERENCE ABOVE), or the intersection of a bunch of halfspaces (REFERENCE ABOVE). Here, we'll define several different notions of duality to help us understand convex functions and convex programs.

3.3.1 Minimax Theorem

We begin by borrowing a theorem from game theory, which formalizes the first move disadvantage in many zero-sum games.

Proposition 3.4. For any $g : M \times N \to \mathbb{R}$,

$$\min_{x\in M}\max_{y\in N} g(x,y) \geq \max_{y\in N}\min_{x\in M} g(x,y). \tag{3.4}$$

Picture a zero-sum game played between two opponents X (Xavier) and Y (say, Yvette), where g is the profit function for Y. That is, if X plays the move x and Y plays the move y, then $g(x,y)$ is the (possibly negative) money paid out to Y from X. Player Y seeks to maximize her profit, while player X seeks to minimize his losses. There is a first player disadvantage in this game: the first player to commit is worse off, as the second player can adapt their strategy in response.

Example 3.5 (Rock Paper Scissors)**.** Alphonse and Beryl are playing a game of rock paper scissors, in which the loser pays the winner one Canadian Peso. From Beryl's perspective (Beryl takes the role of Y above), she has the following payoff matrix:

$G =$ (3.5)

B A	Rock	Paper	Scissors
Rock	0	-1	1
Paper	1	0	-1
Scissors	-1	1	0

Alphonse and Beryl must both choose a vector from the set:

$$M = N = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}, \tag{3.6}$$

and if Alphonse chooses x and Beryl chooses y, the payout for Beryl is

$$g(x, y) = x^\top G y. \tag{3.7}$$

By Theorem 3.4 (as well as common sense), the first person to reveal their choice is significantly disadvantaged (since the other person would just pick a winning match-up), and we can compute

$$1 = \min_{x \in M} \max_{y \in N} x^\top G y \geq \max_{y \in N} \min_{x \in M} x^\top G y = -1, \tag{3.8}$$

so we can verify that Inequality 3.4 holds.

We can interpret this game theoretic perspective as some sort of duality. By defining $F(x) = \max_{y \in N} g(x, y)$ as a sort of "dual objective function" to $f(x) = \min_{x \in M} g(x, y)$ (ie., your opponent's objective function is the dual of your own), we can see that Proposition 3.4 resembles some sort of weak duality statement à la linear programming.

Remarkably, however, sometimes it *doesn't matter* who goes first: with optimal play, the disadvantage of revealing your plan early is nonexistent. John von Neumann was the first to publish a strong duality minimax theorem, which many regard as the start of game theory.

Theorem 3.6 (von Neumann, 1928, slightly modified)**.** Let M and N be compact convex sets, and $g : M \times N \to \mathbb{R}$ be a continuous function satisfying

(i) $g(\cdot, y) : M \to \mathbb{R}$ is convex for a fixed $y \in N$

(ii) $g(x, \cdot) : N \to \mathbb{R}$ is concave for a fixed $x \in M$.

Then

$$\min_{x \in M} \max_{y \in N} g(x, y) = \max_{y \in N} \min_{x \in M} g(x, y). \tag{3.9}$$

There have been many generalizations of von Neumann's minimax theorem. We shall prove one of the same flavour by American/Canadian mathematician Maurice Sion:

Theorem 3.7 (Sion, 1958)**.** Let M and N be convex sets, with at least one of them compact, and $g : M \times N \to \mathbb{R}$ be a l.s.c. quasi-convex function in $x \in M$, and an u.s.c. quasi-concave function on $y \in N$. Then

$$\min_{x \in M} \max_{y \in N} g(x, y) = \max_{y \in N} \min_{x \in M} g(x, y). \tag{3.10}$$

Before we prove this theorem, it's important to note that 3.10 may no longer hold if any of the preconditions are false. Let's look at a few case studies with $g(x, y) = x + y$.

Example 3.8. Sometimes, the second to play gains *a lot*! Consider

$$\min_{x\in\mathbb{R}}\max_{y\in\mathbb{R}} x+y = +\infty \tag{3.11}$$

while

$$\max_{y\in\mathbb{R}}\min_{x\in\mathbb{R}} x+y = -\infty \tag{3.12}$$

This example also illustrates the necessity of the *compact* condition in Theorem 3.7.

Example 3.9. On the other hand, if we do enforce compactness, then Sion's Theorem holds as we'd expect:

$$\min_{x\in\mathbb{R}}\max_{0\leq y\leq 1} x+y = -\infty. \tag{3.13}$$

Example 3.10. Some commentary on how Sion's Theorem is not necessary GOES HERE

$$\min_{x\in\mathbb{R}}\max_{y\leq 0} x+y = -\infty. \tag{3.14}$$

We don't have compactness in either set, but Sion's Theorem still holds.

Example 3.11 (von Neumann's Zero Sum Game)**.** Alphonse and Beryl realize that they should play according to a probability distribution, etc., etc., generalize. Let Δ_n be the standard simplex in $\mathbb{R}^n$, then x, y represent the probability distributions, etc., represent something called a mixed strategy.

$$\min_{x\in\Delta_m}\max_{y\in\Delta_n} x^\top Ay = \max_{y\in\Delta_n}\min_{x\in\Delta_m} x^\top Ay. \tag{3.15}$$

3.3.2 Lagrangian Duality

We can use some of the convexity theory to generalize the theory of Lagrange multipliers. (INTRODUCE LAGRANGE MULTIPLIERS somewhere)

Now consider a non-linear program (NLP) given in standard form:

$$\begin{array}{rcc} p^* = & \min & f(x) \\ & \text{s.t.} & g(x) \preceq_K 0 \in \mathbb{E}^m \\ & & h(x) = 0 \in \mathbb{E}^p \\ & & x \in \Omega. \end{array} \tag{3.16}$$

Definition 3.12. In this framework, we can define the **Lagrangian function** with

$$\mathcal{L}(x,\lambda,\mu) := f(x) + \langle \lambda, g(x)\rangle + \langle \mu, h(x)\rangle. \tag{3.17}$$

Here, we introduce two new parameters λ and μ, often called the **dual variables** (as they will become the variables in our dual program), or sometimes simply the Lagrange multipliers. If we are working in $\mathbb{R}^n$, we usually write the more familiar form

$$\mathcal{L}(x,\lambda,\mu) = f(x) + \sum_{i=1}^{m}\lambda_i g_i(x) + \sum_{i=1}^{p}\mu_i h_i(x), \tag{3.18}$$

although it is slightly less revealing. You really should think of the Lagrangian as an affine functional of λ and μ. We can recover our original NLP by solving the following unconstrained problem.

$$p^* = \min_{x\in\Omega} \max_{\substack{\mu\in\mathbb{E} \\ \lambda\succeq_{K^+}0}} \mathcal{L}(x,\lambda,\mu) = f(x) + \langle\lambda, g(x)\rangle + \langle\mu, h(x)\rangle. \tag{3.19}$$

You should convince yourself why these problems are equivalent, and in particular, why the Lagrange multipliers guarantees feasibility in (3.16). By reversing the order of play (by Proposition 3.4) we immediately get a statement of weak duality:

$$p^* \geq d^* = \max_{\substack{\mu\in\mathbb{E} \\ \lambda\succeq_{K^+}0}} \min_{x\in\Omega} \mathcal{L}(x,\lambda,\mu). \tag{3.20}$$

Let's rewrite this a bit more succinctly. Define the **dual functional** of NLP with

$$\phi(\lambda,\mu) = \min_{x\in\Omega} \mathcal{L}(x,\lambda,\mu). \tag{3.21}$$

We rewrite (3.20) and define the dual problem, which will form the basis of **Lagrange relaxation**.

Definition 3.13. Given a program in the form (3.16), we can define the **dual program**

$$d^* = \max_{\substack{\mu\in\mathbb{E} \\ \lambda\succeq_{K^+}0}} \phi(\lambda,\mu). \tag{3.22}$$

Moreover we have weak duality $p^* \geq d^*$.

What's particularly remarkable is that the function ϕ is concave, as it is the pointwise minimum (infimum) of affine functionals. Hence the dual program is *always* a convex problem, even if the primal was not. This *relaxation* of a non-convex problem to a convex one is what we call Lagrange relaxation.

Remark 3.14. Every choice of $\lambda \succeq_K 0, \mu \in \mathbb{E}$ in $\phi(\lambda,\mu)$ will give us a lower bound for p^*. Even if we cannot solve (3.22) exactly, the approximate solutions will give us better and better bounds for the primal.

Of course, our lower bounds could be quite terrible. We'll devote the next section to understanding when we have strong duality, and talk about more about constraint qualification and optimality conditions in the next. MAYBE CHANGE THIS

3.3.3 Strong duality

Now, we need to restrict NLP to the case of a convex program. Recall the abstract convex program (ACP):

$$\begin{aligned} p^* = \quad \min \quad & f(x) \\ \text{s.t.} \quad & g(x) \preceq_K 0 \\ & x \in \Omega. \end{aligned} \tag{3.23}$$

Here, K is a closed convex cone, $\Omega \subseteq \mathbb{E}$ is a convex set (the domain), $f : \Omega \to \mathbb{R}$ is a convex function, and $g : \Omega \to \mathbb{F}$, a K-convex function on Ω. Then the Lagrangian of (3.23) is

$$\mathcal{L}(x, \lambda, \mu) := f(x) + \langle \lambda, g(x) \rangle \tag{3.24}$$

with dual functional

$$\phi(\lambda) = \min_{x \in \Omega} \mathcal{L}(x, \lambda), \tag{3.25}$$

and weak duality

$$p^* \geq d^* := \max_{\lambda \succeq_{K^+} 0} \phi(\lambda). \tag{3.26}$$

Definition 3.15. A **constraint qualification**, CQ, on (ACP) is a condition on the constraints which guarantees the existance of a Lagrange multiplier $\lambda^* \in K^+$ so that we have strong duality. That is, $p^* = d^*$, and the supremum d^* is attained.

We'll explore constraint qualifications for non-convex problems later. For now, with a convex program, we have the very simple Slater's condition.

Definition 3.16 (Slater's condition)**.** Slater's constraint qualification holds when there exists a strictly feasible point, ie., there exists $\widehat{x} \in \Omega$ so that

$$g(\widehat{x}) \prec_K 0. \tag{3.27}$$

Remark 3.17. This is also a reason why we include the domain Ω in the convex problem. If Slater's CQ fails to hold, then sometimes we can modify the constraints and the setΩ appropriately to introduce strict feasibility[13].

And, as alluded to,

Theorem 3.18 (Strong Duality)**.** Suppose that p^* is finite for (ACP), and that Slater's CQ holds. Then there exists a $\lambda^* \in K^+$ such that

$$p^* = \min_{x \in \Omega} L(x, \lambda^*). \tag{3.28}$$

In other words, we have strong duality in the sense that $p^* = d^*$, there is no *duality gap.* Moreover, if x^* is optimal in (ACP), then it is optimal in (3.28) as well, and

$$\langle \lambda^*, g(x^*) \rangle = 0, \tag{3.29}$$

ie., complementary slackness conditions hold.

Proof. ... □

some commentary about how this strong duality doesn't guarantee attainment, and also of feasibility.

Theorem 3.19. Theorem 5.1.12 in Henry's notes

Proof. Rockafellar Pschenichnyi □

[13] This is the idea behind facial reduction. INSERT SOURCES HERE

3.3.4 Optimality Conditions

Existence of KT vectors, Slater's Condition, KT vector implies compactness which allows us to use Sion's theorem.. kkt conditions, idk

3.3.5 Conjugate Duality

Also called Fenchel duality?

4 Algorithms for Unconstrained Problems

Sometimes, as with the equality constrained quadratic problem, it is possible to determine the minimum of a function analytically. When we can't, however, we turn to iterative algorithms. Throughout the next few sections, we shall be trying to solve the unconstrained problem

$$x^* = \arg \min_{x \in \mathbb{R}^n} f(x), \tag{4.1}$$

where f is real valued and sufficiently smooth[14].

We note that many of these algorithms still converge even if f is not convex, however you may find that they do not converge to a global minimum. For now, we'll assume that f is convex as well, but the reader should be aware of the non-convex case.

The following algorithms all follow the same framework. Suppose that we are currently at a point $x^{(n)} \in \mathbb{E}$, and we would like to iterate towards a new point $x^{(n+1)}$ with the hope that, with enough iterations, $x^{(n)} \to x^*$. For notation's sake, we write $\overline{x} := x^{(n)}$ as our current feasible point.

4.1 First Order Methods

The simplest[15] methods are the first order methods—the methods where we move from point to point by considering only information from the first derivative. However, there has been a fair bit of research into improving these methods, especially with the recent popularity of deep learning. Due to the size of some of the problems, the more sophisticated methods are infeasible[16]. Still, first order methods kind of suck and converge very slowly.

4.1.1 Steepest Descent

The first first-order method was originally proposed by Cauchy in 1847. However despite its age, **Cauchy's steepest descent** illustrates many of the design considerations to be aware of today.

[14] The definition of smooth may change from section to section.

[15] and slowest to converge, although still very useful!

[16] For now! Hopefully this changes in the future.

The idea is simple. Locally, around $x^{(n)}$, we know that f is quite well approximated by a linear function. If we picked a direction d, then we could write

$$g(t) := f(\overline{x} + td) = f(\overline{x}) + \nabla f(\overline{x})^\top d + o(\|td\|). \tag{4.2}$$

There are two immediate questions we need to answer: what choice of t (called the step size) and what choice of d (called the descent direction) do we want?

Cauchy suggested that we should choose the direction on which f decreases the fastest, the *steepest* direction, so to speak. In other words, we would like to minimize the directional derivative $\nabla f(\overline{x})^\top d$. To do this, we'd like to solve the subproblem:

$$\begin{array}{ll} \min & \nabla f(\overline{x})^\top d \\ \text{s.t.} & \|d\|^2 = 1. \end{array} \tag{4.3}$$

We pass constraint qualification, so we can use Lagrange multipliers. The Lagrangian is

$$\mathcal{L}(d, \lambda) = \nabla f(\overline{x})^\top d + \lambda(1 - \|d\|^2), \tag{4.4}$$

with derivative (with respect to d)

$$0 = \nabla \mathcal{L}(d, \lambda) = \nabla f(\overline{x}) - 2\lambda d. \tag{4.5}$$

If $\nabla f(\overline{x}) = 0$ then by Theorem 3.3 $\overline{x}$ would be a local optimum, and we'd be done. Otherwise $\nabla f(\overline{x}) \neq 0$, and we can solve the system of equation to conclude

$$\lambda = \pm\frac{1}{2}\left\|\nabla f(\overline{x})\right\|, \qquad d = \pm\frac{\nabla f(\overline{x})}{\|f(\overline{x})\|}. \tag{4.6}$$

We are looking to minimize, so we conclude that $d = -\frac{\nabla f(\overline{x})}{\|f\|(\overline{x})}$ is the direction of steepest descent. Now we must decide on a suitable step length t to guarantee convergence.

Beware of skiing - show what happens if step size is too small or too large, have an example which shows why its not always the best choice to go to the minimum of g,

4.1.2 Subgradient Methods

Ur function is not differentiable :(

4.2 Second Order Methods

4.2.1 Newton's Method

4.2.2 Trust Region Methods

Probably can/should make this its own section below

5 Equality constrained minimization

5.1 Equality Constrained Quadratic Programs

We have already developed enough blah to solve quadratic programs analytically

5.2 Newton's Method Revisited

5.3 Infeasible Start Newton's Method

5.4 Penalty Function

5.5 Implementation Details

6 Interior Point Methods

Technically speaking, the algorithms we've covered above (steepest descent, second order, etc) are interior point methods, as they travel through the interior of a feasible set rather than along the boundary. However the term interior point method typically refers to a class of algorithms called primal-dual interior point methods[17]. We shall develop these algorithms in this section, first in the general context of non-linear programs, and then after looking at the specific cases of linear and quadratic programs, we shall try to extend to the case of an abstract convex problem with inequality constraints over an affine manifold,

$$p^* = \begin{array}{cc} \min & f(x) \\ \text{s.t.} & g(x) \le 0 \\ & Ax = b \\ & x \in \Omega. \end{array} \tag{6.1}$$

and develop the relevant ideas along the way.

We'll begin by discussing methods to convert a constrained optimization problem into a series of unconstrained problems. We'll doing by modifying our objective function to reward feasibility, by either adding a high cost to infeasibility or when approaching the boundary of the feasible region.

6.1 Penalty and Barrier problems

We present a simple and surprisingly powerful technique for solving general nonlinear problems. Suppose we are given a NLP of the form:

$$\begin{array}{rcc} p^* = & \min & f(x) \\ & \text{s.t.} & g(x) \ge 0 \\ & & h(x) = 0 \\ & & x \in \Omega. \end{array} \tag{6.2}$$

Here, $f : \mathbb{R}^n \to \mathbb{R}, g : \mathbb{R}^n \to \mathbb{R}^{m_e}, h : \mathbb{R}^n \to \mathbb{R}^{m_i}$ are *any* functions (well, sufficiently smooth, twice differentiable is usually enough), and Ω is any simple enough constraint set (for example an affine manifold, or perhaps a polyhedron). Note that we make no convex assumptions. We'd like to reformulate this problem into an equivalent unconstrained optimization problem in the hopes of applying Newton's method. Our first try will be to throw a brick at it.

[17] Which again is a misnomer because some methods are primal or dual only.

Recall that, for a general set $S \subseteq \mathbb{E}$, we can define an indicator function $\mathcal{I}_S : \mathbb{E} \to \mathbb{R} \cup \{+\infty\}$ with

$$\mathcal{I}_S(x) = \begin{cases} 0 & x \in S \\ +\infty & x \notin S. \end{cases} \tag{6.3}$$

Then if we let

$$\mathcal{F} = \{x \in \mathbb{R}^n \,:\, g(x) \geq 0, h(x) = 0\} \tag{6.4}$$

be the feasible region, then the unconstrained problem

$$p^* = \min_{x \in \Omega} f(x) + \mathcal{I}_{\mathcal{F}}(x) \tag{6.5}$$

is equivalent to our original problem. This seems to work out nicely, but unfortunately the gravy train stops here. The indicator function is not continuous, meaning we won't be able to use most of our developed algorithms (subgradient method doesn't work well either, as we are not convex). We'll have to introduce soft approximations of these indicator functions instead.

INTRODUCE PENALTY BARRIER with pictures. SPLIT UP EQUALITY AND INEQUALITY CONSTRAINTS?

Then we can define the **joint penalty-barrier function**

$$P_\mu(x) := f(x) + \frac{1}{2\mu} \|h(x)\|^2 - \mu \left(\sum_i \log g_i(x) \right). \tag{6.6}$$

Here, the second term is called the **quadratic penalty** term, and the third term is called the **log barrier** term. We notice that $P_\mu(x)$ is only defined on the interior of the feasible set (points satisfying $g(x) > 0$). So if we start at an interior point, then successive iterations will also be at interior points (hence the name interior point method). Consider the corresponding optimization problem

$$x_\mu = \arg \min_{g(x) > 0, x \in \Omega} P_\mu(x), \tag{6.7}$$

we can see that the quadratic penalty term encourages feasibility of the equality constraints, while the log barrier term encourages us to stay away from the boundary of the set. As μ decreases to 0, the penalty increases and forces $h(x) = 0$, and the barrier decreases and allows the $g(x)$ to get closer to the boundary, while increasing the influence of the objective function f. Formally,

Theorem 6.1 (Penalty Barrier Global Convergence)**.** Let $\{\mu_k\}_{k \geq 1}$ be a sequence approaching 0 from above, and x_{μ_k} be the corresponding optimal solution to (6.7). Then every limit point x^* of the sequence $\{x_{\mu_k}\}_{k \geq 1}$ is a solution to (6.2).

Exercise 6.2. The log barrier is just one of many barrier functions we could have chosen in REFERENCE. Consider blah

$$...f(x)?? - \sum_{j=1}^{p} \frac{1}{g_j(x)} \tag{6.8}$$

6.2 Barrier Methods

We know how to can convert constrained optimization problems to unconstrained ones. Now, we shall adapt the algorithms we've developed for unconstrained optimization to these new barrier problems. Given our convergence above, it may be tempting to just pick a very small μ and solve the corresponding barrier subproblem,

$$\min_x P_\mu(x) = f(x) + \frac{1}{2\mu} \|h(x)\|^2 - \mu \left(\sum_i \log g_i(x) \right), \tag{6.9}$$

as an unconstrained optimization problem. While in theory this will converge to within an ε of the optimal value, in practice due to numerical issues it does not work well except for small and well-behaved problems, and only to a moderate accuracy.

So we will need to extend our unconstrained optimization algorithms in a more intelligent manner. These adaptations are generally called *barrier methods*.

6.3 Primal-dual interior-point Methods

Now let's try to apply our convexity theory to the barrier methods. By using information from both the primal and dual problems to simultaneously update the primal and dual variables at every iteration, we can observe faster convergence than barrier methods. The search directions we obtain are very similar to those obtained in the barrier method, but not identical.

For many basic classes of problems, including linear, quadratic, semidefinite, etc., primal-dual interior-point methods outperform barrier methods. For more complicated/general convex problems, primal-dual algorithms are still under active research, but we have high hopes.

6.4 Linear and Semidefinite Programs

Let's do a few examples, beginning with linear programming. The first primal-dual interior-point methods are usually credited to Karmakar (1984), with their application to linear programming, however barrier methods have been known since the 1950's.

In contrast with the ellipsoid method ADD SECTION ON THIS?, interior point methods lead to efficient polynomial-time algorithms competitive (and in some cases faster than) the celebrated simplex method. In some sense it combines the best of both algorithms: the theoretical guarantees of the ellipsoid method and the blazing fast real-world performance of the simplex method.

Consider the standard equality form linear program and its dual:

$$\begin{array}{rc} \min & c^\top x \\ \text{s.t.} & Ax = b \\ & x \geq 0, \end{array} \qquad\qquad \begin{array}{rc} \max & b^\top y \\ \text{s.t.} & A^\top y \leq c, \end{array} \tag{6.10}$$

although we'll usually write the dual in terms of a slack variable z:

$$\begin{array}{rcrc} \min & c^\top x & \max & b^\top y \\ \text{s.t.} & Ax = b & \text{s.t.} & A^\top y + z = c \\ & x \geq 0, & & z \geq 0. \end{array} \tag{6.11}$$

From a high level perspective: the interior point method will generate a sequence of strictly feasible points $x^{(i)}, y^{(i)}, z^{(i)}$ where $x^{(i)} > 0, y^{(i)} > 0$, converging to the optimal solution (these points are in the *interior*[18] of the feasible region, hence the name of the algorithm). In practice, we can get within 10^{-8} of the optimal solution after 10-50 (expensive) iterations. In fact, we shall show that just $O(n \log \frac{1}{\varepsilon})$ iterations are enough for $(1+\varepsilon)$ of the optimal value (actually interior-point algorithms with just $O(\sqrt{n} \log \frac{1}{\varepsilon})$ iterations exist, but they are slower in practice).

For now, suppose that we only have the primal problem (we shall derive the dual and slack variables along the way):

$$\begin{array}{rc} \min & c^\top x \\ \text{s.t.} & Ax = b \\ & x \geq 0. \end{array} \tag{6.12}$$

We eliminate the non-negativity constraints by introducing a barrier function. Letting $\mu > 0$ be the barrier parameter, we can formulate the barrier subproblem,

$$\min_{x \in \mathbb{R}^n_+} c^\top x - \mu \sum_{i=1}^n \log x_i \text{ subject to: } Ax = b, \tag{6.13}$$

and corresponding Lagrangian function

$$\mathcal{L}(x, y) = c^\top x - \mu \sum_{i=1}^n \log x_i - y^\top (Ax - b). \tag{6.14}$$

By Theorem CITE THEOREM??, we pass constraint qualification, and so at the optimal solution x there exists a y satisfying the following KKT conditions:

$$A^\top y + \mu X^{-1} e = c \tag{6.15}$$

$$Ax = b. \tag{6.16}$$

Here, $X = \operatorname{diag}(x)$, and so on. We notice by slightly rearranging (6.15) we obtain the perturbed complementary slackness conditions,

$$(A^\top y - c)_i \cdot x_i = \mu, \tag{6.17}$$

for all $i \in [n]$ (resembling the complementary slackness conditions we are used to with linear programs, except with μ on the RHS instead of 0).

[18] well, actually relative interior

Let's call $z := c - A^\top y$. Rearranging the KKT equations a bit, we get the perturbed optimality equations

$$A^\top y + z - c = 0, z > 0 \qquad \text{(dual feasibility)} \tag{6.18}$$

$$Ax - b = 0, x > 0 \qquad \text{(primal feasibility)} \tag{6.19}$$

$$Zx - \mu e = 0. \qquad \text{(perturbed complementary slackness)} \tag{6.20}$$

We denote these conditions as $F_\mu(x, y, z) = 0$. Of course, at our current location, if we're not optimal we have $F_\mu(x, y, z) \neq 0$, so we'd like to take a Newton step towards the root. We can find this Newton search direction by solving the equation $\nabla F_\mu(x, y, z)(\Delta x, \Delta y, \Delta z) = -F_\mu(x, y, z)$, or in block equation form,

$$\begin{bmatrix} 0 & A^\top & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} A^\top + z - c \\ Ax - b \\ Zx - \mu e \end{bmatrix}. \tag{6.21}$$

By solving this system we obtain directions $\Delta x, \Delta y, \Delta z$ of descent. After taking an appropriate step size to remain strictly feasible, we update our points x, y, z. Finally, we update μ, usually by setting $\mu \leftarrow \sigma\mu$ for some fixed $\sigma \in (0, 1)$, and push the solution towards optimality.

INSERT PSEUDOCODE?

Insert picture!!

This is the general idea of the algorithm. There are a few details that we have yet to take care of. Over the next few paragraphs, we'll briefly comment on: what to choose for initial values for x, y, z, μ; what step size to take when updating x, y, z; and how to solve (6.21) relatively efficiently in practice. More detailed analysis will be attached in following sections.

While in theory we could just pick any strictly feasible x and z (that is $x, z > 0$), in practice there are heuristics to follow when choosing initial feasible points (not too close to boundary, etc). However for LPs it tends to work okay. Choosing good initial values for x, y, z is very important, and in general is a hard problem. In non-linear programs, it is difficult to even find strictly feasible solutions. We'll discuss this more when we talk about two phase interior point methods.

When possible, the best possible choice for x is at the optimal solution.

Picking appropriate step sizes α and σ is much easier. Again, there are lots of heuristics to follow, and careful tuning of these parameters lead to faster algorithms, however the most important thing is to make sure you remain feasible at each step.

Finally, I'd like to devote a bit of time to exploring how we can efficiently solve (6.21). Letting $r_d = A^\top y + z - c, r_p = Ax - b$, and $r_c = Zx - \mu e$ be the RHS, we can perform some block gaussian elimination. The first row yields

$$\Delta z = -r_d - A^\top \Delta y, \tag{6.22}$$

which when substituted into the third row gives us

$$\Delta x = -Z^{-1} r_c + Z^{-1} X r_d + Z^{-1} X A^\top \Delta y. \tag{6.23}$$

Finally substituting this into the second equation will give us a system

$$AZ^{-1}XA^\top \Delta y = \text{FILL THIS IN.} \tag{6.24}$$

Hence we've reduced our $2n+m$ variable system to one with just m unknowns (and usually $m << n$) and a symmetric LHS , leading to faster numerical methods. We can recover $\Delta x, \Delta z$ by back substitution. (When these algorithms are actually implemented there are more tricks we can do).

In comparison to the simplex, which takes many cheap iterations to converge, primal-dual methods take fewer, more expensive steps. In particular, we need to solve the perturbed KKT equations at each iteration, which is quite computationally expensive. We'll talk more about the time complexity after we choose specifics on the hyperparameters.

Exercise 6.3. We can also formulate the barrier subproblem in terms of the dual: FROM CO 463 notes. Can you derive the same perturbed KKT conditions?

Exercise 6.4. The point of this exercise, as well as the three (COUNT?) that follow, will be to derive a primal-dual interior point method for various quadratic optimization problems, starting with

$$\begin{array}{ll} \min & q(x) = \frac{1}{2}x^\top Qx + c^\top x \\ \text{s.t.} & \quad Ax = b \in \mathbb{R}^m \\ & \quad x \geq 0, x \in \mathbb{R}^n. \end{array} \tag{6.25}$$

Here we are optimizing over an affine manifold, with an additional non-negativity constraint. Derive a primal-dual interior-point algorithm to solve (6.25), by first adding an appropriate log-barrier term, writing down the perturbed optimality conditions, and computing a Newton direction and suitable step length towards optimality.

Implement your solution in your favourite scientific programming language. How fast can you make it?

Exercise 6.5. generalized trust region subproblem

6.5 Feasibility and Two Phase Methods

how to get feasible start point

6.6 Step Size and Time Analysis

Short step, long step, predictor-corrector. Include pictures?

7 Applications

Reorder, maybe pick some of the better ones

7.1 Semidefinite Programming

7.1.1 Preliminaries

Semidefinite programming has been a speciality of the Waterloo C&O Department. Semidefinite programs (SDPs) resemble linear programs, except with our variable is taken in the space of positive semidefinite matrices, and the non-negativity constraints are replaced by semidefinite ones.

Although SDPs have been studied since at least the 1940's (under different names), it wasn't until the late 1900's and early 2000's that we had efficient algorithms for solving them. There are many diverse applications of SDPs, and hopefully I'll be able to show you many of them[19].

We begin with the many equivalent formulations of semidefiniteness:

Proposition 7.1. Let $A \in \mathbb{S}^n$ be a $n \times n$ real symmetric matrix. The following are equivalent:

(i) A is positive semidefinite (p.s.d.), written as $A \succeq 0$ or $A \in \mathbb{S}^n_+$

(ii) For all $x \in \mathbb{R}^n$, $\langle x, Ax \rangle \geq 0$

(iii) All the eigenvalues of A are real and nonnegative

(iv) All $2^n - 1$ principal minors of A are nonnegative

(v) There exists a (Cholesky) factorization $A = LL^\top$

(vi) There exists a (unique) positive semidefinite square root $A = SS$ ($S \in \mathbb{S}^n_+$).

Of course, a very similar statement holds if we restrict A to be a positive definite matrix:

Proposition 7.2. Let $A \in \mathbb{S}^n$ be a $n \times n$ real symmetric matrix. The following are equivalent:

(i) A is positive definite, written as $A \succ 0$ or $A \in \mathbb{S}^n_{++}$

(ii) For all $x \in \mathbb{R}^n$, $x \neq 0 \implies \langle x, Ax \rangle > 0$

(iii) All the eigenvalues of A are real and positive

(iv) All the leading principal minors of A are positive

(v) There exists a (Cholesky) factorization $A = LL^\top$, where L is square and nonsingular

(vi) There exists a (unique) positive definite square root $A = SS$ ($S \in \mathbb{S}^n_{++}$).

Some other linear algebra facts we may use are:

Proposition 7.3. Let $A \in \mathbb{S}^n_+$ and $T \in \mathbb{M}^n$ be non-singular. Then the signature (number of positive, negative, and zero eigenvalues) of A and $T^\top AT$ are the same.

Proposition 7.4 (Schur Complement)**.** The following are equivalent (for appropriately sized A, B, C)

[19] Semidefinite programming is usually its own course at Waterloo, although some other schools (Stanford) cover it into a second convex optimization course. As a result this section may be way more in depth than some of the others.

(i) $\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \succ 0$

(ii) $A \succ 0, C - B^\top A^{-1} B \succ 0$

(iii) $C \succ 0, A - BC^{-1} B^\top \succ 0$

7.1.2 Semidefinite Programming

Recall the primal linear program in standard equality form:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0. \end{aligned} \tag{7.1}$$

Let $\mathcal{A} : \mathbb{S}^n \to \mathbb{E}^m$ be a linear transformation, $b \in \mathbb{E}^m$, and $C \in \mathbb{S}^n$. Then we can write the primal SDP similarly:

$$\begin{aligned} p^* = \min \quad & \langle C, X \rangle \\ \text{s.t.} \quad & \mathcal{A}X = b \\ & X \succeq 0. \end{aligned} \tag{7.2}$$

Note that we can write the linear transformation a bit more explicitly, if we'd like:

$$\mathcal{A}X = \begin{bmatrix} \langle A_1, X \rangle \\ \vdots \\ \langle A_m, X \rangle \end{bmatrix}, \qquad A_i \in \mathbb{S}^n, \tag{7.3}$$

where $\mathcal{A}$ is determined by the covectors $\langle A_i, \cdot \rangle$. We can derive the dual SDP by considering the Lagrangian primal:

$$p^* = \min_{X \succeq 0} \max_{y} \mathcal{L}(X, y) := \langle C, X \rangle + y^\top (b - \mathcal{A}X) \tag{7.4}$$

and rewriting to obtain the Lagrangian dual:

$$d^* = \max_{y} \min_{X \succeq 0} \mathcal{L}(X, y) = b^\top y + \langle X, C - \mathcal{A}^* y \rangle. \tag{7.5}$$

By weak duality (Proposition 3.4) we know that $p^* \geq d^*$. How can we determine when strong duality holds?

interior point method

7.2 Least Squares

7.3 Quadratic Programming (and Support Vector Machines)

7.4 Quadratic Assignment Problem

7.5 Max Cut

7.6 Sensor Network Localization

7.7 ¿Neural Networks?

- unfortunately not very sophisticated techniques but deep learning is a meme... Deep learning as a field is somewhat like alchemy was in the 16th century. There is a lot of stuff that seems to work, but we really have no idea why. We begin with momentum, an interesting idea inspired by physics even if it isn't mathematically supported.

8 Additional Enrichment

8.1 Convex Optimization on Manifolds

A Appendix I: Prerequisites

A.1 Linear Algebra

Linear algebra is the only field of mathematics which is understood[20]. It is a beautiful theory which forms the foundation of mathematics, including of course optimization. As a consequence, it is crucial that you understand it.

I won't go over the basic definitions, I think it's fair to assume that you've taken a first course in linear algebra, and so you know about vector spaces, bases, linear transformations, rank and nullity, range and nullspace (or kernel as I'll call it), and basic properties about eigenvalues and eigenvectors. In particular, it is of vital importance that you think of vectors as abstract coordinate-free objects rather than n-tuples of scalars or pointed arrows.

In this appendix, V will be an arbitrary finite dimensional vector space over the real numbers $\mathbb{R}$, capital letters will represent linear transformations, lower case letters from the earlier part of the alphabet will represent scalars and those from the latter part will represent scalars. I'll pick and choose the relevant parts of linear algebra for optimization. In particular, I will *not* be covering dual spaces, geometry, scalar fields other than $\mathbb{R}$, and so on. This stuff should really be review, you should definitely take more linear algebra courses if it isn't.

[20] In the extremely crude sense that we have answers to most of the questions.

A.2 Calculus

The world is not linear[21]. However, the nonlinear stuff is usually pretty well approximated by the linear stuff, and even better by higher order approximations. We'll formalize this notion and more with ideas from differential[22] calculus, the study of change.

I'll assume that the reader is familiar with many fundamental ideas covered between pre-calculus and freshman differential calculus. You should know and be familiar with the many definitions and characterizations of the real numbers; the epsilon-delta definition of a limit; the definition and properties of the derivative of a function in one variable; the intermediate value and extreme value theorems; the mean value theorem and Taylor's theorem.

I'll cover multivariate differential calculus[23], as well as bits and pieces of real analysis which we will need.

A.2.1 The first derivative

Local linear approximations, fenchel differentiation

A.2.2 The other derivatives

Hessian, Taylor series

A.3 Linear Programming

Linear programming is not strictly a prerequisite for much of convex optimization. However, the theory of linear programs is a fantastic introduction into the field of optimization, and is beautiful in its own right. I sincerely encourage familiarity in the subject out of interest and the many, many applications.

B Appendix II: Proofs and Sketches

Many of the proofs were omitted in the notes, for two reasons. First, a lot of the proofs are easy yet unrevealing, and I feel that including them would not have added much to the subject. Second, I wish to use these notes myself as a theorem reference, and the proofs add a lot of clutter.

However, proofs are still very important. I'm not going to prove all my claims (that is your job as a student) but I'll leave lots of hints and sketches.

[21] Most unfortunately.

[22] Integral Calculus doesn't really appear that much, at least at the level of these notes. I do want to write more one day on differential geometry.

[23] Usually taught in a third calculus class