Non-Linear Optimization 553.762

Class Website: `http://www.ams.jhu.edu/~fishkind/553_762.html`
Notes by David Mueller

Contents

1 Preliminaries

We start off by introducing some formal notation and fundamental theorems in topology. This is going to be a lot of definitions, so I apologize in advance for that.

1.1 Vectors

Definition: Magnitude

The **magnitude** of a vector is, intuitively, what we can think of as the length of that vector from the origin. It's defined as, $\forall \mathbf{x} \in \mathbb{R}^n$:

$$||\mathbf{x}|| = ||\mathbf{x}||_2 = (\sum_{i=0}^{n} \mathbf{x}_i^2)^{\frac{1}{2}}$$

Definition: Distance

The **distance** between two vectors $\mathbf{x}, \mathbf{y}$ is defined as $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$:

$$dist(\mathbf{x}, \mathbf{y}) = ||x - y||$$

Definition: Epsilon-Neighborhood

The **epsilon-neighborhood** of a vector $\mathbf{y} \in \mathbb{R}^n$ is defined $\forall \mathbf{x} \in \mathbb{R}^n, \epsilon > 0$ as the set:

$$N_\epsilon(\mathbf{y}) = \{\mathbf{x} \in \mathbb{R}^n : dist(\mathbf{x}, \mathbf{y}) < \epsilon\}$$

1.2 Sets

Let $S \subseteq \mathbb{R}^n$

Definition: Interior and Boundary Point

$\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{x}$ is an **interior point** of S, $\mathbf{x} \in Int(S) \iff \exists \epsilon > 0$ s.t. $N_\epsilon(\mathbf{x}) \subseteq S$

$\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{x}$ is a **boundary point** of S, $\mathbf{x} \in Bndr(S)$, if $\forall \epsilon > 0, N_\epsilon(\mathbf{x})$ contains a point in S and a point in S^C.

In other words, if there exists some small enough epsilon neighborhood of $\mathbf{x}$ which is entirely contained within S, then $\mathbf{x} \in Int(S)$. The opposite of this is a boundary point, that is for **all** epsilon neighborhood around $\mathbf{x}$, $\mathbf{x}$ contains both a point in S and a point outside of S. *Note that a point cannot be an interior point and a boundary point simultaneously, by definition.*

Definition: Open and Closed Set

S is **open** if $\forall \mathbf{x} \in S, \mathbf{x} \in Int(S)$. That is, if $S = Int(S)$.

S is **closed** if all boundary points of S are **in** S, i.e. if $Bndr(S) \subseteq S$.

To see some intuitive examples of open and closed sets, we can imagine subsets of a numberline

- The range $[2,3]$ is closed in $\mathbb{R}$, but not open.
- The range $(2,3)$ is open, but not closed.
- The range $[2, 3)$ is neither open nor closed.
- And finally, if we trivially consider $S = \mathbb{R}$, then S is both open and closed, since it has no boundary points.

> **Definition: Bounded Set**
>
> S is **bounded** if $\exists \epsilon > 0$ such that $S \subseteq N_\epsilon(\mathbf{0})$. In other words, if it's sort of centered around the origin by some epsilon.
> *We can imagine this as saying that our set*

> **Definition: Compact Set**
>
> S is **compact** if for any sequence $\{\mathbf{x}^i\}_{i=1}^\infty \subseteq S$, then there exists a convergent subsequence $\{\mathbf{x}^{i,j}\}_{j=1}^\infty$, and $\mathbf{y} \in S$ such that $\mathbf{x}^{i,j} \to \mathbf{y}$.

> **Proposition 1.1**
>
> For any $S \in \mathbb{R}^n$, S is open if and only if S^C is closed.

Proposition 1.1. Trivially, $Bndry(S) = Bndry(S^C)$. Also, each point in S is either $Int(S)$ or $Bndry(S)$ but not both. Finally, S is open iff $S = Int(S)$ which is true iff $Bndry(S) \subseteq S^C$ which is true iff $Bndry(S^C) \subseteq S^C$, i.e. S^C is closed. ∎

> **Theorem 1.1**
>
> $\forall S \in \mathbb{R}^n$, nonempty, S is compact iff S is both closed and bounded.

> **Theorem 1.2**
>
> If $S \in \mathbb{R}^n$ is compact, and you have $f : S \to \mathbb{R}$ continuous, then $\exists \bar{\mathbf{x}} \in S$ such that $\forall \mathbf{x} \in S, f(\bar{x}) \leq f(\mathbf{x})$ and $f(\bar{\bar{\mathbf{x}}}) \geq f(\mathbf{x})$.

1.3   Problem Setting

We will typically work with problems of the following form: Suppose $S \subseteq \mathbb{R}^n$, $f : S \to \mathbb{R}$. Define the problem P as:

$$\min f(\mathbf{x}) \text{ s.t. } \mathbf{x} \in S$$

In this setting we call:

- f the objective function
- S the feasible region
- $\mathbf{x}$ the decision variables

The set S may contain equation constraints, and if $\mathbf{x} \in S$ then we say that $\mathbf{x}$ is feasible. Some definitions:

> **Definition: Local and Global Minimum**
>
> If $\bar{\mathbf{x}} \in S$ satisfies $\exists \epsilon > 0$ s.t. $\forall \mathbf{x} \in N_\epsilon(\bar{\mathbf{x}}) \cap S$, $f(\bar{\mathbf{x}}) \leq f(\mathbf{x})$, then $\bar{\mathbf{x}}$ is a **local minimum**.
> If $\bar{\mathbf{x}} \in S$ satisfies $\exists \epsilon > 0$ s.t. $\forall \mathbf{x} \in N_\epsilon(\bar{\mathbf{x}}) \cap S$, $\mathbf{x} \neq \bar{\mathbf{x}}$, $f(\bar{\mathbf{x}}) < f(\mathbf{x})$, then $\bar{\mathbf{x}}$ is a **strict local minimum**.
>
> If $\bar{\mathbf{x}} \in S$ satisfies $\forall \mathbf{x} \in S, f(\bar{\mathbf{x}}) \leq f(\mathbf{x})$ then we call $\bar{\mathbf{x}}$ the **global minimum**,
> If $\bar{\mathbf{x}} \in S$ satisfies $\forall \mathbf{x} \in S, \mathbf{x} \neq \bar{\mathbf{x}}, f(\bar{\mathbf{x}}) < f(\mathbf{x})$, then $\bar{\mathbf{x}}$ is a **strict global minimum**.
>
> If $\bar{\mathbf{x}}$ is a global min of f, then we call $f(\bar{\mathbf{x}})$ the **optimal objective function value (oofv)**.

If $\forall \epsilon > 0, \exists \mathbf{x} \in S$, such that $f(\mathbf{x}) < -\epsilon$, then P is unbounded, and has *no global min*.

1.4 The Gradient and The Hessian

Definition: Gradient

Suppose $S \subseteq \mathbb{R}^n$, S is open, and $f : S \to \mathbb{R}$ is continuously differentiable. Then the **gradient** of f at any $x \in S$ is

$$\nabla f(x) = \begin{bmatrix} \frac{\delta f(\mathbf{x})}{\delta \mathbf{x}_1} \\ \frac{\delta f(\mathbf{x})}{\delta \mathbf{x}_2} \\ \vdots \\ \frac{\delta f(\mathbf{x})}{\delta \mathbf{x}_n} \end{bmatrix}$$

Definition: Hessian

Suppose $S \subseteq \mathbb{R}^n$, S is open, and $f : S \to \mathbb{R}$ is twice cont. diff. The **Hessian** of f at any $\mathbf{x} \in S$ is

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\delta^2 f(\mathbf{x})}{\delta \mathbf{x}_1 \delta \mathbf{x}_1} & \cdots & \frac{\delta^2 f(\mathbf{x})}{\delta \mathbf{x}_1 \delta \mathbf{x}_n} \\ \vdots & \ddots & \vdots \\ \frac{\delta^2 f(\mathbf{x})}{\delta \mathbf{x}_n \delta \mathbf{x}_1} & \cdots & \frac{\delta^2 f(\mathbf{x})}{\delta \mathbf{x}_n \delta \mathbf{x}_n} \end{bmatrix}$$

This is a symmetric matrix!

2 Convexity

2.1 Taylor Series

We'll introduce Taylor Series in 1-dimension first, and then expand the definition to multiple dimensions. These theorems and approximations will be essential for some of our methods much later on.

2.1.1 Taylor's Theorem in 1-dim

Definition: k-th Order Taylor Polynomial

Suppose $f : (a, b) \to \mathbb{R}$ is k-times cont. diff. for a non-neg integer k. Let $\bar{x} \in (a, b)$.

$$P_{\bar{x},k}(x) \approx f(\bar{x}) + f'(\bar{x})(x - \bar{x}) + \frac{1}{2!}f''(\bar{x})(x - \bar{x})^2 + \cdots + \frac{1}{k!}f^{(k)}(\bar{x})(x - \bar{x})^k$$

This is a Taylor Series expansion which approximates $f(x)$ about $\bar{x}$.

Theorem 2.1: Taylor's Theorem

Suppose $f : (a, b) \to \mathbb{R}$ is $k + 1$-times cont. diff. for some non-negative integer k, AND suppose $\bar{x} \in (a, b)$. Then $\forall x \in (a, b)$, $\exists \xi \in (x, \bar{x})$ such that

$$f(x) = P_{\bar{x},k}(x) + R_{\bar{x},k}(\xi) = f(\bar{x}) + f'(\bar{x})(x - \bar{x}) + \frac{f''(\bar{x})}{2!}(x - \bar{x})^2 + \ldots + \frac{f^{(k)}(\bar{x})}{k!}(x - \bar{x})^k$$

$$+ \frac{1}{(k+1)!}f^{(k+1)}(\xi)(x - \bar{x})^{(k+1)}$$

Where the last term here, $R_{\bar{x},k}(\xi)$, is called the *remainder*.
Note that this is an equality, not an approximation as in the Taylor Polynomials

---

Now we want to scale up this theorem to higher-dimensional spaces, since we are often dealing with points in $\mathbb{R}^n$ and gradients rather than derivatives.

2.1.2 Scaling Taylor up to n-dimensions

Now, we want to scale this up to higher dimensions, so we can approximate multi-dimensional functions about a single point. To do this, let's start by noting the following:

Let $S \subseteq \mathbb{R}^n$. For every linear function $f : S \to \mathbb{R}$ we can say $\forall \bar{x} \in \mathbb{R}^n$, $\exists \mathbf{b} \in \mathbb{R}^n, c \in \mathbb{R}$ such that

$$f(\mathbf{x}) = c + \mathbf{b}^T(\mathbf{x} - \bar{\mathbf{x}})$$

i.e. we can *translate* the function to be $\bar{\mathbf{x}}$-centric. Note also that the gradient $\nabla f(\mathbf{x}) = \mathbf{b}$.

We can do the same for any quadratic function $f : S \to \mathbb{R}$, i.e. we can say $\forall \bar{x} \in \mathbb{R}^n$, $\exists A \in \mathbb{R}^{n \times n}$, $\exists \mathbf{b} \in \mathbb{R}^n, c \in \mathbb{R}$ such that

$$f(\mathbf{x}) = c + \mathbf{b}^T(\mathbf{x} - \bar{\mathbf{x}}) + \frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T A(\mathbf{x} - \bar{\mathbf{x}})$$

Note here that the hessian $\nabla^2 f(\mathbf{x}) = A$.

These translations allow us to begin building up a version of Taylor Approximations in n dimensions, eventually leading to a multi-dimensional version of Taylor's Theorem.

Definition: n-dimension Taylor Approximations

Zero'th Order Taylor Approximation:

Let $S \subseteq \mathbb{R}^n$ be open. Also, $f : S \to \mathbb{R}$ is cont. and $\bar{\mathbf{x}} \in S$.

$$P_{\bar{\mathbf{x}},0}(\mathbf{x}) = f(\bar{\mathbf{x}})$$

is the zero'th order taylor approximation of f about $\bar{\mathbf{x}}$.

First Order Taylor Approximation:

Let $S \subseteq \mathbb{R}^n$ be open. Also, $f : S \to \mathbb{R}$ is cont. diff. and $\bar{\mathbf{x}} \in S$.

$$P_{\bar{\mathbf{x}},1}(\mathbf{x}) = f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})$$

is the first order taylor approximation of f about $\bar{\mathbf{x}}$, and is sometimes called the **tangent hyperplane**.

Second Order Taylor Approximation:

Let $S \subseteq \mathbb{R}^n$ be open. Also, $f : S \to \mathbb{R}$ is twice cont. diff. and $\bar{\mathbf{x}} \in S$.

$$P_{\bar{\mathbf{x}},1}(\mathbf{x}) = f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}}) + \frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \nabla^2 f(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})$$

is the second order taylor approximation of f about $\bar{\mathbf{x}}$.

Now, as before, let $S \subseteq \mathbb{R}^n$ be an open set, $f : S \to \mathbb{R}$, $\bar{\mathbf{x}}$, $\mathbf{d} \in \mathbb{R}^n$, $\mathbf{d} \neq \mathbf{0}$. Then, $\forall \alpha \in \mathbb{R}$ such that $\bar{\mathbf{x}} + \alpha \mathbf{d} \in S$, define

$$g_{\bar{\mathbf{x}},\mathbf{d}}(\alpha) = f(\bar{\mathbf{x}} + \alpha \mathbf{d})$$

Proposition 2.1: Gradients of $g_{\bar{\mathbf{x}},\mathbf{d}}$

Assume f is cont. diff. Then

$$\frac{\delta g}{\delta \alpha} g_{\bar{\mathbf{x}},\mathbf{d}}(\alpha) = \nabla f(\bar{\mathbf{x}} + \alpha \mathbf{d})^T \mathbf{d}$$

Assume f is twice cont. diff. Then

$$\frac{\delta^2 g}{\delta^2 \alpha} g_{\bar{\mathbf{x}},\mathbf{d}}(\alpha) = \mathbf{d}^T \nabla^2 f(\bar{\mathbf{x}} + \alpha \mathbf{d}) \mathbf{d}$$

Definition: Descent Direction

If $\exists \beta > 0$ such that $0 < \alpha < \beta$ and

$$f(\bar{\mathbf{x}}) > f(\bar{\mathbf{x}} + \alpha \mathbf{d}) \text{ i.e. } g_{\bar{\mathbf{x}},\mathbf{d}}(0) > g_{\bar{\mathbf{x}},\mathbf{d}}(\alpha)$$

then $\mathbf{d}$ is a **descent direction** for f at $\bar{\mathbf{x}}$. Another way to write this is if f is cont. diff. and $\nabla f(\bar{\mathbf{x}})^T \mathbf{d} < 0$ then d is a descent direction for f at $\bar{\mathbf{x}}$

The **direction of steepest descent** is just $-\nabla f(\bar{\mathbf{x}})$ because it minimizes $\nabla f(\bar{\mathbf{x}})^T \mathbf{d}$ for all $\mathbf{d}$, where $||\mathbf{d}|| = ||\nabla f(\bar{\mathbf{x}})||$.

Now, we have everything we need to build up a first and second order Taylor Theorem in n-dimensions.

Proposition 2.2: n-dimensional Taylor's Theorem

Let f be continuously differentiable. Further, let $\bar{\mathbf{x}} \in S, \mathbf{d} \in \mathbb{R}^n$.

Then, $\forall \alpha > 0$ s.t. $\bar{\mathbf{x}} + \alpha\mathbf{d} \in S, \exists \mathbf{z} \in (\bar{\mathbf{x}}, \bar{\mathbf{x}} + \alpha\mathbf{d})$ s.t.

$$f(\bar{\mathbf{x}} + \alpha\mathbf{d}) = f(\bar{\mathbf{x}}) + \nabla f(\mathbf{z})^T \mathbf{d}\alpha$$

Let f be twice continuously differentiable, all else as above. Then:

$$f(\bar{\mathbf{x}} + \alpha\mathbf{d}) = f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T \mathbf{d}\alpha + \frac{1}{2!}\mathbf{d}^T \nabla^2 f(\mathbf{z})\mathbf{d}$$

Note that we needed g because it is in the form (i.e. it takes in a single real) of the original Taylor Theorem, so we can use it to prove a Taylor Theorem for f.

n-dim. Taylor's Theorem. Let f be cont. diff., then by Taylor's Theorem $\exists a \in (0, \alpha)$ such that

$$g_{\bar{\mathbf{x}},\mathbf{d}}(\alpha) = g_{\bar{\mathbf{x}},\mathbf{d}}(0) + g'_{\bar{\mathbf{x}},\mathbf{d}}(a)\alpha$$

Notice that, by the definition of $g_{\bar{\mathbf{x}},\mathbf{d}}$, this is essentially saying that $\exists \mathbf{z} \in (\bar{\mathbf{x}}, \bar{\mathbf{x}} + \alpha\mathbf{d})$ such that

$$f(\bar{\mathbf{x}} + \alpha\mathbf{d}) = f(\bar{\mathbf{x}}) + \nabla f(\mathbf{z})^T \mathbf{d}\alpha$$

We can repeat this for the second derivative, i.e. let f be twice cont. diff. then by Taylor $\exists b \in (0, \alpha)$ such that

$$g_{\bar{\mathbf{x}},\mathbf{d}}(\alpha) = g_{\bar{\mathbf{x}},\mathbf{d}}(0) + g'_{\bar{\mathbf{x}},\mathbf{d}}(0)\alpha + \frac{g''_{\bar{\mathbf{x}},\mathbf{d}}(b)b\alpha^2}{2!}$$

Which again, implies $\exists \mathbf{z} \in (\bar{\mathbf{x}}, \bar{\mathbf{x}} + \alpha\mathbf{d})$ such that

$$f(\bar{\mathbf{x}} + \alpha\mathbf{d}) = f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T \mathbf{d}\alpha + \frac{1}{2!}\mathbf{d}^T \nabla^2 f(\mathbf{z})\mathbf{d}$$
$$= P_{\bar{\mathbf{x}},1}(\mathbf{x}) + R_{\bar{\mathbf{x}},1}(\mathbf{z})$$

∎

This is generally as far as we'll care to go, since we'll rarely go beyond examining quadratic functions in the class. So now we've achieved a Taylor Theorem which holds in $\mathbb{R}^n$ for our function f. This will be important for proving conditions for optimality and convexity, etc.

2.2 Conditions of Optimality

Before we dive into optimality and convexity, we need to cover two extremely important types of matrices when considering optimization problems. Namely, positive definite and positive semi-definite matrices.

Definition: Positive (Semi-)Definite Matrices

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

A is **positive definite (PD)** if $\forall \mathbf{d} \in \mathbb{R}^n, \mathbf{d} \neq \mathbf{0}, \mathbf{d}^T A\mathbf{d} > 0$.

A is **positive semi-definite (PSD)** if $\forall \mathbf{d} \in \mathbb{R}^n, \mathbf{d} \neq \mathbf{0}, \mathbf{d}^T A\mathbf{d} \geq 0$.

Theorem 2.2

Let $A \in \mathbb{R}^{n \times n}$, symmetric. A is (PD / PSD) iff the eigenvalues of A are all (> 0 / ≥ 0).

> **Theorem 2.3: 1st order necessary condition of optimality**
>
> Let $S \subseteq \mathbb{R}^n$ be an open set, $f : S \to \mathbb{R}$ be continuously differentiable, and $P : \min f(\mathbf{x})$ s.t. $\mathbf{x} \in S$.
>
> If $\bar{\mathbf{x}}$ is a local min of P, then $\nabla f(\bar{\mathbf{x}}) = 0$.

Proof. The proof is just that if you do not have 0 gradient, then you have a descent direction (namely, $\mathbf{d} = -\nabla f(x)$, in which case you are not at a local minimum. ∎

> **Theorem 2.4: 2nd order necessary condition of optimality**
>
> Let $S \subseteq \mathbb{R}^n$ be an open set, $f : S \to \mathbb{R}$ be *twice* continuously differentiable, and $P : \min f(\mathbf{x})$ s.t. $\mathbf{x} \in S$.
>
> If $\bar{\mathbf{x}} \in S$ is a local min of P then $\nabla^2 f(\bar{\mathbf{x}})$ is PSD.

Proof. Otherwise $\exists \mathbf{d} \in \mathbb{R}^n, \mathbf{d} \neq \mathbf{0}$, s.t. $\mathbf{d}^T \nabla^2 f(\bar{\mathbf{x}})\mathbf{d} < 0$. Then $\forall \alpha > 0$, $\exists \mathbf{z} \in (\bar{\mathbf{x}}, \bar{\mathbf{x}} + \alpha\mathbf{d})$ where

$$f(\bar{\mathbf{x}} + \alpha\mathbf{d}) = f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T \mathbf{d}\alpha + \frac{1}{2}\mathbf{d}^T \nabla^2 f(z)\mathbf{d}\alpha^2$$

Note that $\nabla f(\bar{\mathbf{x}})^T \mathbf{d}\alpha = 0$ by the 1st order necessary condition.

Now, by the continuity of $\nabla^2 f$, $\mathbf{d}^T \nabla^2 f(\mathbf{z})\mathbf{d} < 0$ for $\alpha > 0$ small enough. This means $\mathbf{d}$ is a descent direction, which means $\bar{\mathbf{x}}$ is not a local min! ∎

> **Theorem 2.5: 2nd order sufficient condition of optimality**
>
> Let $S \subseteq \mathbb{R}^n$ be an open set, $f : S \to \mathbb{R}$ be *twice* continuously differentiable, and $P : \min f(\mathbf{x})$ s.t. $\mathbf{x} \in S$.
>
> If $\bar{\mathbf{x}} \in S$ satisfies $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$ and $\nabla^2 f(\bar{\mathbf{x}})$ is PD, then $\bar{\mathbf{x}}$ is a strict local min of P.

Proof. Take any direction $\mathbf{d}$, $||\mathbf{d}|| = 1$, and any $\alpha > 0$ small enough. Then we can apply Taylor: $\exists \mathbf{z} \in (\bar{\mathbf{x}}, \bar{\mathbf{x}} + \alpha\mathbf{d})$ s.t.

$$f(\bar{\mathbf{x}} + \alpha\mathbf{d}) = f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T \mathbf{d}\alpha + \frac{1}{2}\mathbf{d}^T \nabla^2 f(\mathbf{z})\mathbf{d}\alpha^2$$

Note again that, since $\bar{\mathbf{x}}$ is a local min, $\nabla f(\bar{\mathbf{x}}) = 0$. Additionally, by the continuity of $\nabla^2 f$, we have that $\nabla^2 f(z)$ is also PD when $\alpha > 0$ is small enough, so **no matter what your d is** you will always increase the value of f is you move in that direction, $\mathbf{d}$, for α steps. Thus, $\bar{\mathbf{x}}$ is a strict local min. ∎

It's worth noting here that if $\nabla f(\bar{\mathbf{x}}) = 0$ and $\nabla^2 f(\bar{\mathbf{x}})$ is PSD, then **this does not imply $\bar{\mathbf{x}}$ is a local min!**. It is a necessary, but not sufficient condition in that case *e.g. consider $f(x) = x^3$*.

2.3   Convex Sets

> **Definition: Convex Set**
>
> $S \subseteq \mathbb{R}^n$ is called convex if $\forall \mathbf{x}, \mathbf{y} \in S, \forall \lambda \in [0, 1]$, then $\lambda\mathbf{x} + (1-\lambda)\mathbf{y} \in S$.

> **Proposition 2.3**
>
> **Proposition**: $S \subseteq \mathbb{R}^n$ is convex iff $\forall k \geq 1, \forall \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(k)}$, and $\forall \lambda_1, \lambda_2, ..., \lambda_k > 0$ then
>
> $$\sum_{i=1}^{k} \lambda_i x^{(i)} \in S$$

Proof. The proof consists of showing that you can always break the sum down into sums of pairs, the lowest of which is in the set S by the definition of convexity, and the rest which follow from induction. It's a little messy, but a very simple proof. I'm omitting the details for sanity's sake. ∎

Proposition 2.4

Suppose $\{S_\gamma\}_{\gamma \in \Gamma}$ is a collection of convex sets in $\mathbb{R}^n$. Then $\cap_{\gamma \in \Gamma} S_\gamma$ is a convex set.

Definition: Convex Hull

For any set $T \subseteq \mathbb{R}^n$, the convex hull of T, $\mathcal{H}(T)$ is defined as the intersection over S,

$$\mathcal{H}(T) = \cap S \in \{S \subseteq \mathbb{R}^n | T \subseteq S, S \text{ is convex}\}$$

i.e. it's the smallest convex set containing T.

Proposition 2.5

For any integer $k \geq 1$, and some sequence $\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^k \in \mathbb{R}^n$ the convex hull of that sequence is:

$$\mathcal{H}(\{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^k\}) = \left\{ \sum_{i=1}^{k} \lambda_i \mathbf{x}^i : \lambda_1, ..., \lambda_k \geq 0, \sum_{i=1}^{k} \lambda_i = 1 \right\}$$

In other words, it's the set defined by all possible weighted linear combinations of the elements of the sequence, such that all the weights sum to 1.

Definition: Extreme point

Let $S \subseteq \mathbb{R}^n$ be convex. *We first show what an extreme point is not.*

$\bar{\mathbf{x}}$ is **not** an extreme point if $\exists \, \mathbf{x}, \mathbf{y} \in S$, $\mathbf{x} \neq \mathbf{y}$, and $\exists \, \lambda \in (0, 1)$ s.t.

$$\bar{\mathbf{x}} = \lambda \mathbf{x} + (1 - \lambda) \mathbf{y}$$

Otherwise, we say that $\bar{\mathbf{x}}$ is an **extreme point**.

Equivalently, we can say that if $\bar{\mathbf{x}}$ is an extreme point of S, then

$$\forall \mathbf{x}, \mathbf{y} \in S, \text{ and } \lambda \in [0, 1] \text{ if } \lambda \mathbf{x} + (1 - \lambda) \mathbf{y} = \bar{\mathbf{x}} \implies \mathbf{x} = \mathbf{y} = \bar{\mathbf{x}}$$

Intuitively, we can think of *extreme points* as points which are sort of on the boundary of our convex set S, and thus cannot be represented as a combination of any other pairs of elements in the set (since creating such a pair would require a point that was "past" the boundary).

2.3.1 Polyhedral Sets

We now move on to cover Polyhedral Sets. These sets have properties which will prove important for our optimization problems. Generally, this is because we will consider constrants which take the form of polyhedral sets.

2.3.2 Hyperplanes and Halfspaces

Definition: Hyperplane and Halfspace

Let $\mathbf{p} \in \mathbb{R}^n, \mathbf{p} \neq \mathbf{0}$.

We define a **hyperplanes** as precisely any set of the form:

$$\{\mathbf{x} \in \mathbb{R}^n : \mathbf{p}^T\mathbf{x} = \alpha\} \text{ s.t. } \mathbf{p} \in \mathbb{R}^n, \mathbf{p} \neq \mathbf{0}, \alpha \in \mathbb{R}$$

Note: Hyperplanes are always an $n-1$ dimension subspace of $\mathbb{R}^n$.

Similarly, we define a halfspace as precisely any set of the form:

$$\{\mathbf{x} \in \mathbb{R}^n : \mathbf{p}^T\mathbf{x} \geq \alpha\} \text{ s.t. } \mathbf{p} \in \mathbb{R}^n, \mathbf{p} \neq \mathbf{0}, \alpha \in \mathbb{R}$$

Let $\mathbf{p} \in \mathbb{R}^n, \mathbf{p} \neq \mathbf{0}$. Then we say that $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{p}^T\mathbf{x} = 0\}$ is a hyperplane *through the origin* with a normal vector $\mathbf{p}$. If we take some $\bar{\mathbf{x}} \in \mathbb{R}^n$, then $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{p}^T(\mathbf{x} - \bar{\mathbf{x}}) = 0\}$ is a hyperplane through $\bar{\mathbf{x}}$ with normal vector $\mathbf{p}$ (as opposed to the origin).

Intuitively, you can think of a hyperplane separating a space $\mathbb{R}^n$ in "half", and a half-space is all points on one "side" of the hyperplane.

2.3.3 Polyhedral Sets

Definition: Polyhedral Set

A **polyhedral set** is an intersection of finitely many half-spaces.

Proposition 2.6

$S \subseteq \mathbb{R}^n$ is a *polyhedral set* iff $\exists A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$ s.t. $S = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \geq \mathbf{b}\}$.

Proof. We sketch the proof, for simplicity.

Write $A = \begin{bmatrix} \mathbf{p}^{(1)} \\ \mathbf{p}^{(2)} \\ \vdots \\ \mathbf{p}^{(m)} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b^{(1)} \\ b^{(2)} \\ \vdots \\ b^{(m)} \end{bmatrix}$. Note that $\mathbf{x}$ satisfies $A\mathbf{x} = \mathbf{b}$ iff $\mathbf{x}$ satisfies

$$\mathbf{p}^{(1)T}\mathbf{x} \geq b^{(1)}, \mathbf{p}^{(2)T}\mathbf{x} \geq b^{(2)}, ..., \mathbf{p}^{(m)T}\mathbf{x} \geq b^{(m)}$$

which is exactly an intersection of hyperspaces. ∎

Proposition 2.7

Polyhedral sets are convex

Proof. $\forall A \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^n$ consider the polyhedral set $S = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \geq \mathbf{b}\}$.

Then, for $\forall \mathbf{y}, \mathbf{z} \in S, \forall \lambda \in [0,1]$ we know that $A\mathbf{y} \geq \mathbf{b}$, $A\mathbf{z} \geq \mathbf{b}$, and therefore

$$A(\lambda\mathbf{y} + (1-\lambda)\mathbf{z}) = \lambda A\mathbf{y} + (1-\lambda)A\mathbf{z} \geq \lambda\mathbf{b} + (1-\lambda)\mathbf{b} = \mathbf{b} \implies \lambda\mathbf{y} + (1-\lambda)\mathbf{z} \in S$$

∎

2.3.4 Extreme Points

> **Theorem 2.6**
>
> Suppose $A \in \mathbb{R}^{m \times n} = [A^{(1)} | A^{(2)} | \dots | A^{(m)}], \mathbf{b} \in \mathbb{R}^m$ and $S = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$.
>
> *Note that S is a polyhedral set! We can write it as* $\left\{ \mathbf{x} \in \mathbb{R}^n : \begin{bmatrix} A \\ A^{-1} \\ I \end{bmatrix} \mathbf{x} \geq \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\}$
>
> Then, for any $\bar{\mathbf{x}} \in S$, $\bar{\mathbf{x}}$ is an extreme point of S iff $\left\{A^{(i)}\right\}_{\bar{\mathbf{x}}_i \neq 0}$ is linearly independent.
>
> $\left\{A^{(i)}\right\}_{\bar{\mathbf{x}}_i \neq 0}$ *is the set of columns of A for which the corresponding component of $\bar{\mathbf{x}}$ is non-zero.*

Proof. ($\impliedby$): Suppose you have $\bar{\mathbf{x}} \in S$ s.t. $\left\{A^{(i)}\right\}_{\mathbf{x}_i \neq 0}$ are linearly independent and suppose also that $\exists\, \mathbf{y}, \mathbf{z} \in S$ and $\lambda \in (0, 1)$ such that $\bar{\mathbf{x}} = \lambda \mathbf{y} + (1 - \lambda)\mathbf{z}$ *(i.e. assume that $\bar{\mathbf{x}}$ is not an extreme point)*.

Then $\forall i$ where $\bar{\mathbf{x}}_i = 0 \implies 0 = \lambda \mathbf{y}_i + (1 - \lambda)\mathbf{z}_i$. Since $\lambda, (1 - \lambda) > 0$ and $\mathbf{y}_i, \mathbf{z}_i \geq 0$ by membership in S, we know that $\mathbf{y}_i = \mathbf{z}_i = 0$.

Now, note that

$$A\mathbf{x} = \begin{bmatrix} A_{1,1}\mathbf{x}_1 + A_{1,2}\mathbf{x}_2 + \dots + A_{1,n}\mathbf{x}_n \\ A_{2,1}\mathbf{x}_1 + A_{2,2}\mathbf{x}_2 + \dots + A_{2,n}\mathbf{x}_n \\ \vdots \\ A_{m,1}\mathbf{x}_1 + A_{m,2}\mathbf{x}_2 + \dots + A_{m,n}\mathbf{x}_n \end{bmatrix} = A^{(1)}\mathbf{x}_1 + A^{(2)}\mathbf{x}_2 + \dots + A^{(n)}\mathbf{x}_n$$

Therefore,

$$\mathbf{b} = \sum_{i : \bar{\mathbf{x}}_i \neq 0} \bar{\mathbf{x}}_i A^{(i)} = \sum_{i : \bar{\mathbf{x}}_i \neq 0} \mathbf{y}_i A^{(i)} = \sum_{i : \bar{\mathbf{x}}_i \neq 0} \mathbf{z}_i A^{(i)}$$

because both $\mathbf{y}, \mathbf{z}$ are in S. However, by linear independence, this can only be true if $\forall i : \bar{\mathbf{x}}_i \neq 0, \bar{\mathbf{x}}_i = \mathbf{y}_i = \mathbf{z}_i \implies \bar{\mathbf{x}} = \mathbf{y} = \mathbf{z}$, by our previous result about the zero-components of $\bar{\mathbf{x}}$. Thus, $\bar{\mathbf{x}}$ is an extreme point.

($\implies$): Suppose $\bar{\mathbf{x}} \in S$ s.t. $\left\{A^{(i)}\right\}_{\mathbf{x}_i \neq 0}$ is **not** linearly independent. Then,

$$\exists\, \mathbf{w} \in \mathbb{R}^n \mathbf{w} \neq \mathbf{0}, \text{ and } \forall i : \mathbf{w}_i \neq 0 \implies \bar{\mathbf{x}}_i \neq 0 \implies \bar{\mathbf{x}}_i > 0 \text{ s.t. } A\mathbf{w} = \mathbf{0}$$

In otherwords, there exists a $\mathbf{w} \in \mathbb{R}^n$ with the same non-zero components of $\mathbf{x}$, such that $A\mathbf{w} = \mathbf{0}$.

Let ϵ be small enough so that $\hat{\mathbf{x}} = \bar{\mathbf{x}} + \epsilon \mathbf{w} \geq \mathbf{0}$ and $\hat{\hat{\mathbf{x}}} = \bar{\mathbf{x}} - \epsilon \mathbf{w} \geq \mathbf{0}$. Because $\mathbf{w} \neq \mathbf{0}, \hat{\mathbf{x}} \neq \hat{\hat{\mathbf{x}}}$, and since we chose epsilon small enough, $\hat{\mathbf{x}}, \hat{\hat{\mathbf{x}}} \geq \mathbf{0}$. Additionally, $A\hat{\mathbf{x}} = A(\bar{\mathbf{x}} + \epsilon \mathbf{w}) = A\bar{\mathbf{x}} + \mathbf{0} = \mathbf{b}$, and similarly for $\hat{\hat{\mathbf{x}}}$. Therefore, both $\hat{\mathbf{x}}, \hat{\hat{\mathbf{x}}} \in S$.

Finally, we have $\bar{\mathbf{x}} = \frac{1}{2}\hat{\mathbf{x}} + \frac{1}{2}\hat{\hat{\mathbf{x}}} \implies \bar{\mathbf{x}}$ is **not an extreme point**. $\blacksquare$

2.4 Convex Functions

2.4.1 The Epigraph

> **Definition: Convex Functions**
>
> Let $S \subseteq \mathbb{R}^n$ be a non-empty, convex set and $f : S \to \mathbb{R}$
>
> f is **convex** iff $\forall \mathbf{x}, \mathbf{y} \in S, \lambda \in [0, 1]$ it holds that
>
> $$f\left(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}\right) \leq \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y})$$
>
> f is **strictly convex** iff $\forall \mathbf{x}, \mathbf{y} \in S, \mathbf{x} \neq \mathbf{y}, \lambda \in (0, 1)$ it holds that
>
> $$f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) < \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y})$$

In 2 dimensions this is essentially just saying that, for any secant line between 2 points, the function value has to be "under the secant line".

Note a few properties of convex functions:

- f is convex iff $-f$ is *concave*.
- $\forall c \geq 0, cf$ is convex if f is convex
- the sum of convex functions is convex

> **Definition: Epigraph**
>
> Let $S \subseteq \mathbb{R}^n$ be a convex, non-empty set, and $f : S \to \mathbb{R}$. The **epigraph** of f is defined as
>
> $$Ep(f) = \left\{ \begin{bmatrix} \mathbf{x} \\ \beta \end{bmatrix} \in \mathbb{R}^{n+1} : \mathbf{x} \in S, \beta \in \mathbb{R}, \beta \geq f(\mathbf{x}) \right\}$$

In 2 dimensions, you can sort of think of this as the area over the curve, or like the... air of a hilly 3-D landscape. It's everything above the function, if we added an extra dimension to help visualize our space.

> **Theorem 2.7**
>
> Suppose $S \subseteq \mathbb{R}^n$ is a convex, non-empty set and $f : S \to \mathbb{R}$. Then f is a convex function iff $Ep(f)$ is a convex set.

The proof of this is omitted, but is rather trivial with respect to the definition of convex sets and functions.

2.4.2 The Support Theorem

> **Theorem 2.8: Support Theorem**
>
> $S \subseteq \mathbb{R}^n$ non-empty, convex set. Let $\bar{\mathbf{x}} \in Bndr(S)$. Then $\exists$ a hyperplane containing $\bar{\mathbf{x}} \in \mathbb{R}^n$ such that all of S is contained in one of the two half-spaces.

*Intuitively, we can think of this as follows: If a set of convex, then any of it's boundary points define a tangent hyperplane which contains the boundary point but **does not split the set**. In other words, the tangent hyperplane **supports**, or holds the set up.*

> **Theorem 2.9**
>
> Let $S \in \mathbb{R}^n$ be a non-empty, open, convex-set, and let $f : S \to \mathbb{R}$ be continuously differentiable.
>
> Then
> - f is convex iff $\forall \mathbf{x}, \bar{\mathbf{x}} \in S$, $f(\mathbf{x}) \geq f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}})$
> - f is strictly convex iff $\forall \mathbf{x}, \bar{\mathbf{x}} \in S$, $\mathbf{x} \neq \bar{\mathbf{x}}$, $f(\mathbf{x}) > f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}})$

This is basically showing us that the tangent hyperplane is the only possible candidate for supporting hyperplane on the boundary of $Epi(f)$.

> **Theorem 2.10**
>
> Let $S \in \mathbb{R}^n$ be a non-empty, open, convex set. Let $f : S \to \mathbb{R}$ be twice continuously differentiable.
>
> Then f is convex iff $\forall \mathbf{x} \in S, \nabla^2 f(\mathbf{x})$ is *Positive Semi-Definite*(PSD).

Proof. ($\Longleftarrow$): Suppose $\nabla^2 f(x)$ is PSD for all points $\mathbf{x} \in S$. By Taylor $\forall \mathbf{x}, \bar{\mathbf{x}} \in S$

$$f(\mathbf{x}) = f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) + \frac{1}{2} (\mathbf{x} - \bar{\mathbf{x}})^T \nabla^2 f(\mathbf{z})(\mathbf{x} - \bar{\mathbf{x}})$$

where $\mathbf{z}$ is some point between $\mathbf{x}, \bar{\mathbf{x}}$ also in S, since S is convex. Now, note that the last term is ≥ 0, since the Hessian is PSD. Thus,

$$f(\mathbf{x}) \geq f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}})$$

and so f is convex by Thm. 5.3.

($\implies$): Suppose $\exists \bar{\mathbf{x}} \in S$ s.t. $\nabla^2 f(\bar{\mathbf{x}})$ is **not** PSD. Then $\exists \mathbf{d} \in \mathbb{R}^n$ such that $\mathbf{d}^T \nabla^2 f(\bar{\mathbf{x}}) \mathbf{d} < 0$. Then, by continuity of $\nabla^2 f$, there exists an epsilon neighborhood around $\bar{\mathbf{x}}$ such that $\forall \mathbf{z}$ in this neighborhood, $\mathbf{d}^T \nabla^2 f(\mathbf{z}) \mathbf{d} < 0$. For any $\alpha > 0$ small enough such that $\mathbf{x} = \bar{\mathbf{x}} + \alpha \mathbf{d}$ in this neighborhood, then by Taylor

$$\begin{aligned} f(\mathbf{x}) &= f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) + \frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \nabla^2 f(\mathbf{z})(\mathbf{x} - \bar{\mathbf{x}}) \\ &= f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) + \frac{1}{2}(\alpha \mathbf{d})^T \nabla^2 f(\mathbf{z})(\alpha \mathbf{d}) \\ &= f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}) + \frac{1}{2}\alpha^2 \mathbf{d}^T \nabla^2 f(\mathbf{z}) \mathbf{d} \end{aligned}$$

Note also that

$$\alpha^2 \mathbf{d}^T \nabla^2 f(\mathbf{z}) \mathbf{d} < 0 \implies f(\mathbf{x}) < f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})$$

so f is not convex, again by Thm 5.3. ∎

Theorem 2.11

Let $S \in \mathbb{R}^n$ be a non-empty, open, convex set. Let $f : S \to \mathbb{R}$ be twice continuously differentiable.

Then, if $\forall \mathbf{x} \in S$, $\nabla^2 f(\mathbf{x})$ is PD, then f is **strictly convex**.

Proof. Again by Taylor, $\forall \mathbf{x}, \bar{\mathbf{x}} \in S$

$$f(\mathbf{x}) = f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}}) + \frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \nabla^2 f(\mathbf{z})(\mathbf{z} - \bar{\mathbf{x}})$$

As before, the last term here is strictly > 0, so $f(\mathbf{x}) > f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})$, which shows that f is strictly convex. ∎

Note: f strictly convex $\implies\!\!\!/\;\; \forall \mathbf{x} \in S, \nabla^2 f(\mathbf{x})$ is PD. It does imply that it's PSD everywhere. To see a simple example, take x^4. This function is twice diff., and strictly convex, but the second derivative at $x = 0$ is 0.

Theorem 2.12

Let $S \subseteq \mathbb{R}^n$ non-empty, convex, open, and $f : S \to \mathbb{R}$ be a convex, continuously differentiable function. Finally, let $P : min f(\mathbf{x})$, $\mathbf{x} \in S$.

Then $\bar{\mathbf{x}} \in S$ is a global minimum of P iff $\bar{\mathbf{x}}$ is a local minimum iff it's a stationary point.

(i.e. in a convex fn, if you are a local min, you are a global min).

Proof. First note if $\bar{\mathbf{x}}$ is a global min, then it is a local min, which finally means it is a stationary point.

By convexity, and thm 5.3, $\forall \mathbf{x} \in S$, $f(\mathbf{x}) \geq f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})$. Since $\bar{\mathbf{x}}$ is a stationary point, $\nabla f(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})$ is 0, and thus $\forall \mathbf{x} \in S$, $f(\mathbf{x}) \geq f(\bar{\mathbf{x}}) \implies \bar{\mathbf{x}}$ is a global minimum.

Finally, note that if f is strictly convex, then the global min is strict as well. ∎

3 Linear Programming

Now we get into the first, general type of problem that we will be dealing with: Optimizing a linear function with linear constraints.

3.1 Standard and Canonical Form

Definition: Standard and Canonical Form

Let $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$.

The **Standard form** of a linear program LP is:

$$\min \mathbf{c}^T \mathbf{x}$$
$$\text{s.t. } A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}$$

The **Canonical form** or *Symmetric form* of an LP is:

$$\min \mathbf{c}^T \mathbf{x}$$
$$\text{s.t. } A\mathbf{x} \geq \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}$$

Note that in either case, our variables are the components of $\mathbf{x} \in \mathbb{R}^n$.

Note that we can easily go between the two forms. The primary difference here is that Standard form expects $A\mathbf{x} = \mathbf{b}$ and Canonical allows $A\mathbf{x} \geq \mathbf{b}$.

- Standard to canonical $(= \rightarrow \geq)$: We've already seen how to do this when discussing Polyhedral Sets.

$$A\mathbf{x} = \mathbf{b} \rightarrow \begin{bmatrix} A \\ -A \end{bmatrix} \mathbf{x} \geq \mathbf{b}$$

- Canonical to standard $(\geq \rightarrow =)$: This one is less obvious. We need to introduce a slack variable. Let's take an example in one dimension.
$$A_{1,1}\mathbf{x}_1 + A_{1,2}\mathbf{x}_2 \geq b \rightarrow A_{1,1}\mathbf{x}_1 + A_{1,2}\mathbf{x}_2 - \mathbf{x}_3 = b$$
$$\text{s.t. } \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \geq 0$$

  We call $\mathbf{x}_3$ our slack variable. We can make one of these for each of our m rows (constraints), and put them into a vector called $\mathbf{z}$, in the original problem. Thus, we can incorporate these to convert our canonical form into standard form in the following way:
$$A\mathbf{x} \geq \mathbf{b} \rightarrow \begin{bmatrix} A & | & -I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} = \mathbf{b}$$

Other transformations we can make to get things into the correct form:

- $\max \mathbf{c}^T\mathbf{x} = \min -\mathbf{c}^T\mathbf{x}$

- unsigned variable $\mathbf{x}_1$ can be converted into $\mathbf{x}_1 - \mathbf{x}_2$, where both $\mathbf{x}_1, \mathbf{x}_2 \geq 0$.

- we can rearrange variables and columns, e.g.

$$\begin{bmatrix} A^{(1)} & A^{(2)} & A^{(3)} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} A^{(2)} & A^{(3)} & A^{(1)} \end{bmatrix} \begin{bmatrix} \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_1 \end{bmatrix}$$

- We can perform basic *row operations*. Take an augmented matrix representing $A\mathbf{x} = \mathbf{b}$, i.e. $\begin{bmatrix} A & | & \mathbf{b} \end{bmatrix}$. We can always

  1. Multiply a row by non-zero scalar

  2. Swap any 2 rows

  3. Add a row to another

  These row operations can be accomplished through multiplying the augmented matrix with invertible matrices (we can always undo them).

None of these transformations change the solution of the original problem, so we're free to change things up in these ways as we please.

3.2   Basic Feasible Solutions

Let $A \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m, \mathbf{c}, \mathbf{x} \in \mathbb{R}^n$. Consider an LP in standard form:

$$\min \mathbf{c}^T\mathbf{x}$$
$$\text{s.t. } A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$$

Definition: Basic Feasible Solution

WLOG assume that A is full row-rank. *If it isn't, we can row-reduce it to remove redundant constraints.*

If we can write $A = \begin{bmatrix} B & | & N \end{bmatrix}$ where $B \in \mathbb{R}^{m \times m}$ is invertible, and $B^{-1}\mathbf{b} \geq \mathbf{0}$, then

$$\bar{\mathbf{x}} = \begin{bmatrix} B^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{x}}_B \\ \bar{\mathbf{x}}_N \end{bmatrix}$$

is feasible in LP, and called a **basic feasible solution (bfs)**.

We can additionally write $\mathbf{c} = \begin{bmatrix} \mathbf{c}_B \\ \mathbf{c}_N \end{bmatrix}$, and thus the objective function value of this bfs is

$$\mathbf{ofv}(\bar{\mathbf{x}}) = \mathbf{c}_B^T B^{-1}\mathbf{b}$$

In this scenario, we call B the *basis*, $\bar{\mathbf{x}}_B$ the *basic variables*, and $\bar{\mathbf{x}}_N$ the *non-basic variables*.

In general, if $B \in \mathbb{R}^{m \times m}$ is a sub-matrix of A, say

$$B = \begin{bmatrix} A^{(k_1)} & | & A^{(k_2)} & | & A^{(k_3)} & | & \dots & | & A^{(k_m)} \end{bmatrix}$$

and B is invertible and $B^{-1}\mathbf{b} \geq \mathbf{0}$, then we can define an $\bar{\mathbf{x}} \in \mathbb{R}^n$ where $\forall i = 1, \dots, m$ $\bar{\mathbf{x}}_i = (B^{-1}\mathbf{b})_i$ and all other components of $\bar{\mathbf{x}}$ are zero. It's obvious to see why this $\bar{\mathbf{x}}$ is feasible in LP, and thus constitutes a solution (not necessarily an optimal solution in LP).

Note that any bfs is an extreme point of the feasible region of an LP. To see this, simply notice that all of the basis columns (where $\bar{\mathbf{x}}_i$ is non-zero) are linearly independent.

3.2.1 Reduced Cost Coefficients

Continuing as above, let's suppose $\bar{\mathbf{x}}$ is a bfs of LP. Then, WLOG we can write

$$A = \begin{bmatrix} B & | & N \end{bmatrix}, \bar{\mathbf{x}} = \begin{bmatrix} \bar{\mathbf{x}}_B \\ \bar{\mathbf{x}}_N \end{bmatrix} = \begin{bmatrix} B^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix}, \mathbf{c} = \begin{bmatrix} \mathbf{c}_B \\ \mathbf{c}_N \end{bmatrix}$$

Remember, we can always re-arrange our rows and columns until we get to this point!

We can also take any other $\mathbf{x}$ and write it as $\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix}$, separating the components which multiply into the basis B and the components which multiply by the other part of A, N. If $\mathbf{x}$ is feasible, $A\mathbf{x} = \mathbf{b}$, then that means $B\mathbf{x}_B + N\mathbf{x}_N = \mathbf{b}$, which we can rewrite as

$$\mathbf{x}_B = B^{-1}(\mathbf{b} - N\mathbf{x}_N) = \bar{\mathbf{x}}_B - B^{-1}N\mathbf{x}_N$$

Thus, we can re-write the *basic* components of *any* $\mathbf{x}$ in terms of our bfs ($\bar{\mathbf{x}}$), and the non-basic variables of $\mathbf{x}$! Thus, any feasible $\mathbf{x}$ consists of an $\mathbf{x}_N \geq \mathbf{0}$ such that $\mathbf{x}_B = B^{-1}(\mathbf{b} - N\mathbf{x}_N) \geq \mathbf{0}$.

This is kind of like looking at the point $\mathbf{x}$ through the lens of Basis B, where our non-basic variables $\mathbf{x}_N$ determine "how far" $\mathbf{x}$ is from $\bar{\mathbf{x}}$.

All feasible points $\mathbf{x}$ can be enumerated like this. Such a point $\mathbf{x}$ has ofv

$$\begin{aligned}
\mathbf{ofv}(\mathbf{x}) = \mathbf{c}^T\mathbf{x} &= \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N \\
&= \mathbf{c}_B^T B^{-1}(\mathbf{b} - N\mathbf{x}_N) + \mathbf{c}_N^T\mathbf{x}_N \\
&= \mathbf{c}_B^T B^{-1}\mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1}N)\mathbf{x}_N
\end{aligned}$$

Note that $\mathbf{c}_B^T B^{-1}\mathbf{b}$ is the ofv of our bfs $\bar{\mathbf{x}}$.

Definition: Reduced Cost Coefficients

All notation as above.

Let $\mathbf{x} \in \mathbb{R}^n$ be feasible in LP, with ofv:

$$\mathbf{ofv}(\mathbf{x}) = \mathbf{c}_B^T B^{-1}\mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1}N)\mathbf{x}_N$$

Then $(\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1}N) = \mathbf{r}_N^T$ are called the **reduced cost coefficients**.

Each component of $\mathbf{r}_N$ corresponeds to a component of the *non-basic* variables $\mathbf{x}_N$.

The variables of $\mathbf{x}_B$ are now *implicit*, given $\mathbf{x}_N$ and B.

Finally, note that from this equation we can get $\frac{\delta ofv}{\delta \mathbf{x}_N} = \mathbf{r}_N$. This rate of change corresponds to how the ofv of $\bar{\mathbf{x}}$ increases or decreases as we change $\mathbf{x}_N$.

Proposition 3.1

Let B be a basis of LP, and assume that $\bar{\mathbf{x}}$ denote the basic feasible solution for basis B.

If the reduced cost coefficients defined by $\bar{\mathbf{x}}$ and B are all positive,

$$(\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1}N) = \mathbf{r}_N^T \geq \mathbf{0}$$

then your bfs $\bar{\mathbf{x}}$ is optimal in LP.

Proof. We provide a sketch of the proof

We can see *every feasible point* from the basis-lens of $\bar{\mathbf{x}}$. If our reduced cost coefficients are all positive that implies that

$$\forall \mathbf{x} \text{ feasible in LP}, \mathbf{ofv}(\mathbf{x}) \geq \mathbf{ofv}(\bar{\mathbf{x}})$$

since $\mathbf{x} \geq \mathbf{0}$ is a constraint on feasibility. So no other feasible point can have a lower ofv than our current bfs $\bar{\mathbf{x}}$. Thus, $\bar{\mathbf{x}}$ is optimal! ∎

3.3 The Simplex Method

3.3.1 The Simplex Tableau

Consider an LP in standard form, where $A \in \mathbb{R}^{m \times n}$ is rank m:

$$\min \mathbf{c}^T \mathbf{x} \ \text{ s.t. } \ A\mathbf{x} = \mathbf{b},$$
$$\mathbf{x} \geq \mathbf{0}$$

WLOG, $A = \begin{bmatrix} B & | & N \end{bmatrix}$. where $B = \mathbb{R}^{m \times m}$ invertible. *Since A is rank m, we can row-reduce to make this true.*

Definition: Pre-Tableau

All notation as above.

The **pre-tableau** is the *augmented matrix* defined as:

$$PreT = \begin{bmatrix} 1 & -\mathbf{c}^T & | & 0 \\ \mathbf{0} & A & | & \mathbf{b} \end{bmatrix} = \begin{bmatrix} 1 & -\mathbf{c}^T & | & 0 \\ \mathbf{0} & B \mid N & | & \mathbf{b} \end{bmatrix}$$

In essence, we can think of this augmented matrix as implying that there is some slack variable $z \in \mathbb{R}$ s.t. $z - \mathbf{c}^T \mathbf{x} = 0$ (on the top row), and asserting that $A\mathbf{x} = \mathbf{b}$ in all the other rows.

However, we are actually interested in *Basic Feasible Tableaus* in the simplex method. To convert this into *basic form*, we can use the invertible matrix $H \in \mathbb{R}^{m+1 \times m+1}$:

$$H = \begin{bmatrix} 1 & \mathbf{c}_B^T B^{-1} \\ \mathbf{0} & B^{-1} \end{bmatrix}$$

Which allows us to convert out Pre-tableau into a *basic tableau*:

Definition: Basic Tableau and Basic Feasible Tableau

All notation as above.

A **Basic Tableau** is any tableau of the form:

$$Basic\,Tableau = H \times PreT = \begin{bmatrix} 1 & \mathbf{0}^T & -\mathbf{c}_N^T + \mathbf{c}_B^T B^{-1} N & | & \mathbf{c}_B^T B^{-1} \mathbf{b} \\ \mathbf{0} & I & B^{-1}N & | & B^{-1}\mathbf{b} \end{bmatrix}$$

If $B^{-1}\mathbf{b} \geq \mathbf{0}$, then $Basic\,Tableau$ is called the **Basic Feasible Tableau** of basis B with basic feasible solution $\bar{\mathbf{x}} = \begin{bmatrix} \bar{\mathbf{x}}_B \\ \mathbf{0} \end{bmatrix}$.

In this case, we can rewrite the Tableau as:

$$\begin{bmatrix} 1 & \mathbf{0}^T & -\mathbf{r}_N^T & | & \mathbf{ofv}(\bar{\mathbf{x}}) \\ \mathbf{0} & I & \dots & | & \bar{\mathbf{x}}_B \end{bmatrix}$$

Firsly, note that if $B^{-1}\mathbf{b} \geq \mathbf{0}$, then $\bar{\mathbf{x}} = \begin{bmatrix} B^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix}$ is feasible in LP, making it a basic feasible solution and thus making the tableau a *basic feasible tableau*).

From any basic tableau, we can derive a couple key things. We can write out z, our "slack variable" for the objective function value in terms of the non-basic components of $\mathbf{x}$:

$$z - (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N)\mathbf{x}_N = \mathbf{c}_B^T B^{-1}\mathbf{b}$$
$$\implies z = \mathbf{c}_B^T B^{-1}\mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N)\mathbf{x}_N$$

which (if $B^{-1}\mathbf{b} \geq \mathbf{0}$) is *exactly* the **ofv**($\mathbf{x}$) written our with respect to the **ofv** of the basic feasible solution $\bar{\mathbf{x}}$ and the reduced cost coefficients:

$$z = \mathbf{ofv}(\bar{\mathbf{x}}) + r_N^T \mathbf{x}_N$$

We can similarly write out the basic components of $\mathbf{x}$ using the non-basic components:

$$\mathbf{x}_B = B^{-1}\mathbf{b} - B^{-1}N\mathbf{x}_N = \bar{\mathbf{x}}_B - B^{-1}N\mathbf{x}_N$$

Lastly, note that if the top row (not the scalars on either end) is $\leq \mathbf{0}$ in a basic feasible tableau, then our bfs is optimal (because they represent the negated reduced cost coefficients).

3.3.2 Pivoting and the Simplex Method

The **Simplex Method** is a way to move from one Basic Feasible Tableau to another Basic Feasible Tableau with a few guarantees. One guarantee is that the next BFT is, in fact, a basic feasible tableau as well. The other guarantee is that, as we iterate, the objective function value of the bfs for the basic feasible tableau is always going down or staying constant. Thus, the simplex method is a way of iterating through basic feasible solutions of an (LP), via their respective BFTs, until we find the optimal BFS.

Let's assume we start with some basic feasible tableau, representing a basic feasible solution.

$$InitBFT = \begin{bmatrix} 1 & -\hat{\mathbf{c}}^T & | & z \\ \mathbf{0} & \hat{A} & | & \hat{\mathbf{b}} \end{bmatrix}$$

Algorithm 1: The Simplex Method

Input: $InitBFT$
Output: The optmal bfs, $\bar{\mathbf{x}}$
$CurrBFT \leftarrow InitBFT$;
while *Top row $-\hat{\mathbf{c}}^T$ is not all negative* **do**

> Pick any non-basic column (non-identity column), $\hat{A}^{(j)}$, with top row value $-\hat{\mathbf{c}}_j > 0$;
> **if** $\hat{A}^{(j)} \leq \mathbf{0}$ **then**
>> The problem is unbounded.$^\diamond$ Exit;
>
> $i^* \leftarrow \arg\min_{i \in \hat{A}_{i,j} > 0} \frac{\hat{\mathbf{b}}_i}{\hat{A}_{i,j}}$, (i.e. pick from the rows whose value $\hat{A}_{i,j}$ is positive);
> Row reduce $CurrBFT$ to convert column $\hat{A}^{(j)}$ into the indentity column with a value of $\hat{\mathbf{c}}_j = 0$ above it$^\triangle$;
> $CurrBFT \leftarrow$ new, row-reduced BFT;

$\diamond$: Why? $\mathbf{x}_B = \hat{\mathbf{b}} - \mathbf{x}_j \hat{A}^{(j)} \geq \mathbf{0}$ as $\mathbf{x}_j \to \infty$. So, we can keep reducing our $\mathbf{x}_j$ forever, and we'll always have a feasible solution. Additionally, because the $-\hat{\mathbf{c}}_j^T \geq 0 \implies \mathbf{r}_j^T \leq 0$, then our ofv is either staying the same of decreasing as $\mathbf{x}_j \to \infty$.

$\triangle$: This new tableau is still a basic tableau! We still have the same identity property, we've just changed one of the identity columns.

> **Theorem 3.1**
>
> The result of one iterate of the simplex method is a basic feasible tableau with ofv going down.
>
> *I.e. the new $CurrBFT$ is a basic feasible tableau whose basic feasible solution, $\bar{\mathbf{x}}$, has $\mathbf{ofv}(\bar{\mathbf{x}})$ lower than the previous iterate's bfs.*

Proof. The previous column with identity where $i^* = 1$ will no longer be an identity column. Any other columns with identities will not be touched by the row operations (because those columns have 0 for rows i^*. So the property of containing an identity still holds for our tableau, and we can say the same for the top row. Thus, we still have a basic tableau.

$\forall i \neq i^*$, we now have that our new $\hat{\mathbf{b}}_i = \hat{\mathbf{b}}_i - \frac{\hat{A}_{i,j}}{\hat{A}_{i^*,j}}\hat{\mathbf{b}}_{i^*}$. Note that in this term, we know $\hat{\mathbf{b}}_i, \hat{\mathbf{b}}_{i^*} \geq 0$ and $\hat{A}_{i^*,j} > 0$. Thus, if $\hat{A}_{i,j} \leq 0$, then we know that our new $\hat{\mathbf{b}}_i > 0$. If $\hat{A}_{i,j} > 0$, then we know that $\frac{\hat{\mathbf{b}}_i}{\hat{A}_{i,j}} > \frac{\hat{\mathbf{b}}_{i^*}}{\hat{A}_{i^*,j}}$ and thus $\hat{\mathbf{b}}_i \geq 0$. Therefore, the basic feasible solution for our new basis is feasible,

Lastly, the new $\hat{z} = \hat{z} + \frac{\hat{\mathbf{c}}_j}{\hat{A}_{i^*,j}} \hat{\mathbf{b}}^{i^*}$. Note that $\hat{\mathbf{c}}_j < 0$ (that's why we picked it), $\hat{A}_{i^*,j} > 0$ (again, that's why we picked it), and $\hat{\mathbf{b}}_{i^*} \geq 0$ (by feasibility). This will always decrease if $\hat{\mathbf{b}}_{i^*} > 0$, otherwise we will stall (which is typically okay). ∎

*Note: If $\hat{b}_{i^*} = 0$, then your ofv will not go down for that iteration. This is called **degeneracy** or **stalling**. It is not uncommon to experience stalling while running this algorithm. However, **cycling** is uncommon, and rarely happens.*

> **Theorem 3.2**
>
> If your (LP) is feasible, and the ofv has a lower bound, then the simplex method will terminate, and (LP) has a solution at the bfs represented by the final BFT.

3.3.3 Big-M method

How do we pick the first basis in the algorithm? If we are not given a basis, we can create one. Let the original problem, P, be:

$$\min \mathbf{c}^T \mathbf{x}$$
$$\text{s.t. } A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$$

We can augment P to create an artificial basis. Choose $M >> 0$. Write P^* as:

$$\min \begin{bmatrix} \mathbf{c} \\ \mathbf{M} \end{bmatrix}^T \begin{bmatrix} \mathbf{x} \\ \mathbf{w} \end{bmatrix} \text{ s.t. } \begin{bmatrix} A & | & I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{w} \end{bmatrix} = \mathbf{b},$$
$$\begin{bmatrix} \mathbf{x} \\ \mathbf{w} \end{bmatrix} \geq \mathbf{0}$$

Note that all the feasible solutions for the original LP are contained within the feasible solutions for this new LP, since they are obtained by setting $\mathbf{w} = \mathbf{0}$. Since $M >> 0$, it will never be optimal to have a solution in this new $(LP)^*$ where $\mathbf{w} \neq \mathbf{0}$, and thus we should recover a solution that works equally as well for our original LP. However, importantly, our augmented A matrix already contains an identity in it, which constitutes as our fist basis!

4 The Dual of Linear Programs

We have seen Linear Programs and how to solve them. Now we will look at the *dual problem* of a Linear Program.

Definition: Dual of the Canonical Form

Let $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$. A linear program in canonical form is of the form:

$$\min \mathbf{c}^T \mathbf{x} \quad \text{s.t.} \ A\mathbf{x} \geq \mathbf{b},$$
$$\mathbf{x} \geq \mathbf{0}$$

Then the **dual problem**, DP, takes the form:

$$\max \mathbf{b}^T \mathbf{y} \quad \text{s.t.} \ A^T \mathbf{y} \leq \mathbf{c},$$
$$\mathbf{y} \geq \mathbf{0}$$

Definition: Dual of the Standard Form

Let $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$. A linear program in standard form is of the form:

$$\min \mathbf{c}^T \mathbf{x} \quad \text{s.t.} \ A\mathbf{x} = \mathbf{b},$$
$$\mathbf{x} \geq \mathbf{0}$$

Then the **dual problem**, DP, takes the form:

$$\max \mathbf{b}^T \mathbf{y} \quad \text{s.t.} \ A^T \mathbf{y} \leq \mathbf{c}$$

4.0.1 Equality across forms

As before, with the primary Linear Programming formulation, the 2 forms (canonical and dual) are equivalent to each other.

Suppose we have a linear program, LP, in *canonical form*:

$$\min \mathbf{c}^T \mathbf{x}, \quad \text{s.t.} \ A\mathbf{x} \geq \mathbf{b},$$
$$\mathbf{x} \geq \mathbf{0}$$

It can be converted into a *standard form LP*:

$$\min \begin{bmatrix} \mathbf{c} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}, \quad \text{s.t.} \ \begin{bmatrix} A & | & -I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} = \mathbf{b},$$
$$\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} \geq \mathbf{0}$$

With *standard form* dual problem, DP:

$$\max \mathbf{b}^T \mathbf{y} \quad \text{s.t.} \ \begin{bmatrix} A^T \\ -I \end{bmatrix} \mathbf{y} \leq \begin{bmatrix} \mathbf{c} \\ \mathbf{0} \end{bmatrix}$$

Which is exactly the following DP in *canonical form*:

$$\max \mathbf{b}^T \mathbf{y} \quad \text{s.t.} \ A^T \mathbf{y} \leq \mathbf{c},$$
$$\mathbf{y} \geq \mathbf{0}$$

The other direction is similar: Suppose we are given a linear program, LP, in *standard form*:

$$\min \mathbf{c}^T \mathbf{x} \quad \text{s.t.} \ A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}$$

The *canonical form* of this LP is:

$$\min \mathbf{c}^T \mathbf{x} \quad \text{s.t.} \ \begin{bmatrix} A \\ -A \end{bmatrix} \mathbf{x} \geq \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix},$$
$$\mathbf{x} \geq \mathbf{0}$$

The dual of this problem is:

$$\max \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix}^T \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \text{ s.t. } \begin{bmatrix} A^T & | -A^T \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \le \mathbf{c},$$

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \ge \mathbf{0}$$

Which can be converted into *standard dual form* as:

$$\max \mathbf{b}^T (\mathbf{u} - \mathbf{v}) \text{ s.t. } A^T (\mathbf{u} - \mathbf{v}) \le \mathbf{c},$$

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \ge \mathbf{0}$$

We can simplify this by setting $\mathbf{z} = (\mathbf{u} - \mathbf{v})$:

Note: that the non-negativity constraints disappear for $\mathbf{z}$ because it is the difference of two non-negative vectors

$$\max \mathbf{b}^T \mathbf{z} \text{ s.t. } A^T \mathbf{z} \le \mathbf{c}$$

4.0.2 The dual of the dual

Lastly, note that we can show the dual of the dual is the primal:

$$
\begin{aligned}
(LP) &: \min \mathbf{c}^T \mathbf{x}, \quad \text{s.t. } A\mathbf{x} \ge \mathbf{b}, \ \mathbf{x} \ge \mathbf{0} && \rightarrow \\
(Dual) &: \max \mathbf{b}^T \mathbf{y}, \quad \text{s.t. } A^T \mathbf{y} \le \mathbf{c}, \ \mathbf{y} \ge \mathbf{0} && \rightarrow \\
(Equivalently) &: \min -\mathbf{b}^T \mathbf{y} \text{ s.t. } -A^T \mathbf{y} \ge -\mathbf{c}, \ \mathbf{y} \ge \mathbf{0} && \rightarrow \\
(Dual) &: \max -\mathbf{c}^T \mathbf{z} \text{ s.t. } [-A^T]^T \mathbf{z} \ge -\mathbf{b}, \ \mathbf{z} \ge \mathbf{0} && \rightarrow \\
(Resolving) &: \min \mathbf{c}^T \mathbf{z} \text{ s.t. } A\mathbf{z} \ge \mathbf{b}. \ \mathbf{z} \ge \mathbf{0}
\end{aligned}
$$

The last term here is exactly our original LP.

4.1 Duality

Theorem 4.1: Weak Duality

Let LP be a lienar program, with a dual DP.

If $\bar{\mathbf{x}}$ feasible in LP, and $\bar{\mathbf{y}}$ feasible in DP, then

$$\mathbf{ofv}_{DP}(\bar{\mathbf{y}}) \le \mathbf{ofv}_{LP}(\bar{\mathbf{x}})$$

Weak Duality (standard form).

$$\mathbf{b}^T \bar{\mathbf{y}} = (A\bar{\mathbf{x}})^T \bar{\mathbf{y}} = \bar{\mathbf{x}}^T A^T \bar{\mathbf{y}} \le \bar{\mathbf{x}}^T \mathbf{c} = \mathbf{c}^T \bar{\mathbf{x}}$$

The inequality holds by the feasability of $\bar{\mathbf{y}}$ in the dual, which states that $A^T \mathbf{y} \le \mathbf{c}$. ∎

Weak Duality (canonical form).

$$\mathbf{b}^T \bar{\mathbf{y}} \le (A\bar{\mathbf{x}})^T \mathbf{y} = \bar{\mathbf{x}}^T A^T \bar{\mathbf{y}} \le \bar{\mathbf{x}}^T \mathbf{c} = \mathbf{c}^T \bar{\mathbf{x}}$$

∎

Theorem 4.2: Supervisor Principle

Let LP be a lienar program, with a dual DP.

If $\bar{\mathbf{x}}$ feasible in LP, $\bar{\mathbf{y}}$ feasible in DP, and

$$\mathbf{ofv}_{DP}(\bar{\mathbf{y}}) = \mathbf{ofv}_{LP}(\bar{\mathbf{x}})$$

then $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are optimal in their respective problems.

Proof. By weak duality. ∎

> **Theorem 4.3: Strong Duality**
>
> Let LP be a linear program.
>
> If LP is feasible, and the objective function value is lower bounded then:
> - The LP has a solution $\bar{\mathbf{x}}$
> - The DP has a solution $\bar{\mathbf{y}}$
> - $\mathbf{ofv}_{DP}(\bar{\mathbf{y}}) = \mathbf{ofv}_{LP}(\bar{\mathbf{x}})$.

Proof. Let $A \in \mathbb{R}^{m \times n}$ (WLOG assume it is rank m), and $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$. Let LP be a linear program in standard form:

$$\min \mathbf{c}^T \mathbf{x} \text{ s.t. } A^T \mathbf{x} = \mathbf{b},$$
$$\mathbf{x} \geq \mathbf{0}$$

Assume that LP has been solved to optimality with the simplex method, with the optimal bfs $\bar{\mathbf{x}}$ as the solution, and basis B, implying $A = \begin{bmatrix} B & | & N \end{bmatrix}$. By optimality, $\mathbf{r}_N^T = \mathbf{c}_N^T - \mathbf{c}_B^T B^{-1} N \geq \mathbf{0}$. The dual formulation of this problem is:

$$\max \mathbf{y}^T \mathbf{b} \text{ s.t. } \mathbf{y}^T A \geq \mathbf{c}^T$$

Consider $\bar{\mathbf{y}}^T = \mathbf{c}_B^T B^{-1}$. Note that $\bar{\mathbf{y}}^T \mathbf{b} = (\mathbf{c}_B^T B^{-1}) \mathbf{b} = \mathbf{c}_B^T (B^{-1} \mathbf{b}) = \mathbf{c}^T \bar{\mathbf{x}}$. Thus, if $\bar{\mathbf{y}}$ is feasible, then it has $\mathbf{ofv}_{DP}(\bar{\mathbf{y}}) = \mathbf{ofv}_{LP}(\bar{\mathbf{x}})$ and by the Superisor Principle, $\bar{\mathbf{y}}$ must be optimal. If $\bar{\mathbf{y}}A \leq \mathbf{c}$ then $\bar{\mathbf{y}}$ is feasible. This holds, because

$$\bar{\mathbf{y}}^T A = \mathbf{c}_B^T B^{-1} \begin{bmatrix} B & | & N \end{bmatrix} = \begin{bmatrix} \mathbf{c}_B^T & | & \mathbf{c}_B^T B^{-1} N \end{bmatrix} \leq \begin{bmatrix} \mathbf{c}_B^T & | & \mathbf{c}_N^T \end{bmatrix}$$

Note: The inequality holds because the reduced cost variables show that $\mathbf{c}_N^T \geq \mathbf{c}_B^T B^{-1} N$.

Thus $\bar{\mathbf{y}}$ is feasible with **ofv** equal to $\mathbf{ofv}_{LP}(\bar{\mathbf{x}})$ in the primal, and thus is optimal in DP. ∎

4.1.1 Complementary Slackness

> **Definition: Complementarity**
>
> $$\mathbf{x} = \begin{bmatrix} 63 \\ 0 \\ 0 \\ 9 \\ 0 \\ 2 \end{bmatrix}, \mathbf{y} = \begin{bmatrix} 0 \\ 0 \\ 8 \\ 0 \\ 10 \\ 0 \end{bmatrix}$$
>
> are **complementary**, since the non-zero entries of one correspond to zero entries in the other.
> If $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ s.t. $\mathbf{x}, \mathbf{y} \geq \mathbf{0}$, then $\mathbf{x}$ and $\mathbf{y}$ are complementary iff $\mathbf{x}^T \mathbf{y} = 0$.

> **Theorem 4.4: Standard Form Complementary Slackness**
>
> Let LP be a linear program in standard form, with dual problem DP.
>
> If $\bar{\mathbf{x}}$ feasible in LP, $\bar{\mathbf{y}}$ feasible in DP, then $\bar{\mathbf{x}}$ is optimal in the primal, $\bar{\mathbf{y}}$ is optimal in the dual iff $\bar{\mathbf{x}}$ is complementary to the dual slack of $\bar{\mathbf{y}}$, i.e.
>
> $$\bar{\mathbf{x}}_{\geq 0} \perp (\mathbf{c} - A^T \bar{\mathbf{y}})_{\geq 0}$$

Proof. As in the weak duality proof,
$$\mathbf{b}^T \bar{\mathbf{y}} = \bar{\mathbf{x}} A \bar{\mathbf{y}} \leq \bar{\mathbf{x}}^T \mathbf{c} = \mathbf{c}^T \bar{\mathbf{x}}$$

Note that $\bar{\mathbf{x}}$ is optimal in LP and $\bar{\mathbf{y}}$ is optimal in DP iff $\mathbf{b}^T \bar{\mathbf{y}} = \mathbf{c}^T \bar{\mathbf{x}}$. which is true iff the equality $\bar{\mathbf{x}}^T A \bar{\mathbf{y}} = \bar{\mathbf{x}}^T \mathbf{c}$ holds, which is true iff $\bar{\mathbf{x}}^T (\mathbf{c} - A^T \bar{\mathbf{y}}) = 0$! ∎

> **Theorem 4.5: Canonical Form Complementary Slackness**
>
> Let LP be a linear program in canonical form, with dual problem DP.
>
> If $\bar{\mathbf{x}}$ feasible in LP, $\bar{\mathbf{y}}$ feasible in DP, $\bar{\mathbf{x}}$ is optimal in LP and $\bar{\mathbf{y}}$ is optimal in DP iff $\bar{\mathbf{x}} \perp \mathbf{c} - A^T\bar{\mathbf{y}}$ **and** $\bar{\mathbf{y}} \perp A\bar{\mathbf{x}} - \mathbf{b}$.

Proof.

$$\mathbf{b}^T\bar{\mathbf{y}} \leq (A\bar{\mathbf{x}})^T\bar{\mathbf{y}} = \bar{\mathbf{x}}^T A^T\bar{\mathbf{y}} \leq \bar{\mathbf{x}}^T\mathbf{c} = \mathbf{c}^T\bar{\mathbf{x}}$$

Similar to above, we need our inequalities to become equalities, which happens exactly when

- $\mathbf{b}^T\bar{\mathbf{y}} = (A\bar{\mathbf{x}})^T\mathbf{y} \rightarrow (A\bar{\mathbf{x}} - \mathbf{b})^T\bar{\mathbf{y}} = 0$, and
- $\bar{\mathbf{x}}^T A^T\bar{\mathbf{y}} = \bar{\mathbf{x}}^T\mathbf{c} \rightarrow \bar{\mathbf{x}}^T(\mathbf{c} - A^T\bar{\mathbf{y}}) = 0$

∎

4.2 Duality and the Simplex Tableau

Recall the Simplex Method and Simplex Tableau from Linear Programs (§ 3.3). The same construction can be used in solving the dual problem, as well! Let LP be the following linear program:

$$\min \mathbf{c}^T\mathbf{x} \text{ s.t. } A\mathbf{x} = \mathbf{b},$$
$$\mathbf{x} \geq \mathbf{0}$$

Let B be a basis of LP, with the basic tableau:

$$\begin{bmatrix} 1 & \mathbf{0} & \dots & | & \mathbf{c}_B^T B^{-1}\mathbf{b} \\ \mathbf{0} & I & \dots & | & B^{-1}\mathbf{b} \end{bmatrix}$$

and basic solution $\bar{\mathbf{x}} = \begin{bmatrix} B^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix}$. Recall, additionally, that $\bar{\mathbf{x}}$ is feasible iff $\bar{\mathbf{x}} \geq \mathbf{0}$. Set $\bar{\mathbf{y}}^T = \mathbf{c}_B^T B^{-1} \in \mathbb{R}^m$. Note that $\bar{\mathbf{y}}$ is not necessarily feasible in the dual problem, either. However, our choices of $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ vectors have the same objective function value (by design). This means that a basic tableau for which both $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are feasible has found the optimal solution for both LP and DP in $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$, respectively.

The items needed to a basic tableau to be optimal, with vectors $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$, is:

1. Feasability of $\bar{\mathbf{x}}$:
   (a) $A\mathbf{x} = \mathbf{b}$
   (b) $\bar{\mathbf{x}} \geq \mathbf{0}$.
2. Feasibility of $\bar{\mathbf{y}}$: $\bar{\mathbf{y}}^T A \leq \mathbf{c}^T$
3. Equivalent **ofv**s: $\mathbf{c}^T\bar{\mathbf{x}} = \mathbf{b}^T\bar{\mathbf{y}}$

Notice that 1(a) is *always true!*. If B is a valid basis of A, then it holds that $\begin{bmatrix} B & | & N \end{bmatrix}\begin{bmatrix} B^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix} = \mathbf{b}$. Thus, the choice of $\bar{\mathbf{x}}$ satisfies this constraint always. Additionally, by definition of $\bar{\mathbf{y}}$ and $\bar{\mathbf{x}}$, 4 is always true as well.

Thus, solving the LP comes down to finding a basic tableau which satisfies (1b) and (2). (1b) is true precisely when $B^{-1}\mathbf{b} \geq \mathbf{0}$. (2) is true precisely when the top row of the simplex is $\leq \mathbf{0}$. To see this, first notice that the top row is equal to $-\mathbf{c}^T + \mathbf{c}_B^T B^{-1} A = -\mathbf{c}^T + \bar{\mathbf{y}}^T A$. (You can verify this easily, by just reasoning through the values that are on the top row). Thus, the top row is actually the *negative dual slack* $\mathbf{c}^T - \bar{\mathbf{y}}^T A$. If this slack is $\geq \mathbf{0}$, then $\bar{\mathbf{y}}$ is feasible. Thus if the top row of the tableau is $\leq \mathbf{0}$, then $\bar{\mathbf{y}}$ is feasible in DP.

Note: this also gives us a second way to think about the top row - it is both the rate of change of the ofv as you vary the non-basic variables, and it is the negative of the dual slack.

This gives us 2 paradigms to operate in when solving an LP.

1. If we have a basic tableau which is primal feasible, but not dual feasible, then we can use the primal simplex method to move towards a basis which has dual feasibility, while maintaining primal feasibility.

2. If we have a basic tableau which is dual feasible, but not primal feasible, then we can use the **dual simplex method** to move towards a basis that is primal feasible, while maintaining dual feasibility.

4.2.1 Dual Simplex Method

The *dual simplex method* (Algorithm 2) is very similar to the primal simplex method. Given any basic dual feasible tableau $(-\hat{\mathbf{c}}^T \leq \mathbf{0})$

$$BDFT = \begin{bmatrix} 1 & | & -\hat{\mathbf{c}}^T & | & \hat{z} \\ \mathbf{0} & | & \hat{A} & | & \hat{\mathbf{b}} \end{bmatrix}$$

Algorithm 2: The Dual Simplex Method

Input: $InitBDFT$
Output: The optmal bfs, $\bar{\mathbf{x}}$
$CurrBDFT \leftarrow InitBDFT$;
while $\hat{\mathbf{b}}$ *is not* $\geq \mathbf{0}$ **do**

    $i^* \leftarrow$ some row i s.t. $\hat{\mathbf{b}}_i < 0$;
    if *Row* i^* *of* $\hat{A}$, $\hat{A}_{:,i^*} \geq 0$ **then**
        The the LP is infeasible.$^{\Diamond}$ Exit;

    $j^* \leftarrow \arg\min_{j \in \hat{A}_{i^*,j} < 0} \frac{-\hat{\mathbf{c}}_j}{\hat{A}_{i^*,j}}$, (i.e. pick from the columns, j, whose value $\hat{A}_{i^*,j}$ is negative);
    $CurrBDFT \leftarrow$ Pivot on (i^*, j^*) (similar to simplex method), to obtain a new basic dual feasible tableau with an identity column in column j^*;

$\Diamond$: Recall that $\hat{\mathbf{b}}_{i^*} < 0$. However, if $\hat{A}_{:,i^*} \geq 0$ then $\mathbf{b}$ *cannot* be < 0. Thus the problem is not feasible.

5 Moving beyond linear problems

Most sections from now on will largely *not* make the assumption that the problems are linear.

5.1 Quadratic Programs

Another set of problems that we will be considering often are *Quadratic Programs*. Quadratic programs are often of the following form:

> **Definition: Quadratic Program**
>
> Let $Q \in \mathbb{R}^{n \times n}, \mathbf{c} \in \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^n$, and $S \subseteq \mathbb{R}^n$.
>
> A **Quadratic Program**, QP, is of the form:
> $$\min \mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} \ \text{ s.t. } \ \mathbf{x} \in S$$

Note that the gradient and Hessian of this function $f(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x}$ are:

$$\nabla f(\mathbf{x}) = Q\mathbf{x} + \mathbf{c}$$
$$\nabla^2 f(\mathbf{x}) = Q$$

In general, we are going to consider problems where Q is positive definite or positive semi-definite. If a single eigenvalue of Q is negative, then this problem is, in general, NP-Hard and thus very difficult to solve. Therefore, when we consider quadratic problems, we will often be considering convex or strictly convex quadratics.

5.1.1 Existence and Uniqueness of solutions

Let's consider a less general quadratic program. Let $Q \in \mathbb{R}^{n \times n}$ be symmetric, positive definite. Let $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^n$. Finally let $\mathcal{S} \subseteq \mathbb{R}^n$ be a *closed*, non-empty set. Consider the problem, QP:

$$\min \mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} \ \text{ s.t. } \ \mathbf{x} \in \mathcal{S}$$

> **Proposition 5.1**
>
> All notation as above.
>
> There $\exists$ a global minimizer of QP.
> Furthermore, if $\mathcal{S}$ is also convex, then the minimizer is unique.

Note: This last statement cannot be taken for granted. For example, the set $\mathcal{S} = \{\mathbf{x} : f(\mathbf{x}) = 35\}$ is a closed, non-empty set. However, every point of $\mathcal{S}$ is a global minima!.

Note: This problem definition covers a lot of problems that we're interested in. For example, if $\mathcal{S}$ is a polyhedral set, then $\mathcal{S}$ is closed and convex! Or if $\mathcal{S}$ is defined by a set of functions $g_i \leq 0$ where all g_i are convex and continuous then $\mathcal{S}$ is again a closed, convex set.

Proof. Define $f(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x}$. For any $\mathbf{d} \in \mathbb{R}^n$ s.t. $||\mathbf{d}||_2 = 1$, and for any $\alpha \in \mathbb{R}^n_{\geq 0}$ let $\bar{\mathbf{x}} = \alpha * \mathbf{d}$.

$$\begin{aligned} f(\bar{\mathbf{x}}) &= \bar{\mathbf{x}}^T Q \bar{\mathbf{x}} + \mathbf{c}^T \bar{\mathbf{x}} \\ &= \alpha^2 \mathbf{d}^T Q \mathbf{d} + \alpha \mathbf{c}^T \mathbf{d} \\ &\geq^{\diamond} \left(\min_{\lambda \in \sigma(Q)} \lambda \right) \alpha^2 - ||\mathbf{c}||_2 \alpha \end{aligned}$$

$\diamond$: *The first term of the inequality is a special property of symmetric matrices and unit vectors. This value is bounded by the max and min eigenvalues of Q, which in this case are always > 0, because Q is P.D. The second term is due to the fact that $|\mathbf{c}^T \mathbf{d}| \leq ||\mathbf{c}|| * ||\mathbf{d}|| = ||\mathbf{c}||$ and thus the minimum value of $\mathbf{c}^T \mathbf{d} = -||\mathbf{c}||$.*

Note that this bound scales only by α. All other terms are constant. Additionally, the bound is dominated by $\alpha^2 * \min \lambda$, which is positive. Thus, for any arbitrary $\mathbf{x}_0 \in \mathcal{S}$, $\exists \alpha > 0$ s.t. $\forall ||\mathbf{x}||_2 > \alpha\, f(\mathbf{x}) > f(\mathbf{x}_0)$. Then define

$$B = \{\mathbf{x} \in \mathbb{R}^n : ||\mathbf{x}||_2 \le \alpha\}$$

Note that is B is closed and thus $B \cap \mathcal{S}$ is closed. B is bounded, and thus $B \cap \mathcal{S}$ is bounded. Therefore, $B \cap \mathcal{S}$ is a compact, non-empty set. We know that it's non-empty because, at the very least, $\mathbf{x}_0$ is in it!

Because B is compact and non-empty, we know that there must exist a global min of f on B, $\mathbf{x}^*$. Therefore, $\mathbf{x}^*$ is a global min of f on $\mathcal{S}$, because $\forall \mathbf{x} \notin B, f(\mathbf{x}) > f(\mathbf{x}_0) \ge f(\mathbf{x}^*)$. Thus a global min exists for f on $\mathcal{S}$.

Finally, assume $\mathcal{S}$ is a convex set. Suppose $\exists$ two global mins, $\mathbf{x}'$ and $\mathbf{x}''$ for f on $\mathcal{S}$. Let

$$D = \{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \le f(\mathbf{x}')\}$$

Note that $D \cap \mathcal{S}$ is also a convex set, and thus the line segment $[\mathbf{x}', \mathbf{x}''] \in D \cap \mathcal{S}$. Thus f is constant along the line segment, which is a constradiction as f is strictly convex. Therefore, the global min of f on $\mathcal{S}$ is unique.

$\blacksquare$

5.2 Frank-Wolfe Algorithm

This section will introduce one very simple algorithm for approaching general constrained optimization problems.

Let's consider a very general problem, P:

$$\min f(\mathbf{x}) \text{ s.t. } \mathbf{x} \in S$$

where

- f is cont. diff.
- S is a polyhedreal set, e.g. $S = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \ge \mathbf{b}, \mathbf{x} \ge \mathbf{0}\}$, $S = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}, \mathbf{x} \ge \mathbf{0}\}$, $S = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}\}$

The Frank-Wolfe Algorithm offers a very general solution to P. *Note: This is basically choosing which direction to step based*

Algorithm 3: Frank-Wolfe Algorithm

Input: Problem (P)
Output: Solution $\mathbf{x}^{(k)}$
$\mathbf{x}^{(0)} \leftarrow$ some initial guess ;
for $i = 0, 1, \dots$ *until a stopping criterion* **do**
    $\mathbf{c} \leftarrow \nabla f(\mathbf{x}^{(i)})$;
    Define $LP^{(i)} : \min \mathbf{c}^T \mathbf{x}$ s.t. $\mathbf{x} \in S$;
    $\mathbf{z}^{(i)} \leftarrow$ solution to $LP^{(i)}$;
    Define $P^{(i)} : \min_{\alpha \in [0,1]} f(\alpha \mathbf{z}^{(i)} + (1 - \alpha)\mathbf{x}^{(i)})$;
    $\alpha^{(i)} \leftarrow$ the solution to $P^{(i)}$;
    $\mathbf{x}^{(i+1} \leftarrow \alpha^{(i)}\mathbf{z}^{(i)} + (1 - \alpha^{(i)})\mathbf{x}^{(i)}$;

---

on the "descent" level of the immediate gradient, as well as the boundaries of your feasible region S. So you choose the most "promising" direction based on how far you can go and how much the gradient says you'll decrease in that direction.

Note a few important things about this algorithm

- The problem $LP^{(i)}$ is a linear program which we've now seen how to solve now.
- The problem $P^{(i)}$ is a 1-dim optimization problem, that can be easily solved, either analytically or numerically.
- At $\mathbf{x}^{(i)}$, the first-order approxation of f,

$$f(\mathbf{x}) \approx g(\mathbf{x}) = f(\mathbf{x}^{(i)}) + \nabla f(\mathbf{x}^{(i)})^T (\mathbf{x} - \mathbf{x}^{(i)})$$
$$= \left(f(\mathbf{x}^{(i)}) - \nabla f(\mathbf{x}^{(i)})\mathbf{x}^{(i)} \right) + \nabla f(\mathbf{x}^{(i)})^T \mathbf{x}$$
$$= \text{constant} + \mathbf{c}^T \mathbf{x}$$

Thus, $LP^{(i)}$ is equivalent to minimizing $g(\mathbf{x})$, which is like saying what $\mathbf{x}$ around $\mathbf{x}^{(i)}$ is minimal.

6 The KKT Conditions

The KKT Conditions give us an important way to not only *check* points for optimality, but also to *solve exactly* for optimal points in some scenarios. This section will derive the KKT conditions for general constrained optimization problems, and then present them explicitly in a few examples.

6.1 Farkas' and Gordon's Theorems

In this section we will introduce 2 important theorems which will lead to the derivation of the KKT conditions.

Theorem 6.1: Farkas

Let $A \in \mathbb{R}^{n \times m}, \mathbf{b} \in \mathbb{R}^m$.

Then *either*
(1) $\exists \mathbf{x} \in \mathbb{R}^n$ s.t. $A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$
    or
(2) $\exists \mathbf{y} \in \mathbb{R}^m$ s.t. $\mathbf{b}^T \mathbf{y} > 0, A^T \mathbf{y} \leq \mathbf{0}$
One of these must hold, and the other must be false, for any $A, \mathbf{b}$

Proof. Consider the problem and it's dual:

$$LP : \min \mathbf{0}^T \mathbf{x} \text{ s.t. } A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$$
$$DP : \max \mathbf{b}^T \mathbf{y} \text{ s.t. } A^T \mathbf{y} \leq \mathbf{0}$$

If (1) is true, then LP is feasible and the **ofv** $= 0$. Then, by weak duality, DP has no solution with **ofv** > 0. Then (2) is false.

If (2) is false, then since DP is feasible with vector $\mathbf{y} = \mathbf{0}$, with **ofv** $= 0$, we know that this is the max ofv of the DP (by (2) being false). Since the DP has a solution, then LP must also have a solution and be feasible (with **ofv** $= 0$), and thus (1) is true.

∎

Theorem 6.2: Gordon

Let $A \in \mathbb{R}^{n \times m}$.

Then *either*:
(1) $\exists \mathbf{x} \in \mathbb{R}^n$ s.t. $A\mathbf{x} = \mathbf{0}, \mathbf{x} \geq \mathbf{0}, \mathbf{x} \neq \mathbf{0}$, or
(2) $\exists \mathbf{y} \in \mathbb{R}^m$ s.t. $A^T \mathbf{y} < \mathbf{0}$

Proof. (2) being true $\iff \exists \mathbf{y} \in \mathbb{R}^m, \epsilon > 0$ s.t. $A^T \mathbf{y} + \epsilon \mathbf{1} \leq \mathbf{0}$. We can rewrite this statement in matrix form:

$$\exists \begin{bmatrix} \mathbf{y} \\ \epsilon \end{bmatrix} \in \mathbb{R}^{m+1} \text{ s.t. } \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \epsilon \end{bmatrix} > 0, \begin{bmatrix} A^T & | & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \epsilon \end{bmatrix} \leq \mathbf{0}$$

Note that this re-writing looks a lot like Farkas' setup, where $A^T = \begin{bmatrix} A^T & | & \mathbf{1} \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$. Thus, we can say that (2) is true iff

$$\nexists \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \geq \mathbf{0}, \text{ s.t. } \begin{bmatrix} A \\ \mathbf{1}^T \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$$

which we can restate as $\nexists \mathbf{x} \in \mathbb{R}^n$ s.t. $A\mathbf{x} = \mathbf{0}, \mathbf{x} \geq \mathbf{0}$ and the coordinates of $\mathbf{x}$ sum to one, which is true iff $\mathbf{x} \neq \mathbf{0}$ (since we can arbitrarily scale $\mathbf{x}$ until it's components summed to one without affecting the homogenous solution) which is when (1) is false. Thus (2) is true iff (1) is false.

∎

6.2   Deriving the KKT Conditions

We will derive the KKT conditions by making a slow march through a series of propositions.

Let's start by considering the problem P, defined as:

$$\min f(x) \text{ s.t. } g_1(x) \le 0,$$
$$g_2(x) \le 0,$$
$$\vdots$$
$$g_m(x) \le 0,$$
$$\mathbf{x} \in S$$

where we write $\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_m(\mathbf{x}) \end{bmatrix}$ s.t. $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$, with constraints $\mathbf{g}(\mathbf{x}) \le \mathbf{0}$.

Definition: Active Constraints

For any $\mathbf{x} \in S$, feasible in P, the **active constaints** are

$$\mathcal{A}_{\mathbf{x}} = \{i : g_i(\mathbf{x}) = 0\}$$

In other words, they are the constraints for which $\mathbf{x}$ is on the "boundary".

Proposition 6.1: KKT Prop 1

If $\mathbf{x} \in S$, feasible (satisfies $\mathbf{g}$), and is a *local min* of P, then there does not exist a direction $\mathbf{d} \in \mathbb{R}^n$ which is simultaneously a descent direction for f and all active constraints, g_i s.t. $i \in \mathcal{A}_{\mathbf{x}}$.

Proof. If such a $\mathbf{d}$ exists then $\exists \, \alpha > 0$ small enough such that

$$\forall i \in \mathcal{A}_{\mathbf{x}}, g_i(\mathbf{x} + \alpha \mathbf{d}) < g_i(\mathbf{x}) = 0$$
$$\forall i \notin \mathcal{A}_{\mathbf{x}}, g_i(\mathbf{x} + \alpha \mathbf{d}) \approx g_i(\mathbf{x}) < 0$$

thus $\mathbf{x} + \alpha \mathbf{d}$ is feasible, and $f(\mathbf{x} + \alpha \mathbf{d}) < f(\mathbf{x})$ and thus $\mathbf{x}$ is not a local min.   ∎

Proposition 6.2: KKT Prop 2

If $\mathbf{x} \in S$ is feasible and a local min of P then $\nexists \, \mathbf{d} \in \mathbb{R}^n$ (direction, $\ne \mathbf{0}$), s.t. $J^T \mathbf{d} < 0$, where $J^T = \begin{bmatrix} \nabla f^T(x) \\ \nabla \mathbf{g}_{\mathcal{A}_{\mathbf{x}}}^T(\mathbf{x}) \end{bmatrix}$

Proof. Immediate consequence of prop 1, J represents the gradients of f and all active constraints $\mathcal{A}_{\mathbf{x}}$, and if such a d exists, then it is a descent direction for both f and all active constraints $\mathcal{A}_{\mathbf{x}}$.   ∎

Proposition 6.3: KKT Prop 3

If $\mathbf{x} \in S$, feasible, is a local min of P, then $\exists \, \vec{\lambda} \ge \mathbf{0}, \vec{\lambda} \ne \mathbf{0}$, s.t. $J \vec{\lambda} = \mathbf{0}$

Proof. This is equivalent to Prop 2, by Gordon's Theorem.   ∎

> **Proposition 6.4: KKT Prop 4**
>
> If $\exists \mathbf{x} \in S$, feasible, is a local min of P, then $\exists \begin{bmatrix} \beta \\ \vec{\lambda} \end{bmatrix} \in \mathbb{R}^{1+m}$, non-zero, such that
>
> $$\begin{bmatrix} \nabla f(\mathbf{x}) & \nabla g_1(\mathbf{x}) & ... & \nabla g_m(\mathbf{x}) \end{bmatrix} \begin{bmatrix} \beta \\ \vec{\lambda} \end{bmatrix} = \mathbf{0}, \ \text{ s.t. } \ \forall i = 1, ..., m \ \vec{\lambda}_i g_i(\mathbf{x}) = 0$$

Note that this is just a restating of Prop 3 where we include *all* constraints here (not just active ones), but if a constraint is not active (say j is not active), then $\vec{\lambda}_j$ must be zero.

Fritz-John is a re-statement of Prop 4, but utilizing the definition of the *Jacobian*:

> **Definition: The Jacobian**
>
> The **Jacobian** of $\mathbf{g}$ is defined as
>
> $$\nabla \mathbf{g}(\mathbf{x}) = \begin{bmatrix} \nabla g_1(\mathbf{x}) & | & \nabla g_2(\mathbf{x}) & | & ... & | & \nabla g_m(\mathbf{x}) \end{bmatrix}$$

Using the above definition and notes, we can rewrite Prop 4 as the Fritz-John Optimality Conditions:

> **Theorem 6.3: Fritz-John Optimality Conditions**
>
> If $\mathbf{x} \in S$, feasible in P, is a local min of P, then $\exists \vec{\lambda} \in \mathbb{R}^m, \beta \in \mathbb{R}$ s.t.
>
> $$\beta \nabla f(\mathbf{x}) + \nabla \mathbf{g}(\mathbf{x}) \vec{\lambda} = \mathbf{0},$$
> $$\vec{\lambda}^T \mathbf{g}(x) = 0,$$
> $$\text{and } \begin{bmatrix} \beta \\ \vec{\lambda} \end{bmatrix} \geq \mathbf{0}, \neq \mathbf{0}$$
>
> Which we can similarly write as...
>
> $$\beta \nabla f(\mathbf{x}) + \sum_{i=1}^{m} \nabla \mathbf{g}_i(\mathbf{x}) \vec{\lambda}_i = \mathbf{0},$$
> $$\sum_{i=1}^{m} \vec{\lambda}_i \mathbf{g}_i(x) = 0,$$
> $$\text{and } \beta, \vec{\lambda}_1, \vec{\lambda}_2, ..., \vec{\lambda}_m \geq 0, \text{but not all } 0$$

Again, there is no proof. This is just a re-statement of proposition 4, using the definition of a Jacobian. Finally, this leads us to the KKT conditions...

> **Theorem 6.4: Karush-Kuhn-Tucker Conditions**
>
> If $\mathbf{x} \in S$, feasible in P is a local min of P and satisfies the "constraint qualifications" (cq) that $\{\nabla g_i\}_{i \in \mathcal{A}_{\mathbf{x}}}$ are linearly independent, then
>
> $$\exists \vec{\lambda} \in \mathbb{R}^m \text{ s.t. } \nabla f(\mathbf{x}) + \nabla \mathbf{g}(\mathbf{x}) \vec{\lambda} = \mathbf{0},$$
> $$\vec{\lambda}^T \mathbf{g}(x) = 0,$$
> $$\text{and } \vec{\lambda} \geq \mathbf{0}$$

Proof. This follows from Fritz-John. The linear independence of $\{\nabla \mathbf{g}_i\}_{i \in \mathcal{A}_{\mathbf{x}}}$ forbids β from being 0. Thus, we can divide

$\begin{bmatrix} \beta \\ \overrightarrow{\lambda} \end{bmatrix}/\beta$ which sets $\beta = 1$, and thus we only require that $\overrightarrow{\lambda} \geq \mathbf{0}$ (since $\beta > 0$, the "but not all zero" constraint is already satisfied). The $\overrightarrow{\lambda}_i$ are called the KKT multipliers. ∎

6.3 Using the KKT Conditions

We will now examine how our KKT conditions can apply to some problems that we've seen before, and how it can aid us in solving these problems.

As a general formulation, define a problem P, as

$$\min f(x) \text{ s.t. } \mathbf{g}(x) \leq \mathbf{0}, \mathbf{x} \in S$$

where S is some open set, and f and all of our g_i's are continuously differentiable. By KKT, if $\mathbf{x}$ is feasible, and is a local min of P, and our constraint qualification (cq) is satisfied, then we have

$$\nabla f(\mathbf{x}) + \nabla \mathbf{g}(\mathbf{x})\overrightarrow{\lambda} = \mathbf{0},$$
$$\overrightarrow{\lambda}^T \mathbf{g}(\mathbf{x}) = 0,$$
$$\overrightarrow{\lambda} \geq \mathbf{0}$$

Note: The constraint qualification that we used in the definition was that our active constraint gradients were linearly independent. However, there are many types of constraint qualifications that we could use to satisfy the definition.

Now, we are interested in using these KKT conditions to *find* optimality, rather than just *verify* it. To see an example of this, let's revisit a simple linear program.

6.3.1 KKT Conditions for a Linear Program

Let $A \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m, \mathbf{c} \in \mathbb{R}^n$. Lets define the following linear program P, with variables $\mathbf{x} \in \mathbb{R}^n$:

$$\min \mathbf{c}^T \mathbf{x} \text{ s.t. } A\mathbf{x} \geq \mathbf{b},$$
$$\mathbf{x} \geq \mathbf{0}$$

We can rewrite P as:

$$\min \mathbf{c}^T \mathbf{x} \text{ s.t. } \begin{bmatrix} -A \\ -I \end{bmatrix}\mathbf{x} + \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \leq \mathbf{0},$$
$$\mathbf{x} \in \mathbb{R}^n$$

This is now in the form we want! We have our $\mathbf{g}$ and our f for our KKT conditions, and our feasible region $S = \mathbb{R}^n$, which is a nice, open set. So suppose now that we have some local min $\bar{\mathbf{x}}$. What do the KKT conditions say about this point?

1. Firstly, we know what $\bar{\mathbf{x}}$ is feasible:
$$A\bar{\mathbf{x}} \geq \mathbf{b},$$
$$\bar{\mathbf{x}} \geq \mathbf{0}$$

2. We also have that our KKT multipliers are ≥ 0:
$$\overrightarrow{\lambda} = \begin{bmatrix} \bar{\mathbf{y}} \\ \bar{\mathbf{z}} \end{bmatrix} \geq \mathbf{0}, \bar{\mathbf{y}} \in \mathbb{R}^m, \bar{\mathbf{z}} \in \mathbb{R}^n$$

3. We have the complimentarity of our $\mathbf{g}$ and $\overrightarrow{\lambda}$
$$\begin{bmatrix} \bar{\mathbf{y}} \\ \bar{\mathbf{z}} \end{bmatrix}^T \begin{bmatrix} \mathbf{b} - A\bar{\mathbf{x}} \\ -\bar{\mathbf{x}} \end{bmatrix} = 0$$

4. and lastly, we have the gradient constraints of KKT:
$$\mathbf{c} + \begin{bmatrix} -A^T & | & -I \end{bmatrix}\begin{bmatrix} \bar{\mathbf{y}} \\ \bar{\mathbf{z}} \end{bmatrix} = \mathbf{0}$$

From (2), we know that $\bar{\mathbf{y}}, \bar{\mathbf{z}} \geq \mathbf{0}$. From (3), we also see that $\bar{\mathbf{y}}^T(\mathbf{b} - A\mathbf{x}) = 0$ and that $\bar{\mathbf{z}}^T\bar{\mathbf{x}} = 0$. Finally, from (4) we see that $A^T\bar{\mathbf{y}} + \bar{\mathbf{z}} = \mathbf{c}$, which we can rewrite as $\bar{\mathbf{z}} = \mathbf{c} - A^T\bar{\mathbf{y}}$. We have a lot of information about $\bar{\mathbf{z}}$! We can use this to write it out of the equation:

1. Feasibility, same as before

$$Ax \geq \mathbf{b},$$
$$\mathbf{x} \geq \mathbf{0}$$

2. We substitute $\bar{\mathbf{z}} = \mathbf{x} - A^T\bar{\mathbf{y}}$ in, to get:

$$\begin{bmatrix} \bar{\mathbf{y}} \\ \mathbf{c} - A^T\bar{\mathbf{y}} \end{bmatrix} \geq \mathbf{0} \implies A^T\mathbf{y} \leq \mathbf{c},\ \mathbf{y} \geq \mathbf{0}$$

which is *exactly* dual feasibility of $\bar{\mathbf{y}}$!

3. and lastly, subbing it into the complimentarity condition, we get:

$$\bar{\mathbf{y}}^T \perp A\bar{\mathbf{x}} - \mathbf{b} \text{ and } \bar{\mathbf{x}} \perp \mathbf{c} - A^T\bar{\mathbf{y}}$$

which is exactly our LP complementary slackness!

4. the 4th condition gets written into all the other conditions, implicitly, by our substitution.

This is a translation for what the KKT conditions mean for us in a linear program! They are *exactly* the necessary and sufficient conditions for optimality in a LP! Primal and Dual feasibility, and complementary slackness.

6.3.2 KKT Conditions for Quadratic Programs with Equalities

Let's start with a simple quadratic problem formulation. Suppose $Q \in \mathbb{R}^{n \times n}$ is symmetric, P.D., $A \in \mathbb{R}^{m \times n}, \mathbf{c} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m$. Define the quadratic program, QP as:

$$\min \frac{1}{2}\mathbf{x}^T Q\mathbf{x} - \mathbf{c}^T\mathbf{x}, \text{ s.t. } A\mathbf{x} = \mathbf{b},$$
$$\mathbf{x} \in \mathbb{R}^n$$

We can rewrite the constraints into "KKT-friendly" form...

$$\min \frac{1}{2}\mathbf{x}^T Q\mathbf{x} - \mathbf{c}^T\mathbf{x}, \text{ s.t. } A\mathbf{x} - \mathbf{b} = \mathbf{0},$$
$$\mathbf{x} \in \mathbb{R}^n$$

where now we can say that $\nabla f(\mathbf{x}) = Q\mathbf{x} - \mathbf{c}$, and $\nabla \mathbf{g}(\mathbf{x}) = A^T$.

Note: here that we have equalities, where we typically require inequalities for KKT. We can easily convert this into inequalities, by doing the $\leq, \geq$ trick. This actually removes positive constraints on our $\overrightarrow{\lambda}$ vector, since you have the subtraction of two positives in the eventual formula. Thus, our KKT conditions are actually simplified. Additionally, we lose the complimentarity constraints, since the constraints are always active.

Thus, the KKT conditions when $\mathbf{x}$ is feasible ($A\mathbf{x} = \mathbf{b}$) and is a local min of QP are:

$$Q\mathbf{x} - \mathbf{c} + A^T\mathbf{y} = \mathbf{0}$$

i.e. $\mathbf{x} \in \mathbb{R}^n$ is a solution iff $\exists\, \mathbf{y} \in \mathbb{R}^m$ s.t. $\begin{bmatrix} Q & A^T \\ A & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ \mathbf{b} \end{bmatrix}$.

This big matrix enforces a few things

- $Q\mathbf{x} + A^T\mathbf{y} = c$ (KKT condition)

- $A\mathbf{x} = \mathbf{b}$ (primal feasibility)

and that's enough to find a solution! Additionally, this big matrix (M) is square, and in fact invertible, since Q is P.D and we can always make A full row-rank. So we can actually solve this exactly! Which gives us an $\mathbf{x}$ and $\mathbf{y}$ exactly! And since this is a convex QP, the solution is also a global min.

6.3.3   A bit on Quadratic Programs with Inequalities

Let's look at an example of a QP with inequality constraints. Let $A \in \mathbb{R}^{n \times n}$ symmetric PD, $\mathbf{b} \in \mathbb{R}^m$ non-zero, $\mathbf{c} \in \mathbb{R}^n_{>0}$:

$$\min \frac{1}{2} \mathbf{x}^T A \mathbf{x}, \quad \text{s.t. } \mathbf{b}^T \mathbf{x} + \mathbf{c} \leq 0$$

Note that our f is convex (yay) and our g (we only have one!) is affine (yay). Moreover, the gradients are:

$$\nabla f(\mathbf{x}) = A\mathbf{x}, \nabla g(\mathbf{x}) = \mathbf{b} \neq \mathbf{0}$$

Let's solve this problem by looking for KKT points. Since f is convex, then the KKT points are globally optimal! But, we have to consider a couple cases. If g is active for a KKT point, or if g is not active. Let's consider a KKT point where the constraint g is not active. Then KKT *only* tells us that

$$\nabla f(\mathbf{x}) = A\mathbf{x} = \mathbf{0} \implies \mathbf{x} = \mathbf{0}$$

If none of our constraints our active, then the only thing KKT has to tell us is that an optimal point is a stationary point, which happens only when the gradient is 0. However... $\mathbf{x} = \mathbf{0}$ is not feasible! And so, we *must* have an active constraint for our KKT points for this problem. So, let's look at when g is active, then. In this case, KKT gives us that

$$A\mathbf{x} + \lambda \mathbf{b} = \mathbf{0}, \ \lambda \geq 0, \ \mathbf{b}^T \mathbf{x} + c = 0$$

Well, we can solve this for $\mathbf{x}$! A is invertible, since it's PD, and thus we can write this out:

$$\mathbf{x} = A^{-1}(-\lambda \mathbf{b})$$
$$\mathbf{b}^T A^{-1}(-\lambda \mathbf{b}) + \mathbf{c} = \mathbf{0}$$
$$\lambda = \frac{\mathbf{c}}{\mathbf{b}^T A^{-1} \mathbf{b}} \qquad \text{which is greater than 0, since A and } A^{-1} \text{ is P.D.}$$
$$\bar{\mathbf{x}} = \frac{-c}{\mathbf{b}^T A \mathbf{b}} A^{-1} \mathbf{b}$$

which gives us a unique solution for $\mathbf{x}$! Thus... KKT has given us an exact solution for this QP with a single inequality constraint.

7 Newton's Method and Interior Point Methodology

7.1 Newton's Method

Suppose we have a component-wise function $F : \mathbb{R}^n \to \mathbb{R}^n$, component-wise continuously differentiable. Suppose, additionally, that we seek a "zero" or "root" of F, $\mathbf{x}^* \in \mathbb{R}^n$ s.t. $F(\mathbf{x}^*) = \mathbf{0}$. Then **Newton's Method** gives us a way to solve for this $\mathbf{x}^*$.

Note: This comes up a lot in, say, unconstrained optimization, where we seek a point of a function where it's gradient is zero. Well, the gradient is a function from $\mathbb{R}^n \to \mathbb{R}^n$, so it fits into this design.

Newton's method is outlined in Algorithm 4.

Algorithm 4: Newton's Method

Input: $F : \mathbb{R}^n \to \mathbb{R}^n$
Output: $\mathbf{x}^*$ s.t. $F(\mathbf{x}^*) \approx \mathbf{0}$
$k \leftarrow 0$;
$\mathbf{x}^{(0)} \leftarrow$ some initial guess;
while $F(\mathbf{x}^{(k)}) \not\approx \mathbf{0}$ **do**
    Define $G^{(k)} : \mathbb{R}^n \to \mathbb{R}^n$ to be a linear approximation of F about $\mathbf{x}^{(k)}$;
    $\mathbf{x}^{(k+1)}$ is defined s.t. $G^{(k)}(\mathbf{x}^{(k+1)}) = \mathbf{0}$;
    $k \leftarrow k + 1$

F is a component-wise function, of the form $F = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{bmatrix}$, and we require that $\forall i \ F_i : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable.

The G function is meant to be a linear approximation of F about $\mathbf{x}^{(k)}$. We know that we can use Taylor Series to approximate this function. In particular, we'll use the first-order approximation, since we know that F_i is continuously differentiable, and the approximation is linear. So, let's define

$$G_i(\mathbf{x}) = F_i(\mathbf{x}^{(k)}) + \nabla F_i^T(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}) \approx F_i(\mathbf{x})$$

We can stack all of these up, such that:

$$G(\mathbf{x}) = F(\mathbf{x}^{(k)}) + \begin{bmatrix} \nabla F_1^T(\mathbf{x}^{(k)}) \\ \nabla F_2^T(\mathbf{x}^{(k)}) \\ \vdots \\ \nabla F_n^T(\mathbf{x}^{(k)}) \end{bmatrix} (\mathbf{x} - \mathbf{x}^{(k)}) \approx F(\mathbf{x})$$

where we refer to the big matrix of gradients as the Jacobian of F, $\nabla F^T(\mathbf{x}^{(k)})$.

7.1.1 The Newton Direction

In the inner loop of Newton's method we want to find $\mathbf{x}^{(k+1)}$ to be the vector such that $\mathbf{0} = F(\mathbf{x}^{(k)}) + \nabla F^T(\mathbf{x}^{(k)})(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$, i.e. we want

$$-F(\mathbf{x}^{(k)}) = \nabla F^T(\mathbf{x}^{(k)})(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$$

Note here that the, given the previous iterates $\mathbf{x}^{(k)}$, the Jacobian is a fixed matrix in $\mathbb{R}^{n \times n}$, and $-F(\mathbf{x}^{(k)})$ is a fixed vector in $\mathbb{R}^n$, so we only need to solve for $(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$, which we can do by simply solving the linear system.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left(\nabla F(\mathbf{x}^{(k)})^T\right)^{-1} F(\mathbf{x}^{(k)})$$

Note: Normally, we would just solve via computational methods, rather than actually trying to invert the matrix for an analytical solution, since inverting the matrix is often very expensive.

> **Definition: Newton Direction**
>
> All notation as above.
>
> For any given iterate $k + 1$, with solution
>
> $$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left(\nabla F(\mathbf{x}^{(k)})^T \right)^{-1} F(\mathbf{x}^{(k)})$$
>
> The **Newton Direction**,
>
> $$-\left(\nabla F(\mathbf{x}^{(k)})^T \right)^{-1} F(\mathbf{x}^{(k)})$$
>
> is the direction moved away from the last iterates solution, $\mathbf{x}^{(k)}$

7.1.2 An application of Newton's Method: 2nd order optimization

Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable, and you are looking for the min or max of f. In this case, you are looking for a stationary point of f, i.e. an $\mathbf{x}^* \in \mathbb{R}^n$ s.t. $\nabla f(\mathbf{x}^*) = \mathbf{0}$. Notice that $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$. And note that the Jacobian of ∇f is exactly $\nabla^2 f$, the Hessian!

If we plug in our analytical solution at step i, then we have that

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \left(\nabla^2 f(\mathbf{x}^{(i)}) \right)^{-1} \nabla f(\mathbf{x}^{(i)})$$

which gives us a way to optimize towards the min or max of a function using it's Hessian.

7.2 Interior Point Methodology: Solving a Quadratic Program with Inequalities

In this section, we'll see how to tackle a QP with inequality constraints using a modified version of Newton's Method, called Interior Point Methodology.

Suppose $Q \in \mathbb{R}^{n \times n}$ is sym, PD., and $A \in \mathbb{R}^{m \times n}$ is full row rank, $\mathbf{b} \in \mathbb{R}^m, \mathbf{c} \in \mathbb{R}^n$. Let's define the quadratic program QP, in standard form, as:

$$\min \frac{1}{2} \mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} \ \text{ s.t. } A\mathbf{x} = \mathbf{b},$$
$$\mathbf{x} \geq \mathbf{0}$$

If Q had negative eigenvalues, this problem would be NP-hard. Additionally, note that Q being PD means that the hessian is PD as well, and thus our function is strictly convex. Finally, note that A being full-row rank is without loss of generality, since otherwise we could row-reduce to make it so.

Let's start by re-writing our constraints as

$$A\mathbf{x} - \mathbf{b} = \mathbf{0}$$
$$-I\mathbf{x} \leq \mathbf{0}$$

We already know that, since our function is convex, if our feasible region is closed, then there must exist a global min. Additionally, we know that if the feasible region is convex, then the global min is unique. In our case, the feasible region is closed, *and convex*, since it's a polyhedral set. Thus, in this problem we know that we have a unique global min. Finally, since A is full row-rank, this satisfies a constraint qualification. This means that a KKT point is *necessary* for a global min. We will see later that, in fact, the KKT conditions are sufficient for global optimality, in this problem.

7.2.1 KKT Conditions of our problem

The KKT conditions of this QP look like:

- $Q\mathbf{x} + \mathbf{c} + \begin{bmatrix} A^T & | & -I \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \mathbf{0}$

- $\mathbf{z} \geq \mathbf{0}$

- $\mathbf{z}^T\mathbf{x} = 0$

Note: We've broken up our KKT multipliers such that $\mathbf{y} \in \mathbb{R}^m$ correspond to the m constraints, and $\mathbf{z} \in \mathbb{R}^n$ correspond to the negative identity. Additionally, we have greatly simplified the conditions on $\mathbf{y}$ as explained in § 6.3.2

Now, we can compose **both** our primal feasibility constraints and our KKT conditions into one big function $F : \mathbb{R}^{n+m+n} \to \mathbb{R}^{n+m+n}$ as

$$F\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix}\right) = \begin{bmatrix} Q\mathbf{x} + \mathbf{c} + A^T\mathbf{x} - \mathbf{z} \\ A\mathbf{x} - \mathbf{b} \\ \mathbf{z} \circ \mathbf{x} \end{bmatrix}$$

Now, observe that $\mathbf{x}^*$ is feasible and a KKT point with KKT multipliers $\mathbf{y}^*, \mathbf{z}^*$ iff

$$F\left(\begin{bmatrix} \mathbf{x}^* \\ \mathbf{y}^* \\ \mathbf{z}^* \end{bmatrix}\right) = \mathbf{0}, \ \mathbf{x}^* \geq \mathbf{0}, \text{ and } \mathbf{z}^* \geq \mathbf{0}$$

The constraint $A\mathbf{x}^* = \mathbf{b}$ is covered by the "middle chunk" of F, and $\mathbf{x} \geq \mathbf{0}$ is just copied over. So we have primal feasibility, above. Additionally, the first KKT condition is covered by the "top chunk" of F. The second condition is covered by the "bottom chunk" of F (since every element of the dot product must be zero, as shown before). And the last condition is, again, just copied over.

The Jacobian of F is:

$$\nabla F^T\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix}\right) = \begin{bmatrix} Q & A & -I \\ A & 0 & 0 \\ \text{diag}(\mathbf{z}) & 0 & \text{diag}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \nabla F_1(\cdot)^T \\ \nabla F_2(\cdot)^T \\ \vdots \\ \nabla F_{n+m+n}(\cdot)^T \end{bmatrix} \in \mathbb{R}^{(n+m+n)\times(n+m+n)}$$

Note: Since Q is PD, A is full row rank, then if we assume that $\mathbf{z}$ and $\mathbf{x}$ are positive ($> \mathbf{0}$) vectors, then this gigantic matrix is invertible.

Our problem is now in a form that looks *nearly* suitable for Newton! The only concern is the non-negative constraints on $\mathbf{x}$ and $\mathbf{z}$. We need a way to take Newton steps without breaking our positivity constraints.

7.3 Naive Newton

Recall that a single step of Newton's method can be summarized in the following way (assuming the Jacobian is invertible):

$$\begin{bmatrix} \mathbf{x}^{(k+1)} \\ \mathbf{y}^{(k+1)} \\ \mathbf{z}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix} - \nabla^T F\left(\begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix}\right)^{-1} F\left(\begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix}\right)$$

This is our newton iteration step, and as long as our $\mathbf{x}, \mathbf{z} > \mathbf{0}$, our Jacobian will always be invertible. However, in practice we'll never actually invert out Jacobian (it's far too expensive). Instead we'll typically just choose to solve the linear system:

$$\nabla^T F\left(\begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix}\right)\left(\begin{bmatrix} \mathbf{x}^{(k+1)} \\ \mathbf{y}^{(k+1)} \\ \mathbf{z}^{(k+1)} \end{bmatrix} - \begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix}\right) = -F\left(\begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix}\right)$$

Let's rename this middle section, which are the variables that we're solving for, to $\begin{bmatrix} \Delta\mathbf{x}^{(k1)} & \Delta\mathbf{y}^{(k)} & \Delta\mathbf{z}^{(k)} \end{bmatrix}$. This represents the variables in the linear system which we are going to be solving for. This changes our system into:

$$\nabla^T F\left(\begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix}\right)\left(\begin{bmatrix} \Delta\mathbf{x}^{(k1)} \\ \Delta\mathbf{y}^{(k)} \\ \Delta\mathbf{z}^{(k)} \end{bmatrix}\right) = -F\left(\begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix}\right)$$

Note: *This* Δ^k *vector is exactly our Newton Direction (it's what we will add to the next iterate).*

However, we still cannot just apply Newton, because nothing about this Newton direction is guaranteed to keep $\mathbf{x}, \mathbf{z} \geq \mathbf{0}$. A simple solution to this is the Naive Newton Algorithm (Algorithm 5).

Algorithm 5: Naive-Newton Algorithm

Input: F
Output: $\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*$
$\begin{bmatrix} \mathbf{x}^{(0} & \mathbf{y}^{(0)} & \mathbf{z}^{(0)} \end{bmatrix} \leftarrow$ initialize a feasible guess;

for $k = 1, 2, \ldots$ *until* $F\left(\begin{bmatrix} \mathbf{x}^{(k)} & \mathbf{y}^{(k)} & \mathbf{z}^{(k)} \end{bmatrix} \right) \approx \mathbf{0}$ **do**

$\quad$ Solve $\nabla^T F\left(\begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix} \right) \left(\begin{bmatrix} \Delta\mathbf{x}^{(k)} \\ \Delta\mathbf{y}^{(k)} \\ \Delta\mathbf{z}^{(k)} \end{bmatrix} \right) = -F_{\epsilon_k \beta^{(k)}}\left(\begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix} \right);$

$\quad \begin{bmatrix} \mathbf{x}^{(k+1)} \\ \mathbf{y}^{(k+1)} \\ \mathbf{z}^{(k+1)} \end{bmatrix} \leftarrow \max_{\alpha^{(k)}} \left[\begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix} + \alpha^{(k)} \begin{bmatrix} \Delta\mathbf{x}^{(k)} \\ \Delta\mathbf{y}^{(k)} \\ \Delta\mathbf{z}^{(k)} \end{bmatrix} \right]$ s.t. $\mathbf{x}^{(k+1)}), \mathbf{z}^{(k+1)}) > 0;$

$\quad k \leftarrow k + 1;$

In other words, naive newton will go as far as it can in the direction Δ such that positivity of $\mathbf{x}, \mathbf{z}$ is maintained.

However, in practice this naive algorithm typically will fail to converge. "Naive Newton" will always have step sizes which are too small and will often get stuck.

7.4 Interior Point Methodology

At a high level, the solution to Naive Newton's failure is to try to "steer" towards the "inside" of the feasible region, while moving towards the goal. We allow the complimentarity constraint to relax a bit while we attempt to optimize, allowing us to make larger strides towards the root of F the other components of F, while additionally guiding our solution to a more and more "complimentary" solution with respect to $\mathbf{x}$ and $\mathbf{z}$. We'll formalize this below.

7.4.1 The Central Path

> **Definition:** F_τ
>
> For all $\tau > 0$, we will define $F_\tau : \mathbb{R}^{n+m+n} \to \mathbb{R}^{n+m+n}$ as
> $$F_\tau\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} \right) = \begin{bmatrix} Q\mathbf{x} + \mathbf{c} + A^T\mathbf{y} - \mathbf{z} \\ A\mathbf{x} - \mathbf{b} \\ \mathbf{z} \circ \mathbf{x} - \tau\mathbf{1} \end{bmatrix}$$

What does this say? For a given root of F_τ we *don't quite* have complimentarity in $\mathbf{x}, \mathbf{z}$. Namely, if $F_\tau(\begin{bmatrix} \mathbf{x} & \mathbf{y} & \mathbf{x} \end{bmatrix}) = \mathbf{0}$, then we have $\forall i \ \mathbf{x}_i \mathbf{z}_i = \tau$, and in particular that must also mean that $\mathbf{x}_i, \mathbf{z}_i > 0$. *All* of our components are > 0, and the Hadamard product of each component is equal.

> **Theorem 7.1**
>
> All notation as above.

If $\exists \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix}$ s.t. $F(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix})_{1:n+m} = \mathbf{0}$, AND $\mathbf{x}, \mathbf{z} > 0$, then

$$\forall \tau > 0, \exists \text{ unique vector } \begin{bmatrix} \mathbf{x}_\tau \\ \mathbf{y}_\tau \\ \mathbf{z}_\tau \end{bmatrix} \text{ s.t. } F_\tau(\begin{bmatrix} \mathbf{x}_\tau \\ \mathbf{y}_\tau \\ \mathbf{z}_\tau \end{bmatrix}) = \mathbf{0} \text{ and } \mathbf{x}_\tau, \mathbf{z}_\tau > 0$$

Note: $\mathbf{x}_\tau \to \mathbf{x}^*, \mathbf{z}_\tau \to \mathbf{z}^*$ *solution to QP as* $\tau \to 0$

No proof.

This theorem is saying is that if we have a solution which solves the first 2/3rds of our problem (i.e. the complimentarity between $\mathbf{x}$ and $\mathbf{z}$ does not hold, but everything else is satisfied), then for all $\tau > 0$, we have a solution which is *jsut as good*, except the complimentarities of all components of $\mathbf{x} \circ \mathbf{z}$ are equal but not 0.

Definition: Central Path

The **Central Path** is the set of solutions

$$\{\begin{bmatrix} \mathbf{x}_\tau & \mathbf{x}_\tau & \mathbf{z}_\tau \end{bmatrix} : \tau > 0\}$$

Intuitively, if we are "on" the central path, we might think of following it down the decreasing τ direction, to approach the solution where $\tau = 0$, which is exactly a solution to our original QP.

7.4.2 Central Path Neighborhoods

Definition: Average Complimentarity

For some $\begin{bmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} \end{bmatrix} \in \mathbb{R}^{n+m+n}$ s.t. $\mathbf{x}, \mathbf{z} > 0$ we define the **average complimentarity** as

$$\beta = \frac{\mathbf{x}^T \mathbf{z}}{n}$$

Definition: N_2 Neighborhood

For fixed parameter $\gamma \in (0, 1)$, define

$$N_2(\gamma) = \{\begin{bmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} \end{bmatrix} \in \mathbb{R}^{n+m+n} : \mathbf{x}, \mathbf{z} > 0, \ \|\mathbf{z} \circ \mathbf{x} - \beta\|_2 \leq \gamma\beta\}$$

This neighborhood is basically checking how far each complimentarity "violation" (i.e. each component of $\mathbf{z} \circ \mathbf{x}$ is off of the average complimentary violation β). The smaller the gamma, the more you restrict the complimentarity of $\mathbf{x}$ and $\mathbf{z}$ to be uniform across each component.

Note: Remember that if the first 2/3rds of F are equal to $\mathbf{0}$ and every component of $\mathbf{z} \circ \mathbf{x}$ is exactly the average, then you're on the central path! So you can think of N_2 as a space defined around the central path, whose size is controlled by γ, although N_2 itself says nothing about the feasibility of the other 2/3rds of our function F.

Definition: $N_{-\infty}$ Neighborhood

For fixed parameter $\delta \in (0, 1)$, define

$$N_{-\infty}(\delta) = \{\begin{bmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} \end{bmatrix} \in \mathbb{R}^{n+m+n} : \mathbf{x}, \mathbf{z} > 0, \ \forall i \ x_i z_i \geq \delta\beta\}$$

This neighborhood is much less restrictive than the N_2 neighborhood. This neighborhood only requires that each component must violate their complimentarity by at least some fraction of β, the average complimentarity.

*Note that, in this case, as our parameter δ **increases** this neighborhood becomes tighter. It forces each component to be closer to the average, meaning we're closer to the central path as we also optimize for the rest of our constraints. As $\delta \to 0$, this neighborhood becomes less restrictive. Since we can have some components with very small complentarity violations, this allows other components to compensate in drastic ways, which allows a much larger space of residents. Note again that this neighborhood is not defined with the other values of F in mind. It it solely worried about the complimentarity and positivity of $\mathbf{x}$ and $\mathbf{z}$.*

7.4.3 Interior Point Algorithm

The algorithm is defined in 6.

Algorithm 6: Interior-Point Algorithm

Input: $\epsilon_{min}, \epsilon_{max}$ s.t. $0 < \epsilon_{min} < \epsilon_{max} < 1$;
short step (N_2 neighborhood) or **long step** ($N_{-\infty}$ neighborhood);
γ if short step or δ if long step;
Output: $\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*$
$\begin{bmatrix} \mathbf{x}^{(0} & \mathbf{y}^{(0)} & \mathbf{z}^{(0)} \end{bmatrix} \leftarrow$ init. guess $\in N_2(\gamma)$ or $\in N_{-\infty}(\delta)$;

for $k = 1, 2, \dots$ *until* $F\left(\begin{bmatrix} \mathbf{x}^{(k)} & \mathbf{y}^{(k)} & \mathbf{z}^{(k)} \end{bmatrix} \right) \approx \mathbf{0}$ **do**

$\quad \epsilon_k \leftarrow \epsilon_{\min} < \epsilon_k < \epsilon_{\max}$;
$\quad \beta^{(k)} \leftarrow \frac{\mathbf{z}^{(k)T}\mathbf{x}^{(k)}}{n}$;
$\quad \begin{bmatrix} \Delta\mathbf{x}^{(k)} \\ \Delta\mathbf{y}^{(k)} \\ \Delta\mathbf{z}^{(k)} \end{bmatrix} \leftarrow$ solution to $\nabla^T F\left(\begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix} \right)\left(\begin{bmatrix} \Delta\mathbf{x}^{(k)} \\ \Delta\mathbf{y}^{(k)} \\ \Delta\mathbf{z}^{(k)} \end{bmatrix} \right) = -F_{\epsilon_k \beta^{(k)}}\left(\begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix} \right)$;
$\quad \begin{bmatrix} \mathbf{x}^{(k+1)} \\ \mathbf{y}^{(k+1)} \\ \mathbf{z}^{(k+1)} \end{bmatrix} \leftarrow \max_{\alpha^{(k)}} \begin{bmatrix} \mathbf{x}^{(k)} \\ \mathbf{y}^{(k)} \\ \mathbf{z}^{(k)} \end{bmatrix} + \alpha^{(k)} \begin{bmatrix} \Delta\mathbf{x}^{(k)} \\ \Delta\mathbf{y}^{(k)} \\ \Delta\mathbf{z}^{(k)} \end{bmatrix} \in N_2(\gamma)$ or $N_{-\infty}(\delta)$;
$\quad k \leftarrow k + 1$;

Each Newton step is optimizing an F_τ, where $\tau = \epsilon_k \beta^{(k)}$ (The Jacobian is always the same, regardless of τ).

Notice that if ϵ_k is close to 0, then this is a basic Newton step with respect to our original QP. That is, we are taking a step in the direction that finds a solution to the "top 2 chunks" of F, as well as the "bottom chunk" of F. If ϵ_k is close to 1 then we are not *improving* the complimentarity of $\mathbf{z}, \mathbf{x}$ but rather we are only moving ourselves towards equivalent complimentarity of all components of $\mathbf{z} \circ \mathbf{x}$ (i.e. moving towards the central path) while still optimizing for the top "chunks" of F.

Note: Typically in practice, it is common to alternate ϵ_k between values close to 1 and close to 0 to alternate taking steps towards complementarity and taking steps towards the "central path".

This algorithm gives us a way to solve a QP with inequalities in the constraints using an iterative method.

8 Matrix Games

We're going to shift focus for a few sections to talk about min-max games, and Lagrangian duality. This will tie in, in an important way, to our notion of duality in constrained optimization problems.

8.1 Deterministic Matrix Games

In a matrix game, we're going to have two players:

- a row-player, who we'll say is "me"
- a column-player, who we'll say is "you"

Here's an example:

$$M = \begin{bmatrix} 10 & -1 & 2 & -9 \\ 3 & 6 & -2 & 0 \\ -3 & 7 & 5 & 8 \end{bmatrix}$$

The rules of the game are: upon a choice of row and column, *you* give *me* the money equal to the amount in that matrix cell. So I want to *maximize* the value, and you want to *minimize* it.

We're going to start with the deterministic case: there's two cases to consider.

- me first (I pick rows): the problem then looks like $\max_i \min_j M_{i,j}$
- you first (You pick cols): the problem then looks like $\min_j \max_i M_{i,j}$.

In each of these cases, we both know what the other is going to pick given a choice of row or column. In other words, I know what you will pick after I pick any column, because I know you want to minimize. So, in this case, I will always pick the 2nd row, which maximizes (out of all possible rows) your choice. Thus, we say that the *value* of this game is -2 with $\arg(2,3)$.

Now, if you go first, then you will certainly pick the 3rd row... because then the maximum choice for me is 5, and that is the smallest value that you can present to me to pick from. Here, the value of the game is 5, with $\arg(3,3)$.

> **Proposition 8.1: Weak Duality**
>
> $$\max_i \min_j M_{ij} \leq \min_j \max_i M_{ij}$$
>
> or, in other words, *going second is an advantage.*

Proof. Let's say that (i^*, j^*) is the arg for when I go first ($\max_i \min_j M_{i,j}$). Let's say that (i', j') is the arg for when you go first, ($\min_j \max_i M_{i,j}$). Then, we have that $M_{i^*,j^*} \leq M_{i^*,j'} \leq M_{i',j'}$

Why? Notice that M_{i^*,j^*} and $M_{i^*,j'}$ both have i^*. Remember, that for the game where I went first, I picked i^* *first*, knowing that you would pick j^* because it was the min. *So j' can't possibly be better or you would have picked it!*. And we can make the same argument in the other direction! *You* picked j' first, knowing that I would pick i' to *maximize* the value. So (i^*, j') can't be bigger than (i', j') or, again, *I* would have picked it! ∎

8.2 Saddle Points

> **Definition: Saddle Point**
>
> A **saddle point** for $M, (i^*, j^*)$, with value M_{i^*,j^*}, is a point such that
>
> $$\forall i, j \ \ M_{i,j^*} \leq M_{i^*,j^*} \leq M_{i^*,j}$$
>
> In other words, we're at an equilibrium between the players.

An example of this is:

$$M = \begin{bmatrix} 20 & 2 & 9 & -1 \\ 11 & 13 & 10 & 12 \\ -6 & -8 & 7 & 15 \end{bmatrix}$$

Here, the saddle point is $(i^*, j^*) = (2, 3)$. Why? Look at the options between both players, given this point. If I fix the row to i^*, then you don't have any better choices than j^* (your other options are 11, 13, or 12, and you want to minimize). But the same goes for me... if you fix the column to j^*, then I certainly don't want to move off of i^*.

Proposition 8.2

For any M, if (i^*, j^*) and (i', j') are both saddle points, then (i^*, j') and (i', j^*) are also saddle points, and they all have the same value.

Proof. By definition of a saddle point, $\forall i, j \; M_{i,j^*} \leq M_{i^*,j^*} \leq M_{i^*,j}$ and $\forall i, j M_{i,j'} \leq M_{i',j'} \leq M_{i',j}$. Then,

$$M_{i'j'} \leq M_{i',j^*} \leq M_{i^*,j^*} \leq M_{i^*,j'} \leq M_{i',j'}$$

This is trivially true, since both (i', j') and (i^*, j^*) are saddle points. This sandwiching technique shows that everything here is equal. Since they're all equal, then they are all saddle points (by definition). ∎

Now, look at the matrix M with a saddle point above. Play the game where you go first, and then play the game where I go first. Notice how, in both games, the optimal value for both of us to pick is actually *the saddle point*. And in fact... **This is always the case!**

Theorem 8.1

For any M, $\max_i \min_j M_{i,j} = \min_j \max_i M_{i,j}$ iff there exists a saddle point, (i^*, j^*). Furthermore, $(i^*, j^*) \in$ arg$[\max_i \min_j M_{i,j}]$ & arg$[\min_j \max_i M_{i,j}]$.

i.e. if there is a saddle point, then it is the optimal choice for both players.

Proof. ($\Longleftarrow$) Suppose M_{i^*,j^*} is a saddle point. By definition

$$
\begin{aligned}
M_{i^*,j^*} &= \min_j M_{i^*,j} && \textit{even if you choose, you will choose } j^* \textit{ because it's a saddle point} \\
&\leq \max_i \min_j M_{i,j} && \textit{(1) If I am free to pick a row now, then I may pick something worse for you} \\
&\leq \min_j \max_i M_{i,j} && \textit{(2) True by our proposition above.} \\
&\leq \max_i M_{i,j^*} && \textit{(3) Reverse argument of (1)} \\
&= M_{i^*,j^*} && \textit{Again, because it's a saddle point. I'll always choose this}
\end{aligned}
$$

Thus, all of these inequalities are actually equalities. Notice that equality in (2) gives us that $\max_i \min_j M_{i,j} = \min_j \max_i M_{i,j}$. Equality in (1) says that in the $\max_i \min_j$ problem, picking i^* is the best thing I can do. Equality in (3) says that, in the $\min_j \max_i$ problem, j^* is the best choice you can make. By definition of saddle points, if I choose i^* or you choose j^*, the final result will be (i^*, j^*).

This shows that, if you have a saddle point, then both games will converge to the same point, and that's the best point that either side can pick.

($\Longrightarrow$) Suppose $(i', j') \in$ arg$[\max_i \min_j M_{i,j}]$ and that $(i'', j'') \in$ arg$[\min_i \max_j M_{i,j}]$, and further that $M_{i',j'} = M_{i'',j''}$.

Then,

$$M_{i',j''} \leq M_{i'',j''} = M_{i',j'} \leq M_{i'',j'} \implies M_{i'',j''} = M_{i',j'} = M_{i',j''}$$

which implies that $M_{i',j''}$ is a saddle point. ∎

8.3 Probabilistic Matrix Games

In this game, I pick a row probability vector $\mathbf{p}$, and you will pick a column probability vector $\mathbf{q}$. We call these games *mixed-strategy* games, where the deterministic case are called *pure-strategy* games.

For example, let's say we have a matrix M,

$$M = \begin{bmatrix} 10 & -1 & 2 & -9 \\ 3 & 6 & -2 & 0 \\ -3 & 7 & 5 & 8 \end{bmatrix}$$

One choice of $\mathbf{p}$ is $\mathbf{p} = \begin{bmatrix} .1 \\ .2 \\ .7 \end{bmatrix}$. One choice of $\mathbf{q}$ is $\mathbf{q}^T = \begin{bmatrix} .2 & .2 & .1 & .6 \end{bmatrix}$.

So the probabilities for each cell are

$$\mathbf{p}\mathbf{q}^T = \begin{bmatrix} .02 & .01 & .01 & .06 \\ .04 & .04 & .02 & .12 \\ .14 & .07 & .07 & .42 \end{bmatrix}$$

And the way we measure the "success", in this game, is by the expected value. We can write this succinctly as $E_M(\mathbf{p}, \mathbf{q}) = \mathbf{p}^T M \mathbf{q}$.

Note that "pure-strategy" games are a particular instance of mixed strategies, e.g. we can only assign probabilities 1 and 0 to rows or columns.

The 2 possible games that can be played are:

- "me-first": $\max_{\mathbf{p}} \min_{\mathbf{q}} E_M(\mathbf{p}, \mathbf{q})$ - In this problem, I have the much more complicated problem. Your job is easy here. Once you have my $\mathbf{p}$, then all you have to do is pick a $\mathbf{q}$ which minimizes the expected value. But I have to work really hard, because I have to consider *what you'll pick* as I pick my $\mathbf{p}$.

- "you-first": $\min_{\mathbf{q}} \max_{\mathbf{p}} E_M(\mathbf{p}, \mathbf{q})$ - Here, we have the opposite problem. Now, *your* job is the hard one, because you have to consider what I will do after you pick.

8.3.1 Second Player Strategy

Let's look at the example from above:

$$M = \begin{bmatrix} 10 & -1 & 2 & -9 \\ 3 & 6 & -2 & 0 \\ -3 & 7 & 5 & 8 \end{bmatrix}$$

In "me-first", if I pick $\mathbf{p} = \begin{bmatrix} .1 \\ .2 \\ .7 \end{bmatrix}$, what would your $\mathbf{q}$ be?

$$E_M(\begin{bmatrix} .1 \\ .2 \\ .7 \end{bmatrix}, \mathbf{q}) = \mathbf{q}_1\left((.1)10 + .2(3) + .7(-3)\right) + \mathbf{q}_2\left((.1)-1 + .2(6) + .7(7)\right)$$

$$+ \mathbf{q}_3\left((.1)2 + .2(-2) + .7(5)\right) + \mathbf{q}_4\left((.1)-9 + .2(0) + .7(8)\right)$$

$$= \mathbf{q}_1(-.5) + \mathbf{q}_2(6) + \mathbf{q}_3(3.3) + \mathbf{q}_4(4.7)$$

So... if you want to minimize the equation what will you pick? You should just put all of your eggs in one basket! Ie., you should choose $\mathbf{q} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$! This obviously minimizes the expected value.

> **Proposition 8.3**
>
> In mixed-strategy matrix games, the second player can *always* have an optimal strategy that is "pure". e.g.,
>
> $$\text{In "me-first": } \max_{\mathbf{p}} \min_{\mathbf{q}} E_M(\mathbf{p}, \mathbf{q}) = \max_{\mathbf{p}} \min_{j} \sum_{i} \mathbf{p}_i M_{i,j}$$
>
> $$\text{In "you-first": } \min_{\mathbf{p}} \max_{\mathbf{p}} E_M(\mathbf{p}, \mathbf{q}) = \min_{\mathbf{q}} \max_{i} \sum_{j} \mathbf{q}_j M_{i,j}$$

8.3.2   First-player strategy

We've seen that the second-player's strategy is relatively trivial. However, how does the first-player go about picking their vector? Well, it turns out that, for example in "me-first", my strategy is the linear program:

$$\max z \ \text{ s.t. } \ \forall j, \ \sum_{i} p_i m_{i,j} \geq z,$$
$$\forall i, \ p_i \geq 0,$$
$$\sum_{i} p_i = 1$$

with variables $\mathbf{p}, z$. What's going on here? The last two constraints are just making sure that we keep $\mathbf{p}$ to be a valid probability distribution. The top line is where the real magic is happening. Notice that, in this formalization, z is a value which is below the weighted sum of each column, where the weights are our variables $\mathbf{p}$. Since z is forced to be below each column-sum, when we maximize z we are pushing up the minimum value of the column-sums. Remember that when "you" are picking your $\mathbf{q}$ in this game, you will be picking the column with the minimum column-sum, weighted by my $\mathbf{p}$. So my goal is to find a $\mathbf{p}$ which maximizes z, the minimum column-sum value! So, we've sort of "built-in" your decision with z and our desire to maximize it, and now all that's left is to find a $\mathbf{p}$ which maximizes z. And this is a linear-programming problem!

Let's rewrite this into a concise matrix form:

$$\max \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}^T \begin{bmatrix} \mathbf{p} \\ z \end{bmatrix} \ \text{ s.t. } \ \begin{bmatrix} M^T & -\mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ z \end{bmatrix} \begin{bmatrix} \geq \mathbf{0} \\ = 1 \end{bmatrix},$$
$$\begin{bmatrix} \mathbf{p} \\ z \end{bmatrix} \begin{bmatrix} \geq \mathbf{0} \\ unrestricted \end{bmatrix}$$

Let's break this down a bit. The top row of our constraints is saying that $M^T \mathbf{p} \geq z\mathbf{1}$, which is exactly our earlier contraints. It's just assuring us that our weighted column-sum is greater than z. The bottom row just just assuring us that $\mathbf{p}$ sums to one. And the second constraint is just making sure that our $\mathbf{p}$ is also not negative, but we have *no constraint* on z.

Let's write out the dual of this problem:

$$\min \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \begin{bmatrix} -\mathbf{q} \\ w \end{bmatrix} \ \text{ s.t. } \ \begin{bmatrix} M & \mathbf{1} \\ -\mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} -\mathbf{q} \\ w \end{bmatrix} \begin{bmatrix} \geq \mathbf{0} \\ = 1 \end{bmatrix}$$
$$\begin{bmatrix} -\mathbf{q} \\ w \end{bmatrix} \begin{bmatrix} \leq \mathbf{0} \\ unrestricted \end{bmatrix}$$

Here we're just deciding to name our dual variable $-\mathbf{q}$ as opposed to $\mathbf{q}$ for naming convention. So think of $-\mathbf{q}$ as our regular dual variable.

Let's rephrase this (decompose it a little and switch the inequalities with our negative).

$$\min w \ \text{ s.t. } \ \forall i, \ \sum_{j} \mathbf{q}_j M_{i,j} \leq w,$$
$$\sum_{j} \mathbf{q}_j = 1,$$
$$\forall j, \ \mathbf{q}_j \geq \mathbf{0}$$

and... of course, this is *exactly* the "you-first" strategy! So my best strategy in the me-first and your best strategy in the you-first are dual problems! And what we have, given strong duality, is that these two problems have equal optimal objective function values since they're both feasible. So *it actually doesn't matter who goes first!* By strong duality, both games have the same value.

More formally:

Theorem 8.2

In mixed-strategy matrix games:
$$\max_{\mathbf{p}} \min_{\mathbf{q}} E_M(\mathbf{p}, \mathbf{q}) = \min_{\mathbf{q}} \max_{\mathbf{p}} E_M(\mathbf{p}, \mathbf{q})$$

Proof. By strong-duality! ∎

9 Lagrangian Duality

Now we shift from Matrix Games to a *very related* concept, the Lagrangian problem. We will see the notion of saddle-points from matrix game theory comes into play in this optimization problem, and then how it also ties into our KKT conditions.

9.1 The Lagrangian Function

Definition: The Lagrangian Function

Consider the problem P:
$$\min f(\mathbf{x}) \text{ s.t. } \mathbf{g}(\mathbf{x}) \leq \mathbf{0}$$
$$\mathbf{x} \in S$$
$$S \subseteq \mathbb{R}^n, f, \mathbf{g}_1, \mathbf{g}_2, \dots \mathbf{g}_m : S \to \mathbb{R}$$

Then, we define the **Lagrangian function** of P to be:
$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda^T \mathbf{g}(\mathbf{x})$$
$$= f(\mathbf{x}) + \sum_{i=1}^m \lambda_i \mathbf{g}_i(\mathbf{x})$$

defined for all $\mathbf{x} \in S, \lambda \in \mathbb{R}^m_{\geq 0}$.

---

*In essence, we've sort of relocated our **g** into our objective function, and included these things called lagrangian multipliers (λ) into it.*

Definition: Lagrangian Φ-function

$\forall \mathbf{x} \in S$, define
$$\Phi(\mathbf{x}) = \sup_{\lambda \in \mathbb{R}^m_{\geq 0}} \mathcal{L}(\mathbf{x}, \lambda)$$

Proposition 9.1

P is equivalent to
$$\inf_{\mathbf{x} \in S} \Phi(\mathbf{x}) = \inf_{\mathbf{x} \in S} \sup_{\overrightarrow{\lambda} \in \mathbb{R}^m_{\geq 0}} \mathcal{L}(\mathbf{x}, \overrightarrow{\lambda})$$

---

*We can think about matrix games here: You are picking an **x** that tries to minimize this value, and I'm picking λ which then maximizes the value. We've used Φ to sort of abstract "me" away from the equation, in other words I'm kind of like a black-box in this scenario, and you're picking **x** first.*

Proof. For any $\mathbf{x} \in S$, if $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ then $\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda^T \mathbf{g}(\mathbf{x}) \leq f(\mathbf{x})$, so $\Phi(\mathbf{x}) = f(\mathbf{x})$, since I will always set $\lambda = \mathbf{0}$ in order to maximize the equation.

If $\mathbf{g}(\mathbf{x}) \not\leq \mathbf{0}$, say $\mathbf{g}_i(\mathbf{x}) > 0$, then I will set $\lambda_i = \infty$, which will make $\mathcal{L}(\mathbf{x}, \lambda) \to \infty$, so $\Phi(\mathbf{x}) = \infty$.

Thus, $\inf_{\mathbf{x} \in S} \Phi(\mathbf{x}) = P$, since *you* will never ever choose an **x** which is not "feasible" by our original **g** restrictions. Thus, we are basically still doing the same - you're trying to find an **x** which minimizes $f(\mathbf{x})$ but satisfies the constraints $\mathbf{g}(\mathbf{x})$. They're equivalent problems! $\blacksquare$

9.2 The Lagrangian Dual

Now, let's take a look at what the *dual* form of this looks like, where I get to pick λ first: We can define a similar function as before:

> **Theorem 9.1: Weak Duality**
>
> If $\mathbf{x}_*$ feasible in P, λ_* feasible in DP, then $\mathbf{ofv}_{DP}(\lambda_*) \leq \mathbf{ofv}_{LP}(\mathbf{x}_*)$

Proof. Notice that
$$\Theta(\lambda_*) = \inf_{x \in S} \mathcal{L}(\mathbf{x}, \lambda_*) \leq \mathcal{L}(\mathbf{x}_*, \lambda_*) \leq \sup_{\lambda \in \mathbb{R}^m_{\geq 0}} \mathcal{L}(\mathbf{x}_*, \lambda) = \Phi(\mathbf{x}_*)$$

We can see by similar arguments as we used in the game theory arguments earlier. In essence, $\mathcal{L}(\mathbf{x}_*, \lambda_*)$ is a value which has a fixed $\mathbf{x}_*$ and a fixed $\overrightarrow{\lambda}_*$ (*neither of which is necessarily optimal*). This is obviously less than or greater than the sup or inf if we allow the problem to consider *all $\lambda \in \mathbb{R}^m_{\geq 0}$* or *all $\mathbf{x} \in S$* or ∎

> **Theorem 9.2: Supervisor Principle**
>
> If $\mathbf{x}_*$ feasible in P, λ_* feasible in DP   s.t. $\mathbf{ofv}_{DP}(\lambda_*) = \mathbf{ofv}_P(\mathbf{x}_*)$ then $\mathbf{x}_*$ optimal in P and λ_* optimal in DP.

For classes of problems where this happens, we say there is *no duality-gap*. If we have strict inequality ($<$) then we say that there *is a duality-gap*.

9.3   Example: Linear Pogram

Let's look at an example where this duality can occur: Suppose $A \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m, \mathbf{c} \in \mathbb{R}^n$ such that:
$$P : \min \mathbf{c}^T \mathbf{x} \text{ s.t. } A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}$$

9.3.1   A slightly unfair formulation

We want to compute the Lagrangian dual. We can phrase this as $\mathbf{b} - A\mathbf{x} \leq \mathbf{0} = g(\mathbf{x})$ and $\mathbf{x} \geq \mathbf{0} = \mathbf{x} \in S$.

This is actually not a fair translation of the problem, since we are packing some of our restrictions (some $\mathbf{g}$ functions) into restrictions on S instead (making it a closed set). There is a better way to do this, which we'll see in a second.

We now have the Lagrangian function:
$$\forall \mathbf{x} \geq \mathbf{0}, \lambda \in \mathbb{R}^m_{\geq 0}, \ \mathcal{L}(\mathbf{x}, \lambda) = \mathbf{c}^T \mathbf{x} + \lambda^T (\mathbf{b} - A\mathbf{x})$$
$$= \lambda^T \mathbf{b} + (\mathbf{c}^T - \lambda^T A)\mathbf{x}$$

What does the dual problem look like for this Lagrangian? Remember in the dual problem, I am choosing lambdas and trying to *maximize* the overall function value $\sup \Theta(\lambda)$. Since $\lambda \geq \mathbf{0}$, let's see what our options are for λ

- if $\lambda^T A \leq \mathbf{c}^T$ then $(\mathbf{c}^T - \lambda^T A) \geq \mathbf{0}$, and since $\mathbf{x} \geq \mathbf{0}$, then $\Theta(\lambda) = \inf_{\mathbf{x} \geq \mathbf{0}} \mathcal{L}(\mathbf{x}, \lambda) = \lambda^T \mathbf{b}$ since *you* will always choose $\mathbf{x} = \mathbf{0}$ to minimize the function value.

- if $\lambda^T A \not\leq \mathbf{c}^T$ then $(\mathbf{c}^T - \lambda^T A) \not\geq \mathbf{0}$ and now you can choose some component of $\mathbf{x}$ to be ∞ which will make the whole value of the function go to $-\infty$. So, $\Theta(\lambda) = \inf_{\mathbf{x} \geq \mathbf{0}} \mathcal{L}(\mathbf{x}, \lambda) = -\infty$

So, essentially, the dual of the Lagrangian can be re-written as:

$$\sup_{\vec{\lambda} \geq 0} \Theta(\vec{\lambda}) = \max \vec{\lambda}^T \mathbf{b} \ \text{ s.t. } \ \vec{\lambda}^T A \leq \mathbf{c}, \ \vec{\lambda} \geq \mathbf{0}$$

which is *exactly* the Dual Formulation of the original P (a linear program).

9.3.2 A slightly cleaner formulation

Now, let's write this our more fairly.

$$P : \min \mathbf{c}^T \mathbf{x} \ \text{ s.t. } \ \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -A \\ -I \end{bmatrix} \mathbf{x} \leq \mathbf{0}, \mathbf{x} \in \mathbb{R}^n$$

Now, our S is an open set ($\mathbb{R}^n$) and all of our constraints are packed into the $\mathbf{g}$ function, and this is a nicer way to write this out.

To write out the Lagrangian now, let's write out our variables as $\lambda = \begin{bmatrix} \mathbf{y} \in \mathbb{R}^m \\ \mathbf{z} \in \mathbb{R}^n \end{bmatrix}$ and now our Lagrangian looks like, $\forall \mathbf{x} \in \mathbb{R}, \mathbf{y} \in \mathbb{R}^m \geq \mathbf{0}, \mathbf{z} \in \mathbb{R}^n \geq \mathbf{0}$:

$$\mathcal{L}(\mathbf{x}, \lambda) = \mathbf{c}^T \mathbf{x} + \lambda^T \left(\begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -A \\ -I \end{bmatrix} \mathbf{x} \right) = \mathbf{c}^T \mathbf{x} + \mathbf{y}^T (\mathbf{b} - A\mathbf{x}) + \mathbf{z}^T (-\mathbf{x}) = \mathbf{y}^T \mathbf{b} + (\mathbf{c}^T - \mathbf{y}^T A - \mathbf{z}^T) \mathbf{x}$$

As above,let's break up our possible lambda values:

- if $\mathbf{c}^T - \mathbf{y}^T A - \mathbf{z} = \mathbf{0}$ then $\Theta(\vec{\lambda}) = \inf_{\mathbf{x} \in \mathbb{R}} \mathcal{L}(\mathbf{x}, \vec{\lambda}) = \mathbf{y}^T \mathbf{b}$
- if $\mathbf{c}^T - \mathbf{y}^T A - \mathbf{z} \neq \mathbf{0}$ then $\Theta(\vec{\lambda}) = \inf_{\mathbf{x} \in \mathbb{R}} \mathcal{L}(\mathbf{x}, \vec{\lambda}) = -\infty$ *(remember in this case $\mathbf{x}$ is not restricted to be positive).*

And so the dual ultimately looks like

$$\max \mathbf{b}^T \mathbf{y} \ \text{ s.t. } \ \mathbf{c}^T - \mathbf{y}^T A = \mathbf{z}, \ \mathbf{y}, \mathbf{z} \geq \mathbf{0}$$

which we can simplify, since, in this formulation, $\mathbf{z}$ only appears once and is only restricted to be positive, as:

$$\max \mathbf{b}^T \mathbf{y} \ \text{ s.t. } \ \mathbf{c}^T - \mathbf{y}^T A \geq \mathbf{0}, \ \mathbf{y} \geq \mathbf{0}$$

which we *finally* can simplify one more time to look like:

$$\max \mathbf{b}^T \mathbf{y} \ \text{ s.t. } \ A^T \mathbf{y} \leq \mathbf{c}, \ \mathbf{y} \geq \mathbf{0}$$

which is a very comfortable form of the dual of the LP :)

So we've seen that converting our linear programming problem into it's Lagrangian form and solving the Lagrangian dual is the same as solving the dual of the LP.

9.4 Example: A quadratic program

Let's look at another example: Suppose $A \in \mathbb{R}^{n \times n}$ sym. PD, $\mathbf{b} \in \mathbb{R}^n$ non-zero, $\mathbf{c} \in \mathbb{R}_{>0}$ and

$$P : \min \frac{1}{2} \mathbf{x}^T A \mathbf{x} \ \text{ s.t. } \ \mathbf{b}^T \mathbf{x} + \mathbf{c} \leq 0$$

and we want to compute and solve the dual. Well, the lagrangian looks like

$$\forall \mathbf{x} \in \mathbb{R}^n, \lambda \geq \mathbf{0}, \ \mathcal{L}(\mathbf{x}, \lambda) = \frac{1}{2} \mathbf{x}^T A \mathbf{x} + \lambda (\mathbf{b}^T \mathbf{x} + c)$$

Now, note that

$$\Theta(\lambda) = \inf_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \mathbf{x}^T A \mathbf{x} + \lambda (\mathbf{b}^T \mathbf{x} + c)$$

this function is actually convex with respect to **x**. So once we pass in a given λ, the minimization will just solve for the **x** which is a global min of that function. And, since A is invertible (it's PD) that means that we can directly solve for it numerically. For a given λ, the **x** that minimizes the langrangian is given as $\mathbf{x} = -\lambda A^{-1}\mathbf{b}$

We can just plug this into our original Θ function, then, to get

$$\Theta(\lambda) = \frac{1}{2}(-\lambda A^{-1}\mathbf{b})^T A(-\lambda A^{-1}\mathbf{b}) + \lambda(\mathbf{b}^T(-\lambda A^{-1}\mathbf{b}) + c)$$

$$= \frac{\lambda^2}{2}\mathbf{b}^T A^{-1}\mathbf{b} - \lambda^2 \mathbf{b}^T A^{-1}\mathbf{b} + \lambda c$$

and thus the (DP) looks like

$$\sup_{\lambda \geq 0} \frac{-\lambda^2}{2}\mathbf{b}^T A^{-1}\mathbf{b} + \lambda c$$

if we set the derivative $-\lambda\mathbf{b}^T A^{-1}\mathbf{b} + c = 0 \to \lambda = \frac{c}{\mathbf{b}^T A^{-1}\mathbf{b}} > 0$ which is a unique global max, making the **ofv** to be $\frac{c^2}{2\mathbf{b}^T A^{-1}\mathbf{b}}$. So, in this case, we have used the Lagrangian dual to exactly solve our system! It's a pretty powerful tool!

9.5   Lagrangian Saddle Points

Definition: Lagrangian Saddle Point

$\bar{\mathbf{x}} \in S, \bar{\lambda} \in \mathbb{R}^m_{\geq 0}$ are a **saddle point** of the Lagrangian if

$$\forall \mathbf{x} \in S, \lambda \in \mathbb{R}^m_{\geq 0}, \ \ \mathcal{L}(\bar{\mathbf{x}}, \lambda) \leq_{(1)} \mathcal{L}(\bar{\mathbf{x}}, \bar{\lambda}) \leq_{(2)} \mathcal{L}(\mathbf{x}, \bar{\lambda})$$

Theorem 9.3: A

All notation as above.

$\bar{\mathbf{x}}, \bar{\lambda}$ are a saddle point of Lagrangian iff
  i) $\forall \mathbf{x} \in S, \ \mathcal{L}(\bar{\mathbf{x}}, \bar{\lambda}) \leq \mathcal{L}(\mathbf{x}, \bar{\lambda})$ *(i.e. ineq. 2 above)*
  ii) $\mathbf{g}(\bar{\mathbf{x}}) \leq \mathbf{0}$
  iii) $\bar{\lambda}^T \mathbf{g}(\bar{\mathbf{x}}) = 0$

Proof. All we really need to show is that item (ii) & (iii) in the proof imply inquality (1) in the saddle point definition, since inequality (2) is already satisfed by item (i) of the theorem.

So we need to show that $\mathbf{g}(\bar{\mathbf{x}}) \leq \mathbf{0}, \bar{\lambda}^T \mathbf{g}(\bar{\mathbf{x}}) = 0 \iff \forall \lambda \in \mathbb{R}^m_{\geq 0}, \ \mathcal{L}(\bar{\mathbf{x}}, \lambda) \leq \mathcal{L}(\bar{\mathbf{x}}, \bar{\lambda})$.

($\implies$) Suppose (1) in the definition holds(i.e. $\forall \lambda \in \mathbb{R}^m_{\geq 0}, \ f(\bar{\mathbf{x}}) + \lambda^T \mathbf{g}(\bar{\mathbf{x}}) \leq f(\bar{\mathbf{x}}) + \bar{\lambda}^T \mathbf{g}(\bar{\mathbf{x}})$)

By way of contradiction, if $\exists i$ s.t. $\mathbf{g}_i(\bar{\mathbf{x}}) > 0$, then send $\lambda_i = \infty$, and all other components to zero. Then we have just provided a λ that obviously contradicts (1). So this can't be true, and $\mathbf{g}(\bar{\mathbf{x}}) \leq \mathbf{0}$, which satisfies (ii).

Next, consider the case where $\lambda = \mathbf{0}$. Since (1) is true, we can pick this lambda and it still must hold that

$$f(\bar{\mathbf{x}}) + \lambda^T \mathbf{g}(\bar{\mathbf{x}}) \leq f(\bar{\mathbf{x}}) + \bar{\lambda}^T \mathbf{g}(\bar{\mathbf{x}})$$

and thus we have

$$f(\bar{\mathbf{x}}) \leq f(\bar{\mathbf{x}}) + \bar{\lambda}^T \mathbf{g}(\bar{\mathbf{x}}) \implies 0 \leq \bar{\lambda}^T \mathbf{g}(\bar{\mathbf{x}})$$

However, since $\bar{\lambda} \geq 0$ and $\mathbf{g}(\bar{\mathbf{x}}) \leq \mathbf{0}$, then $0 \leq \bar{\lambda}^T \mathbf{g}(\bar{\mathbf{x}}) \leq 0$, and thus $\bar{\lambda}^T \mathbf{g}(\bar{\mathbf{x}}) = 0$, and so (iii) holds as well.

($\impliedby$) Suppose that (ii) and (iii) hold. We want to show that (1) holds. Let's do that:

$$\forall \lambda \in \mathbb{R}^m_{\geq 0}, \ \mathcal{L}(\bar{\mathbf{x}}, \lambda) = f(\bar{\mathbf{x}}) + \lambda^T \mathbf{g}(\bar{\mathbf{x}}) \leq f(\bar{\mathbf{x}}) = f(\bar{\mathbf{x}}) + \bar{\lambda}^T \mathbf{g}(\bar{\mathbf{x}}) = \mathcal{L}(\bar{\mathbf{x}}, \bar{\lambda})$$

which shows that (1) holds, which completes the proof.   ∎

> **Theorem 9.4: B**
>
> $\bar{\mathbf{x}} \in S, \bar{\lambda} \in \mathbb{R}^m_{\geq 0}$ are a saddle point of the Lagrangian iff $\bar{\mathbf{x}}$ is a global min of P and $\bar{\lambda}$ is a global max of P's Lagrangian dual DP, and there is no duality gap.

Proof. ($\implies$) Suppose $\bar{\mathbf{x}} \in S, \bar{\lambda} \in \mathbb{R}^m_{\geq 0}$ are a saddle point of Lagrangian. Then

$$
\begin{aligned}
\mathcal{L}(\bar{\mathbf{x}}, \bar{\lambda}) &= \inf_{\mathbf{x} \in S} \mathcal{L}(x, \bar{\lambda}) && \text{by thm. A(i)} \\
&\leq \sup_{\lambda \in \mathbb{R}^m_{\geq 0}} \inf_{\mathbf{x} \in S} \mathcal{L}(\mathbf{x}, \lambda) && \text{by defs of sup and inf (3)} \\
&= \mathbf{ofv}_{DP} && \text{by def.} \\
&\leq \mathbf{ofv}_P && \text{by weak duality (4)} \\
&\leq f(\bar{\mathbf{x}}) && \bar{\mathbf{x}} \text{ is feasible in (P) by Them. A(ii) so } \mathbf{g}(\bar{\mathbf{x}}) \leq \mathbf{0} \text{ (5)} \\
&= f(\bar{\mathbf{x}}) + \bar{\lambda}^T \mathbf{g}(\bar{\mathbf{x}}) && \text{by Thm. A(iii)} \\
&= \mathcal{L}(\bar{\mathbf{x}}, \bar{\lambda}) && \text{(6)}
\end{aligned}
$$

which completes our squeeze play, meaning that all of the values considered in here are equal. This shows several things, namely:

- $\bar{\mathbf{x}}$ is optimal in (P) by equality in (5).

- $\bar{\lambda}$ is optimal in (DP) by equality in (3).

- no duality gap by equality in (4).

- common **oofv** for (P) and (DP) is $\mathcal{L}(\bar{\mathbf{x}}, \bar{\lambda})$ by (6).

so having a saddle point gives us all these nice things.

($\impliedby$) Suppose $\bar{\mathbf{x}} \in S$ optimal in (P), $\bar{\lambda}$ optimal in (DP) and there is no duality gap.

$$
\begin{aligned}
\mathbf{oofv}_{DP} &= \inf_{\mathbf{x} \in S} \mathcal{L}(\mathbf{x}, \bar{\lambda}) && \text{since } \bar{\lambda} \text{ optimal} \\
&\leq \mathcal{L}(\bar{\mathbf{x}}, \bar{\lambda}) && \text{def of inf. (7)} \\
&= f(\bar{\mathbf{x}}) + \bar{\lambda}^T \mathbf{g}(\bar{\mathbf{x}}) && \text{def of } \mathcal{L} \\
&\leq f(\bar{\mathbf{x}}) && \text{since } \bar{\lambda} \geq \mathbf{0}, \mathbf{g}(\bar{\mathbf{x}}) \leq \mathbf{0}. \text{ (8)} \\
&= \mathbf{oofv}_P && \text{since we said } \bar{\mathbf{x}} \text{ is optimal}
\end{aligned}
$$

and finally, no duality gap implies that all of these are equal! Now, note that (7) implies Thm. A(i). The feasibility of $\bar{\mathbf{x}}$ implies Thm. A(ii). And finally equality in (8) implies Thm. A(iii). Thus, all of Thm. A's conditions hold, and thus $\bar{\mathbf{x}}, \bar{\lambda}$ are saddle points of the Lagrangian. ∎

9.6 KKT Conditions

> **Corollary 9.1**
>
> KKT conditions are sufficient for optimality in a convex program.
>
> *(No need for Constraint Qualifications!)*
>
> A **convex program** is defined as
> $$(P): \min f(x) \text{ s.t. } \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \mathbf{x} \in S$$
>
> where
> - $S \subseteq \mathbb{R}^n$ non-empty, open, convex
> - $f : S \to \mathbb{R}$ cont. diff. and convex
> - $\forall i = 1, 2, \dots, m \ g_i : S \to \mathbb{R}$ cont. diff. and convex.

Proof. The sketch is as follows:

$$\text{KKT conditions} \implies \text{Saddle Point by Thm. A} \iff \text{Optimal by Thm. B}$$

If $\bar{\mathbf{x}}, \bar{\lambda}$ satisfy KKT conditions, that says the following:

$$\mathbf{g}(\bar{\mathbf{x}}) \leq \mathbf{0}, \bar{\mathbf{x}} \in S, \qquad\qquad \text{primal feas. of } \bar{\mathbf{x}}$$
$$\text{and } \bar{\lambda} \geq \mathbf{0} \qquad\qquad \text{dual feasibility}$$
$$\text{and } \bar{\lambda}^T \mathbf{g}(\bar{\mathbf{x}}) = 0 \qquad\qquad \text{complimentary slackness}$$
$$\text{and } \nabla f(\bar{\mathbf{x}}) + \nabla \mathbf{g}(\bar{\mathbf{x}})\bar{\lambda} = \mathbf{0}$$

Thm A, on the other hand, characterizes a saddle point of $\mathcal{L}$ as:

$$\bar{\mathbf{x}} \in S, \bar{\lambda} \in \mathbb{R}^m_{\geq 0}$$
$$\forall \mathbf{x} \in S, \ \mathcal{L}(\bar{\mathbf{x}}, \bar{\lambda}) \leq \mathcal{L}(\mathbf{x}, \bar{\lambda})$$
$$\mathbf{g}(\bar{\mathbf{x}}) \leq \mathbf{0}$$
$$\bar{\lambda}^T \mathbf{g}(\bar{\mathbf{x}}) = 0$$

All we need to show, then, is that $\nabla f(\bar{\mathbf{x}}) + \sum_{i=1}^m \nabla \mathbf{g}_i(\bar{\mathbf{x}})\bar{\lambda}_i = \mathbf{0} \implies \forall \mathbf{x} \in S, \mathcal{L}(\bar{\mathbf{x}}, \bar{\lambda}) \leq \mathcal{L}(\mathbf{x}, \bar{\lambda})$

Note: $f, \mathbf{g}_i$ convex functions $\implies \mathcal{L}(\mathbf{x}, \bar{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^m \bar{\lambda}_i \mathbf{g}_i(\mathbf{x})$ is also a convex function because $\bar{\lambda} \geq 0$.

Let's take the $\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda) = \nabla f(\mathbf{x}) + \sum_{i=1}^m \bar{\lambda}_i \nabla \mathbf{g}_i(\mathbf{x})$ and lastly note that if we plug $\bar{\mathbf{x}}$ into it, then our KKT conditions say that this is equal to 0. But if the gradient of our Lagrangian is 0, then we are at a global min, by the convexity of all of our functions. And thus, $\forall \mathbf{x} \in S, \mathcal{L}(\bar{\mathbf{x}}, \bar{\lambda}) \leq \mathcal{L}(\mathbf{x}, \bar{\lambda})$ ∎

It's worth noting again that, in this formulation, we needed *no constraint qualifications*! So we've seen that KKT conditions are sufficient for global optimality, i.e. if we find a KKT point of P (when P is a convex program), then that is enough to guarantee global optimality. What about for a non-convex program?

Theorem 9.5

Let S by a non-empty, open set, and $f, \mathbf{g}_i : S \to \mathbb{R}$ all be twice continuously differentiable. Consider the problem P:

$$\min f(x) \ \text{s.t.} \ \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \mathbf{x} \in S$$

Suppose also $\bar{\mathbf{x}} \in S$ is such that $\forall i \in \mathcal{A}_{\bar{\mathbf{x}}} \ \nabla^2 \mathbf{g}_i(\bar{\mathbf{x}})$ is PD, $\nabla^2 f(\bar{\mathbf{x}})$ is PD,

Remember that $\mathcal{A}_{\bar{\mathbf{x}}}$ represents the active constraints of $\bar{\mathbf{x}}$.

Then $\bar{\mathbf{x}}$ being a KKT point $\implies \bar{\mathbf{x}}$ is a local min of P.

Proof. By continuity of S being open, egien values and $\mathbf{g}, \exists \epsilon > 0$ s.t. $N_\epsilon(\bar{\mathbf{x}}) \in S$ and:

$$\forall i \notin \mathcal{A}_{\bar{\mathbf{x}}} \ \forall \mathbf{x} \in N_\epsilon(\bar{\mathbf{x}}) \text{ then } \mathbf{g}(\mathbf{x})_i \leq 0$$
$$\forall i \in \mathcal{A}_{\bar{\mathbf{x}}} \ \forall \mathbf{x} \in N_\epsilon(\bar{\mathbf{x}}) \text{ then } \nabla^2 \mathbf{g}(\mathbf{x})_i \text{ is PSD}$$
$$\forall \mathbf{x} \in N_\epsilon(\bar{\mathbf{x}}) \text{ then } \nabla^2 f(\mathbf{x}) \text{ is PSD}$$

Now, consider

$$\hat{P}: \ \min f(\mathbf{x}) \ \text{s.t.} \ \forall i \in \mathcal{A}_{\bar{\mathbf{x}}} \ \mathbf{g}_i(\mathbf{x}) \leq 0$$
$$\mathbf{x} \in N_\epsilon(\bar{\mathbf{x}})$$

Firstly, note that if $\bar{\mathbf{x}}$ is a KKT point of P, then it is a KKT point of $\hat{P}$ (by the complimentary slackness, the λ_i for the non-active constraints are all 0 anyways).

Now, notice that $\hat{P}$ is *convex program!*. $N_\epsilon(\bar{\mathbf{x}})$ is an open, convex set, and all of our constraints have PSD hessians for all $\mathbf{x} \in N_\epsilon(\bar{\mathbf{x}})$. Thus, if $\bar{\mathbf{x}}$ is a KKT point of $\hat{P}$, then it is a global min of $\hat{P}$, and therefore $\bar{\mathbf{x}}$ is a local min of P, since $\hat{P}$ is a "small local problem" within P.

$\blacksquare$

So, we have now shown how to use local convexity (Positive Definiteness of active constraints and f at the point $\bar{\mathbf{x}}$) to show that KKT points imply local minimality of more general problems (i.e. outside the range of *convex* programs).

Note that this "direction" of KKT points is a bit of a departure from our previous conversation of KKT points in earlier sections. Remember earlier, we talked about how (with constraint qualifications), then $\bar{\mathbf{x}}$ being a local min implied that you were a KKT point. This meant that when we were solving for optimal points of our problem, we discarded all points that were not KKT points. *We can't do that in this case!*. Without Constraint Qualifications, we haven't shown that *all local or global min* are KKT points. So we need to be careful here.

9.7 A final example: A different quadratic program

This section is a slight tanget. We will see how we can use the Lagrangian dual to derive a dual problem of a quadratic program with inequality constraints that is, in general, much easier to solve *in practice* than the original problem.

Consider a symmetric, Positive Definite matrix $Q \in \mathbb{R}^{n \times n}, A \in \mathbb{R}^{m \times n}, \mathbf{c} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m$ and the quadratic problem, QP:

$$\min \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} \text{ s.t. } A\mathbf{x} \geq \mathbf{b}$$

Earlier, recall that we examined the KKT points of a Quadratic Program with equality constraints. Here we have *inequality constraints.*. First, we need to convert our constraints into a friendly form:

$$\mathbf{g}(\mathbf{x}) = \mathbf{b} - A\mathbf{x} \leq \mathbf{0}$$

Then, let's write out the Lagrangian of this:

$$\mathcal{L}(\mathbf{x}, \lambda) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} + \lambda^T \mathbf{b} - \lambda^T A \mathbf{x}$$

defined for all $\mathbf{x} \in \mathbb{R}^n$, $\lambda \geq \mathbf{0}$.

Now, the dual of this problem is a sup inf problem, of the form

$$\forall \lambda \in \mathbb{R}^m_{\geq 0} \ \Theta(\lambda) = \inf_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} + \lambda^T \mathbf{b} - \lambda^T A \mathbf{x}$$

This last term is *fixed* with respect to λ (once we pass in lambda, it's fixed). *And* it turns out that this function is convex in $\mathbf{x}$, because the Hessian (Q) is P.D. everywhere. Therefore the solution to the inf is a stationary point... we can simply set $Q\mathbf{x} + \mathbf{c} - A^T \lambda = 0$ and solve to get $\mathbf{x} = Q^{-1}(A^T \lambda - \mathbf{c})$. Thus, we can write $\Theta(\lambda)$ as

$$\Theta(\lambda) = \frac{1}{2}(A^T \lambda - \mathbf{c})Q^{-1}(A^T \lambda - \mathbf{c}) + \mathbf{c}^T\left(Q^{-1}(A^T \lambda - \mathbf{c})\right) + \lambda^T \mathbf{b} - \lambda^T A\left(Q^{-1}(A^T \lambda - \mathbf{c})\right)$$

$$= -\frac{1}{2}\lambda^T\left(AQ^{-1}A^T\right)\lambda + \lambda^T\left(\mathbf{b} + AQ^{-1}\mathbf{c}\right) - \frac{1}{2}\mathbf{c}^T Q^{-1}\mathbf{c}$$

So, given any *positive* lambda, if we plug it into *Theta*, the above is the result that we will get. Thus, we can write the Dual of the Quadratic Program (DQP) as:

$$\max -\frac{1}{2}\lambda^T\left(AQ^{-1}A^T\right)\lambda + \lambda^T\left(\mathbf{b} + AQ^{-1}\mathbf{c}\right) - \frac{1}{2}\mathbf{c}^T Q^{-1}\mathbf{c} \text{ s.t. } \lambda \geq \mathbf{0}$$

Call $M = AQ^{-1}A, \mathbf{v} = \mathbf{b} + AQ^{-1}\mathbf{c}, \alpha = \frac{1}{2}\mathbf{c}^T Q^{-1}\mathbf{c}$ which are all constant.

Note that QP is a *convex program*. $\mathbf{g}_i$ is affine, for all i, the hessian of f is P.D., and our feasible region is closed and convex. Additionally, we have a constraint qualification holding. Thus, we know that we have a KKT point and that the KKT point is

the global min of this problem. In particular, this means that we have no duality gap, so QP and DQP are equivalent to each other.

Now, note additionally that $AQ^{-1}A^T$ is positive semi-definite:

$$\forall \mathbf{z}, \ \mathbf{z}^T A Q^{-1} A^T \mathbf{z} = \mathbf{y}^T Q^{-1} \mathbf{y} \geq 0$$

because of the positive definiteness of Q. If A is full row-rank, then $\forall \mathbf{z} \neq \mathbf{0}$, $A^T \mathbf{z} \neq 0$.

Note: We cannot just arbitrarily row-reduce A to make it full row-rank in this case, because we are dealing with inequalities! Regardless, we can see that the objective function of DQP is convex.

Now, we know that the problems DQP and QP have equivalent solutions (there is no duality gap). We additionally can see that solving DQP is similarly difficult to solving QP, as they are both convex (or concave) quadratic programs. However, here's the punchline: *The constraints for DQP are vastly easier to work with than the constraints of QP.* It is much, much easier to solve problems in which we only need to maintain positivity rather than needing to maintain $A\mathbf{x} \geq \mathbf{b}$. So the dual formulation, in this case, gives us a much easier type of problem to solve.

In particular, there is a set of methods (which won't be convered in these notes) that involve solving a problem by significantly relaxing the problem's constraints and then *mapping* the found solution *back into* the feasible region of the original problem. This is much easier to do with the constraints of DQP than it is with the constraints of QP.

While this is slightly tangential, since we don't cover these methods, they are *extremely popular* in practice, and thus this relation is worth nothing.

10 Penalty and Barrier Methods

Now we're going to examine a slightly different approach to solving these constrained optimization problems. Namely, we're going to frame them in such a way that we can view them as unconstrained optimization problems. The purpose of this, of course, is to be able to use *unconstrained* optimization techniques for our constrained problems.

Let's start by talking about *penaly functions*.

10.1 Penalty Functions

Definition: positive function

For any $t \in \mathbb{R}$ let

$$pos(t) = \begin{cases} t \text{ if } t \geq 0 \\ 0 \text{ if } t < 0 \end{cases}$$

It's easy to see the point of this function. We only return the value of the function if it's positive, i.e. greater than 0.

Now, let's return to our traditional problem setting. Consider the problem, P:

$$\min f(x) \text{ s.t. } \mathbf{g}(\mathbf{x}) \leq \mathbf{0},$$
$$\mathbf{x} \in S$$

where $S \subseteq \mathbb{R}^n$ is a non-empty set, and f, $\forall i \ \mathbf{g}_i : S \to \mathbb{R}$ are continuous.

Definition: Penalty Function

Assume you are given a positive integer γ.

Then the **penalty function** is, for all $\mathbf{x} \in S$

$$\alpha(\mathbf{x}) = \sum_{i=1}^{m} pos(\mathbf{g}_i(\mathbf{x}))^\gamma$$

This function is just concerned with the *positive* parts of $\mathbf{g}_i(\mathbf{x})$, where the purpose of γ is to almost *emphasize* that positive part.

Note: $\forall \mathbf{x} \in S$ that $\alpha(\mathbf{x}) = 0 \iff \mathbf{x}$ *feasible in P*. So we can think of this as giving us a *weight* for each piece of $\mathbf{g}$ that is infeasible, giving us a sense of how infeasible a given $\mathbf{x}$ is.

Note: if you pick $\gamma = 2$ then the function is continuously differentiable. If you pick $\gamma = 3$ then the function is twice continuously differentiable. So you can pick and choose how differentiable your penalty function is.

Definition: Auxiliary Function and Problem

Assume you are given a positive scalar β

The **auxiliary function** is, for all $\mathbf{x} \in S$,

$$\Psi_\beta(\mathbf{x}) = f(x) + \beta \sum_{i=1}^{m} pos(\mathbf{g}_i(\mathbf{x}))^\gamma$$
$$= f(\mathbf{x}) + \beta * \alpha(\mathbf{x})$$

Ψ_β is continuous, since β is finite.

Each auxiliary function gives us an optmization problem called the **Auxiliary Problem** defined as

$$P_\beta : \min \Psi_\beta(\mathbf{x}),$$
$$\text{s.t. } \mathbf{x} \in S$$

which we're interested in optimizing.

We can see here that if $\beta = \infty$ then solving this problem is essentially the same as solving the constrained problem. That's not quite applicable though, because if β is ∞, then our function is not continuous. However, that notion is the general idea of this setup, (to punish infeasibility while we optimize).

Now, we have a simple algorithm sketch which we claim solves this problem: Note that our $\beta = \delta^k$ is growing each iteration.

Algorithm 7: Penalty Function Algorithm

Input: $\delta > 1, \gamma$, tolerance τ
Output: $\mathbf{x}$ that is "feasible enough"
for $k = 1, 2, 3, \ldots$, *until termination condition* **do**
  $\quad \mathbf{x} \leftarrow$ solution to *Auxiliary Problem* P_{δ^k} ;
  $\quad$ Terminate if $\delta^k * \alpha(\mathbf{x})$ less than τ ;

This is sometimes called the "exterior penalty function methodology" since most of the $\mathbf{x}$ being considered are *outside* of the feasible region of P.

Theorem 10.1

All notation as above.

For all $\beta > 0$, suppose that the *auxiliary problem* P_β has a global min, which we name $\mathbf{x}_\beta$. Suppose, also, that P also has a global min, $\mathbf{x}_\infty$. Suppose further that $\{\mathbf{x}_\beta\}_{\beta>0}$ are contained in a compact subset of S (*compactness* here just means closed and bounded). In particular, this implies that $\exists$ a convergent sequence $\mathbf{x}_{\beta_k} \to \bar{\mathbf{x}} \in S$ as $\beta_k \to \infty$ (section 1.2).

Note: If S is compact, then all of the above happens! If S is compact, and we know that $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ defines a closed set, then the intersection of these sets if compact!

Then we have all of the following:
  i) a) $\alpha(\mathbf{x}_\beta)$ decreases as $\beta \to \infty$.
    b) $f(\mathbf{x}_\beta)$ increases as $\beta \to \infty$.
    c) $\Psi_\beta(\mathbf{x}_\beta)$ increases as $\beta \to \infty$.
  ii) $\alpha(\mathbf{x}_\beta) \to 0$ as $\beta \to \infty$
  iii) $\sup_{\beta>0} \Psi_\beta(\mathbf{x}_\beta) = \lim_{\beta\to\infty}\Psi_\beta(\mathbf{x}_\beta) = f(\mathbf{x}_\infty)$
  iv) $\bar{\mathbf{x}}$ is a global min of P
  v) $\beta * \alpha(\mathbf{x}_\beta) \to 0$ as $\beta \to \infty$

We have 5 (or really, 8) things to prove here, which we'll do in the order presented in the Theorem.

(i). To show (i), note that for any $0 < \beta' < \beta''$ then

$$f(\mathbf{x}_{\beta''}) + \beta' * \alpha(\mathbf{x}_{\beta''}) \geq^{(*)} f(\mathbf{x}_{\beta'}) + \beta' * \alpha(\mathbf{x}_{\beta'})$$

by definition of $\mathbf{x}_{\beta'}$ and similarly

$$f(\mathbf{x}_{\beta'}) + \beta'' * \alpha(\mathbf{x}_{\beta'}) \geq f(\mathbf{x}_{\beta''}) + \beta'' * \alpha(\mathbf{x}_{\beta''})$$

again by definition of $\mathbf{x}_{\beta''}$. If we add these inequalities, this yields

$$(\beta'' - \beta')\left(\alpha(\mathbf{x}_{\beta'}) - \alpha(\mathbf{x}_{\beta''})\right) \geq 0$$

We know that $\beta'' - \beta' > 0$ and so the second term here *must* be ≥ 0. Thus $\alpha(\mathbf{x}_\beta)$ decreases as β increases, as claimed in (i.a). By (i.a) and inequality (*) we have

$$f(\mathbf{x}_{\beta''}) + \beta' * \alpha(\mathbf{x}_{\beta'}) \geq f(\mathbf{x}_{\beta''}) + \beta' * \alpha(\mathbf{x}_{\beta''}) \geq f(\mathbf{x}_{\beta'}) + \beta' * \alpha(\mathbf{x}_{\beta'})$$

Note the $\alpha(\mathbf{x}_{\beta'})$ in the first equation. We can say this because we know (from (i.a)) that $\alpha(\mathbf{x}_{\beta'}) > \alpha(\mathbf{x}_{\beta''})$, since $\beta'' > \beta'$. In this form, we can *cancel out* the $\alpha(\mathbf{x}_{\beta'})$ on either side, to obtain the inequality

$$f(\mathbf{x}_{\beta''}) \geq f(\mathbf{x}_{\beta'})$$

which shows that $f(\mathbf{x}_\beta)$ increases as β increases, as stated in (i.b). Next note that:

$$\Psi_{\beta''}(\mathbf{x}_{\beta''}) = f(\mathbf{x}_{\beta''}) + \beta'' * \alpha(\mathbf{x}_{\beta''})$$
$$\geq f(\mathbf{x}_{\beta''}) + \beta'' * \alpha(\mathbf{x}_{\beta''}) + (\beta' - \beta'')\alpha(\mathbf{x}_{\beta''}) \quad \text{since } \beta' < \beta''$$
$$= f(\mathbf{x}_{\beta''}) + \beta' * \alpha(\mathbf{x}_{\beta''})$$
$$\geq^{(*)} f(\mathbf{x}_{\beta'}) + \beta' * \alpha(\mathbf{x}_{\beta'})$$
$$= \Psi'_\beta(\mathbf{x}_{\beta'})$$

which finally shows that $\Psi(\mathbf{x}_\beta)$ is increasing as β increases, as claimed in (i.c). ∎

(ii). For any $\epsilon > 0$, we *claim* that for every β such that

$$\beta > \frac{1}{\epsilon}|f(\mathbf{x}_\infty) - f(\mathbf{x}_{\beta=1})| + 2$$

then $\alpha(\mathbf{x}_\beta) \leq \epsilon$. Note that if we show that this claim holds, then this proves *(ii)*, since by *(ia)* and the fact that epsilon is arbitrary, then $\alpha(\mathbf{x}_\beta) \to 0$.

Suppose the above is *not true*, i.e.

$$\exists \beta \text{ s.t. } \beta > \frac{1}{\epsilon}|f(\mathbf{x}_\infty) - f(\mathbf{x}_{\beta=1})| + 2 \text{ and } \alpha(\mathbf{x}_\beta) > \epsilon$$

Then

$$f(\mathbf{x}_\infty) = \Psi_\beta(\mathbf{x}_\infty) \qquad \text{because } \mathbf{x}_\infty \text{ is feasible, and thus has no penalty}$$
$$\geq \Psi_\beta(\mathbf{x}_\beta) \qquad \text{Because } \mathbf{x}_\beta \text{ is optimal in } \Psi_\beta$$
$$= f(\mathbf{x}_\beta) + \beta * \alpha(\mathbf{x}_\beta)$$
$$\geq f(\mathbf{x}_1) + \beta * \alpha(\mathbf{x}_\beta) \qquad \text{because } f(\mathbf{x}_\beta) \text{ increases as beta increases (ib)}$$
$$> f(\mathbf{x}_1) + |f(\mathbf{x}_\infty) - f(\mathbf{x}_1)| + 2\epsilon \qquad \text{by note below}$$
$$> f(\mathbf{x}_\infty)$$

Note: This last inequality is due to the following arithmetic using our assumptions above:

$$\beta\alpha(\mathbf{x}_\beta) \geq \epsilon\beta > \epsilon * \frac{1}{\epsilon}|f(\mathbf{x}_\infty) - f(\mathbf{x}_1)| + \epsilon * 2$$

This series of inequalities is a contradiction, showing that our initial claim is correct and thus *(ii)* holds. ∎

(iii) & (iv). Recall that α is a continuous function. Then, $\mathbf{x}_{\beta_k} \to \bar{\mathbf{x}}$ as $\beta_k \to \infty$ (by compactness), and *(ii)* together imply that $\alpha(\bar{\mathbf{x}}) = 0$, i.e. $\bar{\mathbf{x}}$ is feasible in P. Then:

$$\forall \beta > 0, f(\mathbf{x}_\infty) = \Psi_\beta(\mathbf{x}_\infty) \geq \Psi_\beta(\mathbf{x}_\beta) \implies f(\mathbf{x}_\infty) \geq \sup_{\beta>0} \Psi_\beta(\mathbf{x}_\beta)$$
$$\implies f(\mathbf{x}_\infty) \geq \sup_{\beta>0} \Psi(\beta(\mathbf{x}_\beta) \geq f(\mathbf{x}_{\beta_k}), \forall k$$
$$\text{(by the continuity of } f) \implies f(\mathbf{x}_\infty) \geq^{(1)} \sup_{\beta>0} \Psi_\beta(\mathbf{x}_\beta) \geq f(\bar{\mathbf{x}}) \geq^{(2)} f(\mathbf{x}_\infty)$$

Note that we then must have equality throughout this last line. Equality in inequality (1) shows *(iii)* and equality in inequality (2) shows *(iv)*. ∎

(v). First, note that

$$\Psi_\beta(\mathbf{x}_\beta) - f(\mathbf{x}_\beta) = \beta * \alpha(\mathbf{x}_\beta)$$

Now, recall that, as shown in *(iii)*, the $\lim_{\beta\to\infty} \Psi_\beta(\mathbf{x}_\beta) = f(\mathbf{x}_\infty)$. Then, by *(i.b) & (iv)*,

$$\lim_{\beta\to\infty} f(\mathbf{x}_\beta) = f(\bar{\mathbf{x}}) = f(\mathbf{x}_\infty) \implies \Psi_\beta(\mathbf{x}_\beta) - f(\mathbf{x}_\beta) \to f(\mathbf{x}_\infty) - f(\mathbf{x}_\infty) = 0 \text{ as } \beta \to \infty$$
$$\implies \beta * \alpha(\mathbf{x}_\beta) \to 0 \text{ as } \beta \to \infty$$

which finally shows *(v)*. ∎

Finally, we have now shown all the claims in the theorem, and thus the theorem is proven.

10.2 Barrier Methods

Lastly, we will be considering *Barrier Methods* of optimization, which are sometimed also called "*interior penalty function method*". Similar to penalty methods, this will be a way of fitting our constrained optimization problems into an unconstrained setting. However, whereas in penalty methods we were largely considering points that were not feasible, or *exterior*, in barrier methods we will find our way through feasible points, or *interior points*, until we find ourselves on the boundary of our region.

Consider the problem P:

$$\min f(\mathbf{x}) \text{ s.t. } \mathbf{g}(\mathbf{x}) \leq \mathbf{0}$$
$$\mathbf{x} \in S$$

where $S \subseteq \mathbb{R}^n$ is non-empty, and all functions f, g_i are continuous.

Definition: Barrier Function

Define $\Lambda : (-\infty, 0) \to \mathbb{R}$ be a continuous, non-negative function such that

$$\lim_{t \to 0} \Lambda(t) = \infty$$

A **barrier function** $B : \{\mathbf{x} \in S : \mathbf{g}(\mathbf{x}) < \mathbf{0}\} \to \mathbb{R}$ is defined by

$$B(\mathbf{x}) = \sum_{i=1}^{m} \Lambda(\mathbf{g}_i(\mathbf{x}))$$

Note: As a $\mathbf{g}_i(\mathbf{x})$ get's closer to 0, the value of $\Lambda(\mathbf{g}_i(\mathbf{x}))$ get's closer to ∞. As we get closer and closer to the boundaries of our region, the barrier value goes up.

Examples of Λ:

$$B(\mathbf{x}) = -\sum_{i=1}^{m} \frac{1}{\mathbf{g}(\mathbf{x})}$$

and

$$B(\mathbf{x}) = -\sum_{i=1}^{m} ln(\min\{-\mathbf{g}_i(\mathbf{x}), 1\})$$

Note: This last one, we can think of as having 0 value for each constraint whose value is more negative than -1, and otherwise barrier values start to occur.

Definition: Barrier Auxiliary Function and Problem

Given any barrier function B and a scalar $\beta > 0$, the **auxiliary function**, $\Psi_\beta : \{\mathbf{x} \in S : \mathbf{g}(\mathbf{x}) < \mathbf{0}\} \to \mathbb{R}$, is defined by

$$\Psi_\beta(\mathbf{x}) = f(\mathbf{x}) + \beta * B(\mathbf{x})$$

The **auxiliary problem**, then is defined as P_β :

$$\min \Psi_\beta(\mathbf{x}) \text{ s.t. } \mathbf{g}(\mathbf{x}) < \mathbf{0}$$
$$\mathbf{x} \in S$$

Note: In this formulation, we have made our feasible region something of an open set. Now, we can utilize un-constrained optimization techniques inside this set, since the Barrier Function will penalize us heavily for approaching the tips of our feasible region. Then, as we dial down the β parameter, we are allowed to approach the barrier.

This yields the **Barrier Function Algorithm** (Algorithm 8).

Algorithm 8: Barrier Function Algorithm

Input: Barrier Function B, δ s.t. $0 < \delta < 1$, positive tolerance τ
Output: $\mathbf{x}^*$ optimal in P
for $k = 1, 2, 3, \ldots$ *until terminate* **do**

    Define auxiliary problem P_{δ^k} and solve (using unconstrained optimization techniques);

    $\bar{\mathbf{x}} \leftarrow$ solution to P_{δ^k} ;

    Terminate if $\delta^k * B(\mathbf{x})$ is less than τ;