Course:

Optimization III
Convex Analysis
Nonlinear Programming Theory
Nonlinear Programming Algorithms

ISyE 6663 Spring 2022

Lecturer: Dr. Arkadi Nemirovski https://www.isye.gatech.edu/~nemirovs
nemirovs@isye.gatech.edu
Office hours: virtual (BlueJeans), Thursday 10:00-12:00; link:
https://gatech.bluejeans.com/657034900
In-person meetings (Groseclose 446) by preliminary appointments only.

- **Teaching Assistant:** Georgios Kotsalis, gkotsalis3@gatech.edu

Office hours: virtual (BlueJeans), Monday 3:30-4:30 pm
In-person meetings (Main 422) by preliminary appointments only.

- **Classes:** Monday, Wednesday 2:00-3:15 pm, IC 105
- **Kaltura Recordings, Lecture Notes, Transparencies, Assignments:**

course site on Canvas and
https://www2.isye.gatech.edu/~nemirovs/OPTIIILN2022.pdf
https://www2.isye.gatech.edu/~nemirovs/OPTIIITRSpring2022.pdf
https://www2.isye.gatech.edu/~nemirovs/OPTIIIAssignments2022.pdf

Grading Policy:

Assignments	5%
Midterm exam	35%
Final exam	60%

- *Both MidTerm and Final exams will be take-home and will be posted at least 5 days prior to the respective submission deadlines.*
- *Submission of homeworks and take-home exams digital only. No hard-copies!*

♣ To make decisions optimally is one of the most basic desires of a human being.

Whenever the candidate decisions, design restrictions and design goals can be properly quantified, optimal decision-making yields an *optimization problem*, most typically, a *Mathematical Programming* one:

$$\begin{array}{rcll} \text{minimize} & & & \\ & & f(x) & [\text{ objective }] \\ \text{subject to} & & & \\ h_i(x) & = & 0,\ i = 1, ..., m & \left[\begin{array}{l}\text{equality}\\ \text{constraints}\end{array}\right] \\ g_j(x) & \leq & 0,\ j = 1, ..., k & \left[\begin{array}{l}\text{inequality}\\ \text{constraints}\end{array}\right] \\ x & \in & X & [\text{ domain }] \end{array} \qquad \text{(MP)}$$

♣ In (MP),

◇ a *solution* $x \in \mathbb{R}^n$ represents a candidate decision,

◇ the *constraints* express restrictions on the meaningful decisions (balance and state equations, bounds on resources, etc.),

◇ the *objective* to be minimized represents the losses (minus profit) associated with a decision.

$$
\begin{array}{lcll}
\text{minimize} & & & \\
 & f(x) & & [\text{ objective }] \\
\text{subject to} & & & \\
h_i(x) & = & 0,\ i = 1, ..., m & \left[\begin{array}{l}\text{equality}\\ \text{constraints}\end{array}\right] \\
g_j(x) & \leq & 0,\ j = 1, ..., k & \left[\begin{array}{l}\text{inequality}\\ \text{constraints}\end{array}\right] \\
x & \in & X & [\text{ domain }]
\end{array}
\qquad \text{(MP)}
$$

♣ To solve problem (MP) means to find its *optimal solution* x_*, that is, a *feasible* (i.e., satisfying the constraints) solution with the value of the objective $\leq$ its value at any other feasible solution:

$$
x_* : \left\{ \begin{array}{l}
h_i(x_*) = 0\,\forall i \ \&\ g_j(x_*) \leq 0\,\forall j \ \&\ x_* \in X \\
h_i(x) = 0\,\forall i \ \&\ g_j(x) \leq 0\,\forall j \ \&\ x \in X \\
\qquad\qquad\qquad\qquad \Rightarrow f(x_*) \leq f(x)
\end{array} \right.
$$

$$
\begin{array}{rcl}
 & & \min\limits_{x} f(x) \\
\text{s.t.} & & \\
h_i(x) & = & 0,\ i = 1, ..., m \\
g_j(x) & \leq & 0,\ j = 1, ..., k \\
x & \in & X
\end{array}
\qquad \text{(MP)}
$$

♣ In *Combinatorial* (or *Discrete*) Optimization, the domain X is a discrete set, like the set of all integral or 0/1 vectors.

In contrast to this, in *Continuous* Optimization we will focus on, X is a "continuum" set like the entire $\mathbb{R}^n$, a *box* $\{x : a \leq x \leq b\}$, or *simplex* $\{x \geq 0 : \sum\limits_j x_j = 1\}$, etc., and the objective and the constraints are (at least) continuous on X.

♣ In *Linear Programming*, $X = \mathbb{R}^n$ and the objective and the constraints are linear functions of x.
In contrast to this, in *Nonlinear Continuous Optimization*, the objective and the constraints can be nonlinear functions.

$$
\begin{array}{rcl}
 & & \min\limits_{x} f(x) \\
\text{s.t.} & & \\
h_i(x) & = & 0,\ i = 1, ..., m \\
g_j(x) & \leq & 0,\ j = 1, ..., k \\
x & \in & X
\end{array}
\qquad \text{(MP)}
$$

♣ The goals of our course is to present

- *basic theory* of Continuous Optimization, with emphasis on *existence* and *uniqueness* of optimal solutions and their *characterization* (i.e., necessary and/or sufficient optimality conditions);

- *traditional algorithms* for building (approximate) optimal solutions to Continuous Optimization problems.

♣ *Mathematical foundation* of Optimization Theory is given by *Convex Analysis* – a specific combination of Real Analysis and Geometry unified by and focusing on investigating convexity-related notions.

Part I

Continuous Optimization: Basic Theory

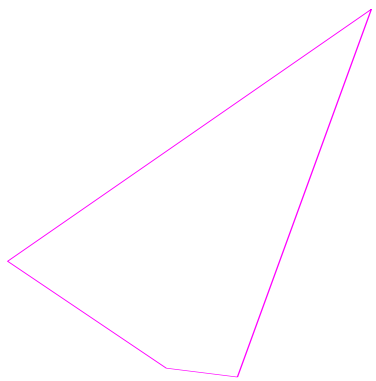
Lecture 1:
Convex Sets, I

Convex Sets

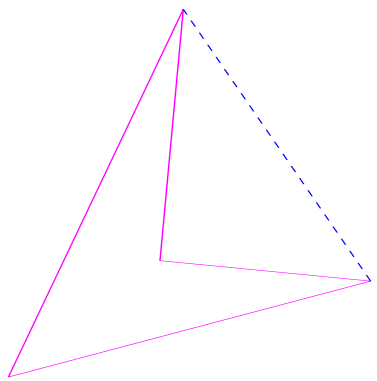
<u>Definition.</u> A set $X \subset \mathbb{R}^n$ is called *convex*, if X contains, along with every pair x, y of its points, the entire segment $[x, y]$ with the endpoints x, y:

$$x, y \in X \Rightarrow (1-\lambda)x + \lambda y \in X \ \forall \lambda \in [0, 1].$$

<u>Note:</u> when λ runs through $[0, 1]$, the point $(1-\lambda)x + \lambda y \equiv x + \lambda(y - x)$ runs through the segment $[x, y]$.



The set in $\mathbb{R}^2$ bounded by red line is convex



The set in $\mathbb{R}^2$ bounded by red line is *not* convex

♣ Immediate examples of convex sets in $\mathbb{R}^n$:

- $\mathbb{R}^n$
- $\emptyset$
- singleton $\{x\}$.

Examples of convex sets, I: Affine sets

<u>Definition:</u> *Affine set* M in $\mathbb{R}^n$ is a set which can be obtained as a shift of a *linear subspace* $L \subset \mathbb{R}^n$ by a vector $a \in \mathbb{R}^n$:

$$M = a + L = \{x = a + y : y \in L\} \tag{1}$$

<u>Note:</u> I. The linear subspace L is uniquely defined by affine subspace M and is the set of differences of vectors from M:

$$(1) \Rightarrow L = M - M = \{y = x' - x'' : x', x'' \in M\}$$

II. The shift vector a is *not* uniquely defined by affine subspace M; in (1), one can take as a *every* vector from M (and *only* vector from M):

$$(1) \Rightarrow M = a' + L \; \forall a' \in M.$$

III. Generic example of affine subspace: the set of solutions of a *solvable* system of linear equations:

$$M \text{ is affine subspace in } \mathbb{R}^n$$
$$\Updownarrow$$
$$\emptyset \neq M \equiv \{x \in \mathbb{R}^n : Ax = b\} \equiv \underbrace{a}_{Aa=b} + \underbrace{\{x : Ax = 0\}}_{\mathrm{Ker}A}$$

♣ By **III**, affine subspace is convex, due to

Proposition. *The solution set of an arbitrary (finite or infinite) system of linear inequalities is convex:*

$$X = \{x \in \mathbb{R}^n : a_\alpha^T x \le b_\alpha, \alpha \in \mathcal{A}\} \Rightarrow X \text{ is convex}$$

In particular, every polyhedral set $\{x : Ax \le b\}$ *is convex.*

Proof:

$$\begin{aligned}
& x, y \in X, \lambda \in [0,1] \\
\Leftrightarrow\ & a_\alpha^T x \le b_\alpha,\ a_\alpha^T y \le b_\alpha \forall \alpha \in \mathcal{A},\ \lambda \in [0,1] \\
\Rightarrow\ & \underbrace{\lambda a_\alpha^T x + (1-\lambda) a_\alpha^T y}_{a_\alpha^T[\lambda x + (1-\lambda) y]} \le \underbrace{\lambda b_\alpha + (1-\lambda) b_\alpha}_{b_\alpha}\ \forall \alpha \in \mathcal{A} \\
\Rightarrow\ & [\lambda x + (1-\lambda) y] \in X\ \forall \lambda \in [0,1].
\end{aligned}$$

Remark: *Proposition remains valid when part of the nonstrict inequalities* $a_\alpha^T x \le b_\alpha$ *are replaced with their strict versions* $a_\alpha^T x < b_\alpha$.

Remark: *The solution set*

$$X = \{x : a_\alpha^T x \le b_\alpha, \alpha \in \mathcal{A}\}$$

of a system of nonstrict inequalities is not only convex, it is closed (i.e., contains limits of all converging sequences $\{x_i \in X\}_{i=1}^\infty$ *of points from* X*).*

We shall see in the mean time that

- Vice versa, *every closed and convex set* $X \subset \mathbb{R}^n$ *is the solution set of an appropriate countable system of nonstrict linear inequalities:*

$$X \text{ is closed and convex}$$
$$\Downarrow$$
$$X = \{x : a_i^T x \le b_i,\ i = 1, 2, ...\}$$

Examples of convex sets, II: Unit balls of norms

Definition: A real-valued function $\|x\|$ on $\mathbb{R}^n$ is called *a norm*, if it possesses the following three properties:

◇ [positivity] $\|x\| \geq 0$ for all x and $\|x\| = 0$ iff $x = 0$;

◇ [homogeneity] $\|\lambda x\| = |\lambda|\|x\|$ for all vectors x and reals λ;

◇ [triangle inequality] $\|x + y\| \leq \|x\| + \|y\|$ for all vectors x, y.

Proposition: *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. The unit ball of this norm – the set $\{x : \|x\| \leq 1\}$, same as any other $\|\cdot\|$-ball $\{x : \|x - a\| \leq r\}$, is convex.*

Proof:

$$\|x - a\| \leq r, \|y - a\| \leq r, \lambda \in [0, 1]$$

$$\begin{aligned} \Rightarrow r &\geq \lambda\|x - a\| + (1 - \lambda)\|y - a\| = \|\lambda(x - a)\| + \|(1 - \lambda)(y - a)\| \\ &\geq \|\lambda(x - a) + (1 - \lambda)(y - a)\| = \|[\lambda x + (1 - \lambda)y] - a\| \end{aligned}$$

$$\Rightarrow \quad \|[\lambda x + (1 - \lambda)y] - a\| \leq r \ \forall \lambda \in [0, 1].$$

Standard examples of norms on $\mathbb{R}^n$: ℓ_p-norms

$$\|x\|_p = \begin{cases} \left(\sum\limits_{i=1}^{n} |x_i|^p\right)^{1/p}, & 1 \le p < \infty \\ \max\limits_i |x_i|, & p = \infty \end{cases}$$

Note: • $\|x\|_2 = \sqrt{\sum\limits_i x_i^2}$ is the standard Euclidean norm;

• $\|x\|_1 = \sum\limits_i |x_i|$;

• $\|x\|_\infty = \max\limits_i |x_i|$ (uniform norm).

Note: except for the cases $p = 1$ and $p = \infty$, triangle inequality for $\|\cdot\|_p$ requires a nontrivial proof!

Proposition [characterization of $\|\cdot\|$-balls] *A set V in $\mathbb{R}^n$ is the unit ball of a norm iff V is*

(a) convex and symmetric w.r.t. 0: $V = -V$,

(b) bounded and closed, and

(c) contains a neighbourhood of the origin.

Fact: A norm $\|\cdot\|$ norm on $\mathbb{R}^n$ defines a metrics $d(x,y) = \|x-y\|$ satisfying the usual axioms of metrics:

- [positivity] $d(x,y) \geq 0$ and $d(x,y) = 0 \Leftrightarrow x = y$
- [symmetry] $d(x,y) \equiv d(y,x)$
- [triangle inequality] $d(x,y) + d(y,z) \geq d(x,z)$

and linked to the linear structure of $\mathbb{R}^n$ by

- [shift invariance] $d(x+a, y+a) \equiv d(x,y)$
- [homogeneity] $d(\lambda x, \lambda y) = |\lambda| d(x,y)$.

Fact: As every metrics, $d(x,y) = \|x-y\|$ specifies the notion of convergence: by definition, *a sequence of vectors* $\{x^t \in \mathbb{R}^n\}_{t\geq 1}$ *converges to vector* $\bar{x} \in \mathbb{R}^n$ *as* $t \to \infty$ (notation: $\bar{x} = \lim_{t\to\infty} x^t$) *iff* $\lim_{t\to\infty} \|x^t - \bar{x}\| = 0$.

Fact: Every two norms $\|\cdot\|$, $\|\cdot\|'$ on $\mathbb{R}^n$ are equivalent: for some positive constant c (depending on the norms), one has $\forall (x \neq 0) : c^{-1} \leq \frac{\|x\|}{\|x\|'} \leq c$

$\Rightarrow$ *All norms on* $\mathbb{R}^n$ *specify the same convergence; in particular,* $\lim_{t\to\infty} x^t = \bar{x}$ *iff* $\lim_{t\to\infty} x_i^t = \bar{x}_i$ *for all* $i = 1, ..., n$.

Similarly, *All norms on* $\mathbb{R}^n$ *specify the same notion of boundedness of a subset of* $\mathbb{R}^n$ (recall that a set X in metric space is called bounded is the distances between all pairs of its points form a bounded set on the axis).

Note: Equivalence of all norms on a linear space is a characteristic property of *finite dimensional* linear spaces.

Proof of norm equivalence: Clearly, it suffices to prove that every norm $\|\cdot\|$ on $\mathbb{R}^n$ is equivalent to the norm $\|x\|_1 = \sum_i |x_i|$.

• Given a norm $\|\cdot\|$ on $\mathbb{R}^n$, let e_i, $i \leq n$, be the standard basic orths in $\mathbb{R}^n$, and let $C = \max_i \|e_i\|$. Then

$$\forall x \in \mathbb{R}^n : \|x\| = \|\sum_i x_i e_i\| \leq \sum_i \|x_i e_i\| = \sum_i |x_i| \|e_i\| \leq C\|x\|_1 \qquad (a)$$

• Now let $S = \{x \in \mathbb{R}^n : \sum_{j=1}^n |x_j| = 1\}$. Given a sequence of points $\{x^t\}$ of S, observe that the sequences of reals $\{x_i^t\}_{t\geq 1}$, $i = 1, ..., n$, are bounded, implying that we can find a subsequence $\{x^{t_j}\}_{j\geq 1}$, $j_1 < j_2 < ...$, which converges coordinate-wise to some vector $\bar{x}$ which clearly belongs to S along with all vectors x^{t_j}. Besides this, the subsequence in question converges to $\bar{x}$ coordinate-wise and therefore converges to $\bar{x}$ in the metrics $d_1(\cdot,\cdot)$ stemming from $\|\cdot\|_1$.

$\Rightarrow$ *Equipping S with metrics $d_1(\cdot,\cdot)$, we obtain compact metric space — from every sequence of points from S one can select a subsequence converging to a point from the set.*

• Observe that by (a) the function $f(x) = \|x\|$ is continuous on the just defined metric space: $|f(x) - f(y)| \leq \|x - y\| \leq Cd_1(x, y)$ for all $x, y \in X$ (the first $\leq$ is due to $\|x\| \leq \|y\| + \|x - y\|$ and $\|y\| \leq \|x\| + \|y - x\| = \|x\| + \|x - y\|$).

$\Rightarrow$ By Weierstrass Theorem, continuous function f on compact metric space $(S, d_1(\cdot,\cdot))$ attains it minimum on S. Since S does not contain origin, this minimum c is positive: $\forall(x, \|x\|_1 = 1) : \|x\| \geq c > 0$, implying by homogeneity that

$$\forall(x \neq 0) : \|x\|/\|x\|_1 \geq c > 0 \qquad (b)$$

(a) and (b) together say that $\|\cdot\|$ is equivalent to $\|\cdot\|_1$ $\square$

Examples of convex sets, III: Ellipsoid

Definition: An ellipsoid in $\mathbb{R}^n$ is a set X given by

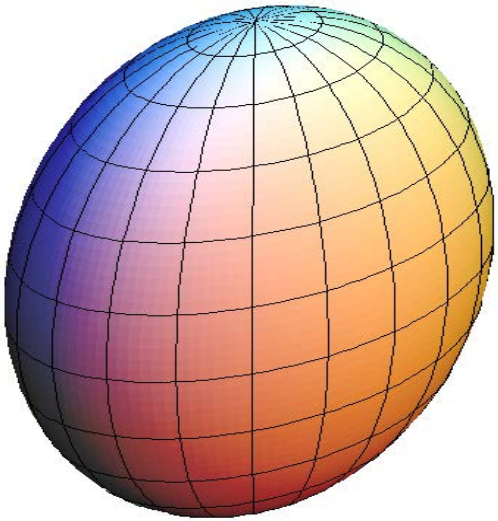
$\diamondsuit$ positive definite and symmetric $n \times n$ matrix Q (that is, $Q = Q^T$ and $u^T Q u > 0$ whenever $u \neq 0$),

$\diamondsuit$ *center* $a \in \mathbb{R}^n$,

$\diamondsuit$ *radius* $r > 0$

via the relation

$$X = \{x : (x-a)^T Q (x-a) \leq r^2\}.$$



3D ellipsoid

$$X = \{x : (x-a)^T Q(x-a) \le r^2\}.$$

Proposition: *An ellipsoid is convex.*

Proof: Since Q is symmetric positive definite, by Linear Algebra $Q = (Q^{1/2})^2$ for uniquely defined symmetric positive definite matrix $Q^{1/2}$. Setting $\|x\|_Q = \|Q^{1/2}x\|_2$, we clearly get a norm on $\mathbb{R}^n$ (since $\|\cdot\|_2$ is a norm and $Q^{1/2}$ is nonsingular). We have

$$\begin{aligned}(x-a)^T Q(x-a) &= [(x-a)^T Q^{1/2}][Q^{1/2}(x-a)] \\ &= \|Q^{1/2}(x-a)\|_2^2 = \|x-a\|_Q^2,\end{aligned}$$

so that X is a $\|\cdot\|_Q$-ball and is therefore a convex set.

Examples of convex sets, IV: ϵ-neighbourhood of convex set

Proposition: *Let M be a nonempty convex set in $\mathbb{R}^n$, $\|\cdot\|$ be a norm, and $\epsilon \geq 0$. Then the set*

$$X = \{x : \text{dist}_{\|\cdot\|}(x, M) \equiv \inf_{y \in M} \|x - y\| \leq \epsilon\}$$

is convex.

Proof: $x \in X$ if and only if for every $\epsilon' > \epsilon$ there exists $y \in M$ such that $\|x - y\| \leq \epsilon'$. We now have

$$\begin{aligned}
& x, y \in X, \lambda \in [0,1] \\
\Rightarrow \quad & \forall \epsilon' > \epsilon \exists u, v \in M : \|x - u\| \leq \epsilon', \|y - v\| \leq \epsilon' \\
\Rightarrow \quad & \forall \epsilon' > \epsilon \exists u, v \in M : \\
& \quad \underbrace{\lambda\|x - u\| + (1-\lambda)\|y - v\|}_{\geq \|[\lambda x + (1-\lambda)y] - [\lambda u + (1-\lambda)v]\|} \leq \epsilon' \ \forall \lambda \in [0,1] \\
\Rightarrow \quad & \forall \epsilon' > \epsilon \, \forall \lambda \in [0,1] \exists w = \lambda u + (1-\lambda)v \in M : \\
& \quad \|[\lambda x + (1-\lambda)y] - w\| \leq \epsilon' \\
\Rightarrow \quad & \lambda x + (1-\lambda)y \in X \ \forall \lambda \in [0,1]
\end{aligned}$$

Convex Combinations and Convex Hulls

<u>Definition:</u> A *convex combination* of m vectors $x_1, ..., x_m \in \mathbb{R}^n$ is their linear combination

$$\sum_i \lambda_i x_i$$

with *nonnegative* coefficients and *unit sum of the coefficients*:

$$\lambda_i \geq 0 \ \forall i, \sum_i \lambda_i = 1.$$

Proposition: *A set $X \subset \mathbb{R}^n$ is convex iff it is closed w.r.t. taking convex combinations of its points:*

$$X \text{ is convex}$$
$$\Updownarrow$$
$$x_i \in X, \lambda_i \geq 0, \sum_i \lambda_i = 1 \Rightarrow \sum_i \lambda_i x_i \in X.$$

Proof, $\Rightarrow$: Assume that X is convex, and let us prove by induction in k that every k-term convex combination of vectors from X belongs to X. Base $k = 1$ is evident. Step $k \Rightarrow k+1$: let $x_1, ..., x_{k+1} \in X$ and $\lambda_i \geq 0$, $\sum_{i=1}^{k+1} \lambda_i = 1$; we should prove that $\sum_{i=1}^{k+1} \lambda_i x_i \in X$. Assume w.l.o.g. that $0 \leq \lambda_{k+1} < 1$. Then

$$\sum_{i=1}^{k+1} \lambda_i x_i = (1 - \lambda_{k+1}) \underbrace{\left(\sum_{i=1}^{k} \frac{\lambda_i}{1 - \lambda_{k+1}} x_i \right)}_{\in X} + \lambda_{k+1} x_{k+1} \in X.$$

Proof, $\Leftarrow$: evident, since the definition of convexity of X is nothing but the requirement for every *2-term* convex combination of points from X to belong to X.

Proposition: *The intersection* $X = \bigcap_{\alpha \in \mathcal{A}} X_\alpha$ *of an arbitrary family* $\{X_\alpha\}_{\alpha \in \mathcal{A}}$ *of convex subsets of* $\mathbb{R}^n$ *is convex.*

Proof: evident.

Corollary: *Let* $X \subset \mathbb{R}^n$ *be an arbitrary set. Then among convex sets containing* X *(which do exist, e.g.* $\mathbb{R}^n$*) there exists the smallest one, namely, the intersection of all convex sets containing* X*.*

Definition: The smallest convex set containing X is called the *convex hull* $\mathrm{Conv}(X)$ of X.

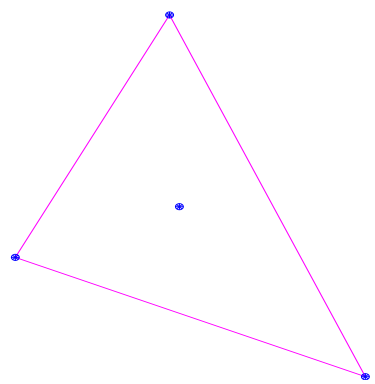
Proposition [convex hull via convex combinations] *For every subset X of $\mathbb{R}^n$, its convex hull* $\mathrm{Conv}(X)$ *is exactly the set $\widehat{X}$ of all convex combinations of points from X.*

Proof. 1) Every convex set which contains X contains every convex combination of points from X as well. Therefore $\mathrm{Conv}(X) \supset \widehat{X}$.
2) It remains to prove that $\mathrm{Conv}(X) \subset \widehat{X}$. To this end, by definition of $\mathrm{Conv}(X)$, it suffices to verify that the set $\widehat{X}$ contains X (evident) and is convex. To see that $\widehat{X}$ is convex, let $x = \sum\limits_i \nu_i x_i$, $y = \sum\limits_i \mu_i x_i$ be two points from $\widehat{X}$ represented as convex combinations of points from X, and let $\lambda \in [0, 1]$. We have

$$\lambda x + (1 - \lambda) y = \sum_i [\lambda \nu_i + (1 - \lambda) \mu_i] x_i,$$

i.e., the left hand side vector is a convex combination of vectors from X.

4-point set in $\mathbb{R}^2$



Convex hull of the set
(red triangle)

Examples of convex sets, V: simplex

<u>Definition:</u> A collection of $m+1$ points x_i, $i = 0, ..., m$, in $\mathbb{R}^n$ is called *affine independent*, if no nontrivial combination of the points *with zero sum of the coefficients* is zero:

$$x_0, ..., x_m \text{ are affine independent}$$
$$\Updownarrow$$
$$\sum_{i=0}^{m} \lambda_i x_i = 0 \ \& \ \sum_i \lambda_i = 0 \Rightarrow \lambda_i = 0, 0 \leq i \leq m$$

<u>Motivation:</u> Let $X \subset \mathbb{R}^n$ be nonempty.

I. The intersection of all affine subspaces containing X is an affine subspace. This clearly is the *smallest* affine subspace containing X; it is called the *affine span* (or affine hull) $\mathrm{Aff}(X)$ of X.

Compare: *The intersection of all linear subspaces containing X is a linear subspace. This clearly is the smallest linear subspace containing X; it is called the linear span* $\mathrm{Lin}(X)$ *of X.*

II. It is easily seen that the affine span $\mathsf{Aff}(X)$ of X is nothing but the set of all *affine combinations* of points from X, that is, linear combinations *with unit sum of coefficients*:

$$\mathsf{Aff}(X) = \{x = \sum_i \lambda_i x_i : x_i \in X, \sum_i \lambda_i = 1\}.$$

Compare: It is easily seen that the linear span $\mathsf{Lin}(X)$ of X is nothing but the set of all *linear combinations* of points from X:

$$\mathsf{Lin}(X) = \{x = \sum_i \lambda_i x_i, x_i \in X\}$$

III. $m+1$ points $x_0, ..., x_m$ are affinely independent iff every point $x \in \mathrm{Aff}(\{x_0, ..., x_m\})$ of their affine span can be *uniquely represented* as an affine combination of $x_0, ..., x_m$:

$$\sum_i \lambda_i x_i = \sum_i \mu_i x_i \ \& \ \sum_i \lambda_i = \sum_i \mu_i = 1 \Rightarrow \lambda_i \equiv \mu_i$$

Compare:

• Vectors $y_1, ..., y_k$ are called linearly independent if no nontrivial linear combination of these vectors is zero:

$$\sum_i \lambda_i y_i = 0 \Rightarrow \lambda_i = 0 \, \forall i$$

• k vectors $y_1, ..., y_k$ are linearly independent iff every point $y \in \mathrm{Lin}(\{y_1, ..., y_k\})$ of their linear span can be *uniquely represented* as a linear combination of $y_1, ..., y_k$:

$$\sum_i \lambda_i y_i = \sum_i \mu_i y_i \ \Rightarrow \lambda_i \equiv \mu_i$$

♣ When $x_0, ..., x_m$ are affinely independent, the coefficients λ_i in the representation

$$x = \sum_{i=0}^{m} \lambda_i x_i \qquad [\sum_i \lambda_i = 1]$$

of a point $x \in M = \mathrm{Aff}(\{x_0, ..., x_m\})$ as an affine combination of $x_0, ..., x_m$ are uniquely defined by x and are called the *barycentric coordinates* of $x \in M$ *taken w.r.t. affine basis* $x_0, ..., x_m$ *of* M.

<u>Definition:</u> m-dimensional simplex Δ with vertices $x_0, ..., x_m$ is the convex hull of $m+1$ affine independent points $x_0, ..., x_m$:

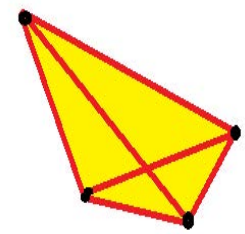
$$\Delta = \Delta(x_0, ..., x_m) = \mathrm{Conv}(\{x_0, ..., x_m\}).$$

<u>Examples:</u> **A.** 2-dimensional simplex is given by 3 points not belonging to a line and is the triangle with vertices at these points.

B. Let $e_1, ..., e_n$ be the standard basic orths in $\mathbb{R}^n$. These n points are affinely independent, and the corresponding $(n-1)$-dimensional simplex is the *standard simplex* $\Delta_n = \{x \in \mathbb{R}^n : x \geq 0, \sum_i x_i = 1\}$.

C. Adding to $e_1, ..., e_n$ the vector $e_0 = 0$, we get $n+1$ affine independent points. The corresponding n-dimensional simplex is
$\Delta_n^+ = \{x \in \mathbb{R}^n : x \geq 0, \sum_i x_i \leq 1\}$.

• *Simplex with vertices $x_0, ..., x_m$ is convex (as a convex hull of a set), and every point from the simplex is a convex combination of the vertices with the coefficients uniquely defined by the point.*

			
point, $m = 0$	segment, $m = 1$	triangle, $m = 2$	tetrahedron, $m = 3$

Simplexes

Examples of convex sets, VI: cone

Definition: A nonempty subset K of $\mathbb{R}^n$ is called *conic*, if it contains, along with every point x, the entire ray emanating from the origin and passing through x:

$$K \text{ is conic}$$
$$\Updownarrow$$
$$K \neq \emptyset \ \& \ \forall (x \in K, t \geq 0) : tx \in K.$$

A *convex conic set* is called a *cone*.

Examples: **A.** *Nonnegative orthant*

$$\mathbb{R}^n_+ = \{x \in \mathbb{R}^n : x \geq 0\}$$

B. *Lorentz cone*

$$\mathbf{L}^n = \{x \in \mathbb{R}^n : x_n \geq \sqrt{x_1^2 + ... + x_{n-1}^2}\}$$

C. *Semidefinite cone* $\mathbf{S}^n_+$. This cone "lives" in the space $\mathbf{S}^n$ of $n \times n$ symmetric matrices and is comprised of all positive semidefinite symmetric $n \times n$ matrices

D. *The solution set* $\{x : a_\alpha^T x \leq 0 \, \forall \alpha \in \mathcal{A}\}$ of an arbitrary (finite or infinite) *homogeneous* system of *nonstrict* linear inequalities is a *closed* cone. In particular, so is a *polyhedral cone* $\{x : Ax \leq 0\}$.

Note: Every *closed* cone in $\mathbb{R}^n$ is the solution set of a countable system of nonstrict *homogeneous* linear inequalities.

Proposition: *A nonempty subset* K *of* $\mathbb{R}^n$ *is a cone iff*

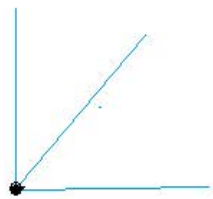
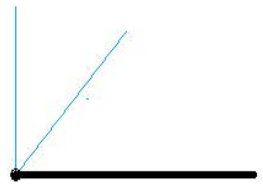
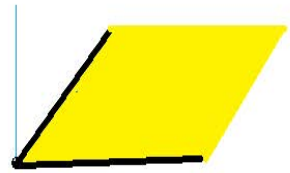
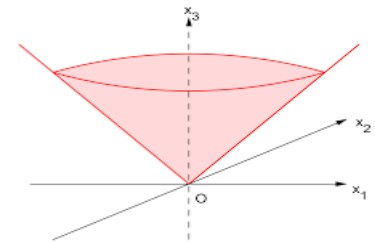
◇ K *is conic:* $x \in K, t \geq 0 \Rightarrow tx \in K$, *and*

◇ K *is closed w.r.t. addition:*

$$x, y \in K \Rightarrow x + y \in K.$$

Proof, $\Rightarrow$: Let K be convex and $x, y \in K$, Then $\frac{1}{2}(x+y) \in K$ by convexity, and since K is conic, we also have $x + y \in K$. Thus, a convex conic set is closed w.r.t. addition.

Proof, $\Leftarrow$: Let K be conic and closed w.r.t. addition. In this case, a convex combination $\lambda x + (1-\lambda)y$ of vectors x, y from K is the sum of the vectors λx and $(1-\lambda)y$ and thus belongs to K, since K is closed w.r.t. addition. Thus, a conic set which is closed w.r.t. addition is convex.

			
Origin $\{x = 0\}$	Ray of x_1-axis $\{x_1 \geq 0, x_2 = x_3 = 0\}$	Quadrant of x_1x_2-plane $\{x_1 \geq 0, x_2 \geq 0, x_3 = 0\}$	Ice-cream cone $\{x_3 \geq \sqrt{x_1^2 + x_2^2}\}$

Several cones in $\mathbb{R}^3$

♣ Cones form an extremely important class of convex sets with properties "parallel" to those of general convex sets. For example,

♢ Intersection of an arbitrary family of cones again is a cone. As a result, *for every nonempty set X, among the cones containing X there exists the smallest cone* Cone(X), *called the conic hull of X.*

♢ A nonempty set is a cone iff it is closed w.r.t. taking *conic* combinations of its elements (i.e., linear combinations with nonnegative coefficients).

♢ The conic hull of a nonempty set X is exactly the set of all conic combinations of elements of X.

"Calculus" of Convex Sets

Proposition. *The following operations preserve convexity of sets:*

1. Intersection: *If* $X_\alpha \subset \mathbb{R}^n$, $\alpha \in \mathcal{A}$, *are convex sets, so is* $\bigcap_{\alpha \in \mathcal{A}} X_\alpha$

2. Direct product: *If* $X_\ell \subset \mathbb{R}^{n_\ell}$, $1 \le \ell \le L$, *are convex sets, so is the set*

$$\begin{array}{rcl} X & = & X_1 \times ... \times X_L \\ & \equiv & \{x = (x^1, ..., x^L) : x^\ell \in X_\ell, 1 \le \ell \le L\} \\ & \subset & \mathbb{R}^{n_1 + ... + n_L} \end{array}$$

3. Taking weighted sums: *If* $X_1, ..., X_L$ *are nonempty convex sets in* $\mathbb{R}^n$ *and* λ_1,...,λ_L *are reals, then the set*

$$\begin{array}{rl} & \lambda_1 X_1 + ... + \lambda_L X_L \\ \equiv & \{x = \lambda_1 x_1 + ... + \lambda_L x_L : x_\ell \in X_\ell, 1 \le \ell \le L\} \end{array}$$

is convex.

4. Affine image: *Let* $X \subset \mathbb{R}^n$ *be convex and* $x \mapsto \mathcal{A}(x) = Ax + b$ *be an affine mapping from* $\mathbb{R}^n$ *to* $\mathbb{R}^k$*. Then the image of* X *under the mapping – the set*

$$\mathcal{A}(X) = \{y = Ax + b : x \in X\}$$

is convex.

5. Inverse affine image: *Let* $X \subset \mathbb{R}^n$ *be convex and* $y \mapsto \mathcal{A}(y) = Ay + b$ *be an affine mapping from* $\mathbb{R}^k$ *to* $\mathbb{R}^n$*. Then the inverse image of* X *under the mapping – the set*

$$\mathcal{A}^{-1}(X) = \{y : Ay + b \in X\}$$

is convex.

Application example: Consider a factory which can utilize at various intensities n types of production processes, consuming k types of resources and producing m types of products. Given the available volumes of resources $r = [r_1; ...; r_k]$ and requested volumes of products $p = [p_1; ...; p_m]$, the management should decide on *production plan* – vector $x = [x_1; ...; x_n]$ of intensities at which the production processes will be used. A production plan $x = [x_1; ...; x_n]$ is feasible if and only if x, r, and p satisfy the system of constraints

$$\begin{array}{rcll} Px & \geq & p & \text{[demand must be satisfied]} \\ Rx & \leq & r & \text{[resource bounds must be obeyed]} \\ x & \in & X & \text{[technological feasibilty constraints]} \end{array} \qquad (S)$$

Assume that the set X of feasible production plans is convex.

Question: *What is the convexity status of the set of implementable pairs* (r, p)*, that is, the set* $\mathcal{RP} = \{(r, p) : \exists x : (x, r, p) \text{ satisfy } (S)\}$ *?*

Answer: $\mathcal{RP}$ is convex.

$$\frac{Px \geq p \quad (a), \qquad Rx \leq r \quad (b), \qquad x \in X \quad (c)}{\mathcal{RP} = \{(r,p) : \exists x : (x,r,p) \text{ satisfy } \ (a),(b),(c)\}}$$

Claim: When X is convex, so is $\mathcal{RP}$.
Indeed,

- the set $\mathcal{S}$ of solutions (x,r,p) to the system of linear constraints $(a),(b)$ is polyhedral and thus convex,

- the set $\mathcal{X} = \{(x,r,p) : x \in X\}$ is the direct product of convex sets X, $\mathbb{R}^k_r$ and $\mathbb{R}^m_p$ and this is convex,

- $\Rightarrow$ the set $\mathcal{XS} = \mathcal{X} \cap \mathcal{S}$ is convex as intersection of two convex sets

- $\Rightarrow$ the set $\mathcal{RP}$ is convex as the image of the set $\mathcal{XS}$ under the linear mapping $(x,r,p) \mapsto (r,p)$.

Nice Topological Properties of Convex Sets

♣ Recall that the set $X \subset \mathbb{R}^n$ is called

♢ *closed*, if X contains limits of all converging sequences of its points:

$$x_i \in X \ \& \ x_i \to x, i \to \infty \Rightarrow x \in X$$

♢ *open*, if it contains, along with every of its points x, a ball of a positive radius centered at x:

$$x \in X \Rightarrow \exists r > 0 : \{y : \|y - x\|_2 \leq r\} \subset X.$$

E.g., the solution set of an arbitrary system of *nonstrict* linear inequalities $\{x : a_\alpha^T x \leq b_\alpha\}$ is closed; the solution set of *finite* system of *strict* linear inequalities $\{x : Ax < b\}$ is open.

<u>Facts:</u> A. X is closed iff $\mathbb{R}^n \backslash X$ is open

B. The intersection of an arbitrary family of closed sets and the union of a finite family of closed sets are closed

B′. The union of an arbitrary family of open sets and the intersection of a finite family of open sets are open

$\diamondsuit$ From **B** it follows that *the intersection of all closed sets containing a given set X is closed*; this intersection, called *the closure* $\mathrm{cl}X$ of X, is the smallest closed set containing X. $\mathrm{cl}X$ is exactly the set of limits of all converging sequences of points of X:

$$\mathrm{cl}X = \{x : \exists x_i \in X : x = \lim_{i\to\infty} x_i\}.$$

$\diamondsuit$ From **B**$'$ it follows that *the union of all open sets contained in a given set X is open*; this union, called *the interior* $\mathrm{int}\,X$ of X, is the largest open set contained in X. $\mathrm{int}\,X$ is exactly the set of all *interior* points of X – points x belonging to X along with balls of positive radii centered at the points:

$$\mathrm{int}\,X = \{x : \exists r > 0 : \{y : \|y - x\|_2 \le r\} \subset X\}.$$

$\diamondsuit$ Let $X \subset \mathbb{R}^n$. Then $\mathrm{int}\,X \subset X \subset \mathrm{cl}X$. The "difference" $\partial X = \mathrm{cl}X \backslash \mathrm{int}\,X$ is called the *boundary* of X; boundary always is closed (as the intersection of the closed sets $\mathrm{cl}X$ and the complement of $\mathrm{int}\,X$).

$$\mathrm{int}\, X \subset X \subset \mathrm{cl} X \qquad (*)$$

♣ In general, the discrepancy between $\mathrm{int}\, X$ and $\mathrm{cl} X$ can be pretty large. E.g., let $X \subset \mathbb{R}^1$ be the set of irrational numbers in $[0,1]$. Then $\mathrm{int}\, X = \emptyset$, $\mathrm{cl} X = [0,1]$, so that $\mathrm{int}\, X$ and $\mathrm{cl} X$ differ dramatically.

♣ Fortunately, a *convex* set is perfectly well approximated by its closure (and by interior, if the latter is nonempty).

Proposition: *Let $X \subset \mathbb{R}^n$ be a nonempty convex set. Then*
(i) Both $\mathrm{int}\, X$ and $\mathrm{cl} X$ are convex
(ii) If $\mathrm{int}\, X$ is nonempty, then $\mathrm{int}\, X$ is dense in $\mathrm{cl} X$, density of a set Y in a set X meaning that every point from X can be approximated to whatever high accuracy by points of Y. Formally: *Y is dense in X $\Leftrightarrow$ Every point from X is the limit of a converging sequence of points from Y.*
Moreover,

$$\begin{array}{l} x \in \mathrm{int}\, X,\ y \in \mathrm{cl} X \Rightarrow \\ \qquad \lambda x + (1-\lambda) y \in \mathrm{int}\, X \ \forall \lambda \in (0,1] \end{array} \qquad (!)$$

• **Claim (i)**: Let X be convex. Then both $\operatorname{int} X$ and $\operatorname{cl} X$ are convex

Proof. (i) is nearly evident. Indeed, to prove that $\operatorname{int} X$ is convex, note that for every two points $x, y \in \operatorname{int} X$ there exists a common $r > 0$ such that the balls B_x, B_y of radius r centered at x and y belong to X. Since X is convex, for every $\lambda \in [0,1]$ X contains the set $\lambda B_x + (1-\lambda) B_y$, which clearly is nothing but the ball of the radius r centered at $\lambda x + (1-\lambda) y$. Thus, $\lambda x + (1-\lambda) y \in \operatorname{int} X$ for all $\lambda \in [0,1]$.

Similarly, to prove that $\operatorname{cl} X$ is convex, assume that $x, y \in \operatorname{cl} X$, so that $x = \lim_{i\to\infty} x_i$ and $y = \lim\limits_{i\to\infty} y_i$ for appropriately chosen $x_i, y_i \in X$. Then for $\lambda \in [0,1]$ we have

$$\lambda x + (1-\lambda) y = \lim_{i\to\infty} \underbrace{[\lambda x_i + (1-\lambda) y_i]}_{\in X},$$

so that $\lambda x + (1-\lambda) y \in \operatorname{cl} X$ for all $\lambda \in [0,1]$.

• **Claim (ii)**: Let X be convex and $\operatorname{int} X$ be nonempty. Then $\operatorname{int} X$ is dense in $\operatorname{cl} X$; moreover,

$$\begin{aligned} &x \in \operatorname{int} X,\ y \in \operatorname{cl} X \Rightarrow \\ &\qquad \lambda x + (1-\lambda) y \in \operatorname{int} X \ \forall \lambda \in (0,1] \end{aligned} \tag{!}$$

Proof. It suffices to prove (!). Indeed, let $\bar{x} \in \operatorname{int} X$ (the latter set is nonempty). Every point $x \in \operatorname{cl} X$ is the limit of the sequence $x_i = \frac{1}{i}\bar{x} + \left(1 - \frac{1}{i}\right) x$. Given (!), all points x_i belong to $\operatorname{int} X$, thus $\operatorname{int} X$ is dense in $\operatorname{cl} X$.

- **<u>Claim (ii)</u>**: Let X be convex and $\operatorname{int} X$ be nonempty. Then

$$\begin{array}{c} x \in \operatorname{int} X,\ y \in \operatorname{cl} X \Rightarrow \\ \lambda x + (1-\lambda) y \in \operatorname{int} X\ \forall \lambda \in (0,1] \end{array} \qquad (!)$$

Proof of (!): Let $x \in \operatorname{int} X$, $y \in \operatorname{cl} X$, $\lambda \in (0,1]$. Let us prove that $\lambda x + (1-\lambda) y \in \operatorname{int} X$.

Since $x \in \operatorname{int} X$, there exists $r > 0$ such that the ball B of radius r centered at x belongs to X. Since $y \in \operatorname{cl} X$, there exists a sequence $y_i \in X$ such that $y = \lim_{i\to\infty} y_i$. Now let

$$\begin{array}{rcl} B^i & = & \lambda B + (1-\lambda) y_i \\ & = & \{\underbrace{z = [\lambda x + (1-\lambda) y_i]}_{z_i} + \lambda h : \|h\|_2 \le r\} \\ & \equiv & \{z = z_i + \delta : \|\delta\|_2 \le r' = \lambda r\}. \end{array}$$

Since $B \subset X$, $y_i \in X$ and X is convex, the sets B^i (which are balls of radius $r' > 0$ centered at z_i) are contained in X. Since $z_i \to z = \lambda x + (1-\lambda) y$ as $i \to \infty$, all these balls, starting with certain number, contain the ball B' of radius $r'/2$ centered at z. Thus, $B' \subset X$, i.e., $z \in \operatorname{int} X$.

♣ Let X be a convex set. It may happen that $\operatorname{int} X = \emptyset$ (e.g., X is a segment in 3D); in this case, interior definitely does not approximate X and $\operatorname{cl} X$. What to do?

The natural way to overcome this difficulty is to pass to *relative* interior, which is nothing but the interior of X *taken w.r.t. the affine hull* $\mathrm{Aff}(X)$ *of* X rather than to $\mathbb{R}^n$. This affine hull, geometrically, is just certain $\mathbb{R}^m$ with $m \leq n$; replacing, if necessary, $\mathbb{R}^n$ with this $\mathbb{R}^m$, we arrive at the situation where $\operatorname{int} X$ is nonempty.

Implementation of the outlined idea goes through the following

<u>Definition:</u> [relative interior and relative boundary] Let X be a nonempty convex set and M be the affine hull of X. The *relative interior* $\operatorname{rint} X$ is the set of all points $x \in X$ such that a ball *in* M of a positive radius, centered at x, is contained in X:

$$\begin{aligned} \operatorname{rint} X \;=\; \{x : \exists r > 0 : & \\ & \{y \in \mathrm{Aff}(X), \|y - x\|_2 \leq r\} \subset X\}. \end{aligned}$$

The *relative boundary* of X is, by definition, $\operatorname{cl} X \setminus \operatorname{rint} X$.

Note: An affine subspace M is given by a list of linear equations and thus is closed; as such, it contains the closure of every subset $Y \subset M$; this closure is nothing but the closure of Y which we would get when replacing the original "universe" $\mathbb{R}^n$ with the affine subspace M (which, geometrically, is nothing but $\mathbb{R}^m$ with certain $m \leq n$).

The essence of the matter is in the following fact:
Proposition: *Let $X \subset \mathbb{R}^n$ be a nonempty convex set. Then* $\operatorname{rint} X \neq \emptyset$.
♣ Thus, replacing, if necessary, the original "universe" $\mathbb{R}^n$ with a smaller *geometrically similar* universe, we can reduce investigating an *arbitrary* nonempty convex set X to the case where this set has a nonempty interior (which is nothing but the *relative* interior of X). In particular, our results for the "full-dimensional" case imply that
For a nonempty convex set X, both $\operatorname{rint} X$ *and* $\operatorname{cl} X$ *are convex sets such that*

$$\emptyset \neq \operatorname{rint} X \subset X \subset \operatorname{cl} X \subset \operatorname{Aff}(X)$$

and $\operatorname{rint} X$ *is dense in* $\operatorname{cl} X$. *Moreover, whenever* $x \in \operatorname{rint} X$, $y \in \operatorname{cl} X$ *and* $\lambda \in (0, 1]$, *one has*

$$\lambda x + (1 - \lambda) y \in \operatorname{rint} X.$$

$$\emptyset \neq X \text{ is convex ??} \;\Rightarrow\; \text{?? rint } X \neq \emptyset$$

Proof. A. By Linear Algebra, whenever $X \subset \mathbb{R}^n$ is nonempty, one can find in X an affine basis for the affine hull Aff(X) of X:

$\exists x_0, x_1, ..., x_m \in X$:
Every $x \in$ Aff(X) admits a representation

$$x = \sum_{i=0}^{m} \lambda_i x_i, \sum_i \lambda_i = 1$$

and the coefficients in this representation are uniquely defined by x.

B. When $x_i \in X$, $i = 0, 1, ..., m$, form an affine basis in $\mathrm{Aff}(X)$, the system of linear equations

$$\begin{array}{rcl} \sum\limits_{i=0}^{m} \lambda_i x_i & = & x \\ \sum\limits_{i=0}^{m} \lambda_i & = & 1 \end{array}$$

in variables λ has a *unique* solution whenever $x \in \mathrm{Aff}(X)$. Since this solution is unique, it, again by Linear Algebra, depends continuously on $x \in \mathrm{Aff}(X)$. In particular, when $x = \bar{x} = \frac{1}{m+1} \sum_{i=0}^{m} x_i$, the solution is positive; by continuity, it remains positive when $x \in \mathrm{Aff}(X)$ is close enough to $\bar{x}$:

$$\exists r > 0 : x \in \mathrm{Aff}(X), \|x - \bar{x}\|_2 \leq r \Rightarrow$$
$$x = \sum_{i=0}^{m} \lambda_i(x) x_i$$
$$\text{with } \sum_i \lambda_i(x) = 1 \text{ and } \lambda_i(x) > 0$$

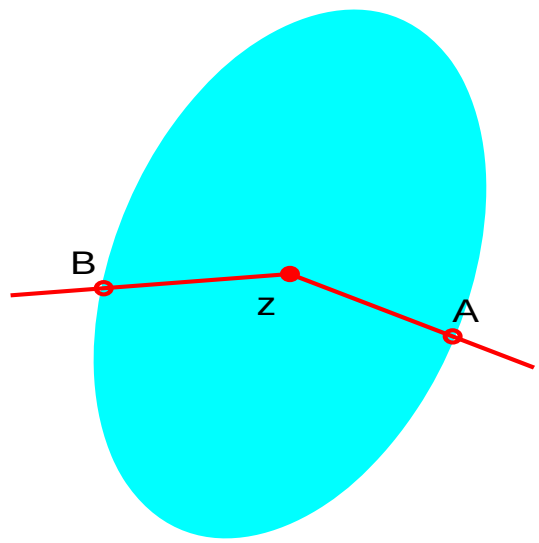
We see that when X is convex, $\bar{x} \in \mathrm{rint}\, X$, Q.E.D.

♣ Let X be convex and $z \in \operatorname{rint} X$. As we know,

$$\lambda \in [0,1],\ y \in \operatorname{cl} X \Rightarrow y_\lambda = \lambda z + (1-\lambda) y \in X.$$

It follows that *in order to pass from X to its closure* $\operatorname{cl} X$, *it suffices to pass to "radial closure"*:

Informally: We look at all rays in $\operatorname{Aff}(x)$ emanating from z and add to X all "missing" – not contained in X from the very beginning – boundary points, like A and B, of the intersections of rays with X.



Formally: For every direction $0 \neq d \in \mathrm{Aff}(X) - z$, let

$$T_d = \{t \geq 0 : z + td \in X\}.$$

<u>Note:</u> T_d is a convex subset of $\mathbb{R}_+$ which contains all small enough positive t's.

$\diamondsuit$ If T_d is unbounded or is a bounded segment: $T_d = \{t : 0 \leq t \leq t(d) < \infty\}$, the intersection of $\mathrm{cl}X$ with the ray $\{z + td : t \geq 0\}$ is exactly the same as the intersection of X with the same ray.

$\diamondsuit$ If T_d is a bounded half-segment: $T_d = \{t : 0 \leq t < t(d) < \infty\}$, the intersection of $\mathrm{cl}X$ with the ray $\{z + td : t \geq 0\}$ is larger than the intersection of X with the same ray by exactly one point, namely, $z + t(d)d$. Adding to X these "missing points" for all d, we arrive at $\mathrm{cl}X$.

Lecture 2:
Convex Sets, II

Main Theorems on Convex Sets, I: Caratheodory Theorem

Definition: Let M be affine subspace in $\mathbb{R}^n$, so that $M = a + L$ for a linear subspace L. The *linear* dimension of L is called the *affine* dimension $\dim M$ of M.

Examples: The affine dimension of a singleton is 0. The affine dimension of $\mathbb{R}^n$ is n. The affine dimension of an affine subspace $M = \{x : Ax = b\}$ is $n - \text{Rank}(A)$.

For a nonempty set $X \subset \mathbb{R}^n$, the *affine dimension* $\dim X$ *of* X is exactly the affine dimension of the affine hull $\text{Aff}(X)$ of X.

Theorem [Caratheodory] *Let* $\emptyset \neq X \subset \mathbb{R}^n$. *Then every point* $x \in \text{Conv}(X)$ *is a convex combination of at most* $\dim(X) + 1$ *points of* X.

Theorem [Caratheodory] *Let* $\emptyset \neq X \subset \mathbb{R}^n$. *Then every point* $x \in \mathrm{Conv}(X)$ *is a convex combination of at most* $\dim(X) + 1$ *points of* X.

Proof. 1^0. We should prove that if x is a convex combination of finitely many points $x_1, ..., x_k$ of X, then x is a convex combination of at most $m + 1$ of these points, where $m = \dim(X)$. Replacing, if necessary, $\mathbb{R}^n$ with $\mathrm{Aff}(X)$, it suffices to consider the case of $m = n$.

2^0. Consider a representation of x as a convex combination of $x_1, ..., x_k$ *with minimum possible number of nonzero coefficients*; it suffices to prove that this number is $\leq n + 1$. Let, on the contrary, the "minimum representation" of x

$$x = \sum_{i=1}^{p} \lambda_i x_i \qquad [\lambda_i \geq 0, \sum_i \lambda_i = 1]$$

has $p > n + 1$ terms.

3^0. Consider the homogeneous system of linear equations in p variables δ_i

$$\begin{cases} (a) & \sum\limits_{i=1}^{p} \delta_i x_i & = & 0 & [n \text{ linear equations}] \\ (b) & \sum\limits_{i} \delta_i & = & 0 & [\text{single linear equation}] \end{cases}$$

Since $p > n+1$, this system has a nontrivial solution δ. Observe that for every $t \geq 0$ one has

$$x = \sum_{i=1}^{p} \underbrace{[\lambda_i + t\delta_i]}_{\lambda_i(t)} x_i \;\&\; \sum_i \lambda_i(t) = 1.$$

$$\delta: \qquad \delta \neq 0 \ \& \ \sum_i \delta_i = 0$$

$$\forall t \geq 0: \ \ x = \sum_{i=1}^{p} \underbrace{[\lambda_i + t\delta_i]}_{\lambda_i(t)} x_i \ \& \ \sum_i \lambda_i(t) = 1.$$

♢ When $t = 0$, all coefficients $\lambda_i(t)$ are nonnegative

♢ When $t \to \infty$, some of the coefficients $\lambda_i(t)$ go to $-\infty$ (indeed, otherwise we would have $\delta_i \geq 0$ for all i, which is impossible since $\sum_i \delta_i = 0$ and not all δ_i are zeros).

♢ It follows that the quantity

$$t_* = \max\{t : t \geq 0 \ \& \ \lambda_i(t) \geq 0 \forall i\}$$

is well defined; when $t = t_*$, all coefficients in the representation

$$x = \sum_{i=1}^{p} \lambda_i(t_*) x_i$$

are nonnegative, sum of them equals to 1, *and at least one of the coefficients* $\lambda_i(t_*)$ *vanishes*. This contradicts the assumption of minimality of the original representation of x as a convex combination of x_i.

Theorem [Caratheodory, Conic Version.] *Let $\emptyset \neq X \subset \mathbb{R}^n$. Then every vector $x \in \mathrm{Cone}\,(X)$ is a conic combination of at most n vectors from X.*

Remark: The bounds given by Caratheodory Theorems (usual and conic version) are sharp:

$\diamondsuit$ for a simplex Δ with $m+1$ vertices $v_0, ..., v_m$ one has $\dim \Delta = m$, and it takes *all* the vertices to represent the barycenter $\frac{1}{m+1} \sum\limits_{i=0}^{m} v_i$ as a convex combination of the vertices;

$\diamondsuit$ The conic hull of n standard basic orths in $\mathbb{R}^n$ is exactly the nonnegative orthant $\mathbb{R}^n_+$, and it takes *all* these vectors to get, as their conic combination, the n-dimensional vector of ones.

Problem: Supermarkets sell 99 different herbal teas; every one of them is certain blend of 26 herbs A,...,Z. In spite of such a variety of marketed blends, John is not satisfied with any one of them; the only herbal tea he likes is their mixture, in the proportion

$$1 : 2 : 3 : \ldots : 98 : 99$$

Once it occurred to John that in order to prepare his favorite tea, there is no necessity to buy all 99 marketed blends; a smaller number of them will do. With some arithmetics, John found a combination of 66 marketed blends which still allows to prepare his tea. Do you believe John's result can be improved?

Answer: *In fact, just 26 properly selected market bends are enough.*
Indeed, let us represent a blend by its unit weight portion, say, 1g. Such a portion can be identified with 26-dimensional vector $x = [x_1; ...; x_{26}]$ with nonnegative entries summing up to 1, where x_i is the weight, in grams, of herb $\#i$ in the portion. Clearly, we have

$$x \in \mathbb{R}^{26}_+ \ \& \ \sum_i x_i = 1.$$

When mixing market blends $x^1, x^2, ..., x^{99}$ to get unit weight portion x of mixture, we take $\lambda_i \geq 0$ grams of market blend x^i, $i = 1, ..., 99$, and mix them together, that is,

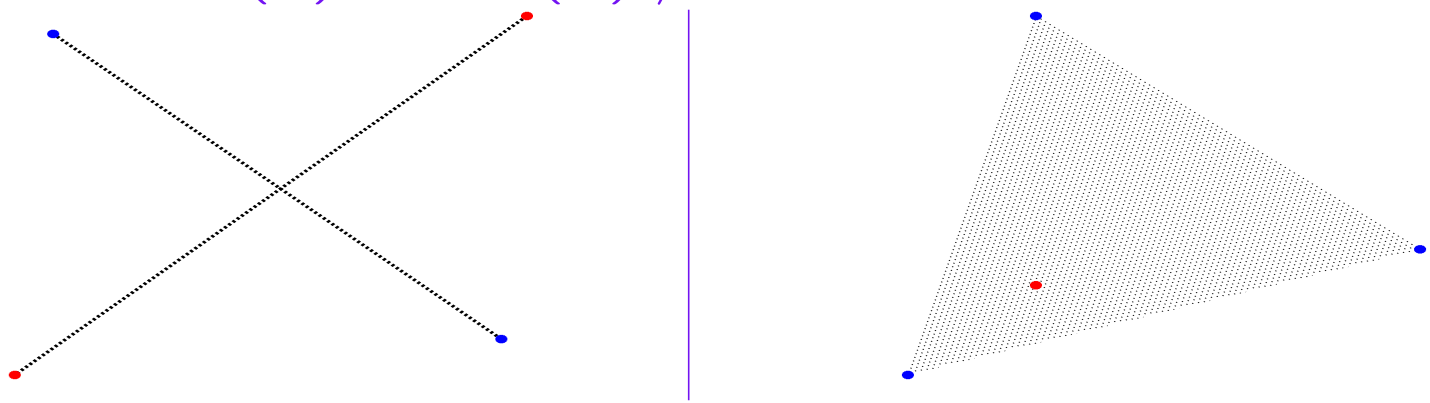
$$x = \sum_i \lambda_i x_i.$$

Looking at the weights of both sides, we get $\sum_i \lambda_i = 1$.
The bottom line: blend x can be obtained by mixing market blends $x^1, ..., x^{99}$ if and only if $x \in \mathrm{Conv}\{x^1, ..., x^{99}\}$.
By Caratheodory Theorem, every blend which can be obtained my mixing market blends can be obtained by mixing $m + 1$ of them, where m is the affine dimension of the affine span of $x^1, ..., x^{99}$. In our case, this span belongs to the 25-dimensional affine plane

$$\{x \in \mathbb{R}^{26} : \sum_i x_i = 1\}$$

that is, $m \leq 25$.

Theorem [Radon] *Let $x_1, ..., x_m$ be $m \geq n+2$ vectors in $\mathbb{R}^n$. One can split these vectors into two nonempty and non-overlapping groups A, B such that* $\mathrm{Conv}(A) \cap \mathrm{Conv}(B) \neq \emptyset$.



Coloring 4 points from $\mathbb{R}^2$ to make convex hulls of red and of blue points intersecting

Proof. Consider the homogeneous system of linear equations in m variables δ_i:

$$\left\{ \begin{array}{rcll} \sum_{i=1}^m \delta_i x_i & = & 0 & [n \text{ linear equations}] \\ \sum_{i=1}^m \delta_i & = & 0 & [\text{single linear equation}] \end{array} \right.$$

Since $m \geq n+2$, the system has a nontrivial solution δ. Setting

$$I = \{i : \delta_i > 0\},\ J = \{i : \delta_i \leq 0\},$$

we split indices $\{1, ..., m\}$ into two *nonempty* (due to $\delta \neq 0, \sum_i \delta_i = 0$) groups such that

$$\sum_{i \in I} \delta_i x_i = \sum_{j \in J} [-\delta_j] x_j,\ \ \gamma = \sum_{i \in I} \delta_i = \sum_{j \in J} -\delta_j > 0$$

whence

$$\underbrace{\sum_{i \in I} \frac{\delta_i}{\gamma} x_i}_{\in \mathrm{Conv}(\{x_i : i \in I\})} = \underbrace{\sum_{j \in J} \frac{-\delta_j}{\gamma} x_j}_{\in \mathrm{Conv}(\{x_j : j \in J\})} .$$

Theorem [Helley] *Let $A_1, ..., A_M$ be convex sets in $\mathbb{R}^n$. Assume that every $n+1$ sets from the family have a point in common. Then all M sets have point in common.*

Proof: induction in M.

Base $M \leq n+1$ is trivially true.

Step: Assume that for certain $M \geq n+1$ our statement hods true for every M-member family of convex sets, and let us prove that it holds true for $M+1$-member family of convex sets $A_1, ..., A_{M+1}$.

$\diamondsuit$ By inductive hypotheses, every one of the $M+1$ sets

$$B_\ell = A_1 \cap A_2 \cap ... \cap A_{\ell-1} \cap A_{\ell+1} \cap ... \cap A_{M+1}$$

is nonempty. Let us choose $x_\ell \in B_\ell$, $\ell = 1, ..., M+1$.

$\diamondsuit$ By Radon's Theorem, the collection $x_1, ..., x_{M+1}$ can be split in two sub-collections with intersecting convex hulls. W.l.o.g., let the split be $\{x_1, ..., x_{J-1}\} \cup \{x_J, ..., x_{M+1}\}$, and let

$$z \in \mathrm{Conv}(\{x_1, ..., x_{J-1}\}) \bigcap \mathrm{Conv}(\{x_J, ..., x_{M+1}\}).$$

<u>Situation:</u> x_j belongs to all sets A_ℓ except, perhaps, for A_j and

$$z \in \mathrm{Conv}(\{x_1, ..., x_{J-1}\}) \bigcap \mathrm{Conv}(\{x_J, ..., x_{M+1}\}).$$

<u>Claim:</u> $z \in A_\ell$ for all $\ell \leq M + 1$.
Indeed, for $\ell \leq J - 1$, the points $x_J, x_{J+1}, ..., x_{M+1}$ belong to the convex set A_ℓ, whence

$$z \in \mathrm{Conv}(\{x_J, ..., x_{M+1}\}) \subset A_\ell.$$

For $\ell \geq J$, the points $x_1, ..., x_{J-1}$ belong to the convex set A_ℓ, whence

$$z \in \mathrm{Conv}(\{x_1, ..., x_{J-1}\}) \subset A_\ell.$$

<u>Refinement:</u> *Assume that $A_1, ..., A_M$ are convex sets in $\mathbb{R}^n$ and that*
$\diamondsuit$ *the union $A_1 \cup A_2 \cup ... \cup A_M$ of the sets belongs to an affine subspace P of affine dimension m*
$\diamondsuit$ *every $m+1$ sets from the family have a point in common*
Then all the sets have a point in common.
Proof. We can think of A_j as of sets in P, or, which is the same, as sets in $\mathbb{R}^m$ and apply the Helley Theorem!

What about infinite collections $\{A_\alpha\}_{\alpha\in\mathcal{A}}$?

• When trying to extend Helley's Theorem from finite to infinite collections of convex sets, we meet two immediate obstacles:

• *Things can go wrong when the sets A_α are not closed.* E.g. for the collection $\{A_i = (0, 1/i)\}_{i\geq 1}$ of convex subsets of $\mathbb{R}$, intersection of sets from every finite subcollection is nonempty, but the intersection of all A_i is empty

• *Things can go wrong when the intersections of sets from finite subcollections can "run to infinity,"* as is the case for collection $\{A_i = [i, \infty)\}_{i\geq 1}$ of convex subsets of $\mathbb{R}$. Here again intersection of sets from every finite subcollection is nonempty, but the intersection of all A_i is empty.

♠ It turns out that these are the only two obstacles for Helley Theorem to be applicable to infinite collections of convex sets.

Helley Theorem II: *Let A_α, $\alpha \in \mathcal{A}$, be a family of convex sets in $\mathbb{R}^n$ such that every $n+1$ sets from the family have a point in common.*
Assume, in addition, that
$\diamondsuit$ *the sets A_α are closed*
$\diamondsuit$ *one can find finitely many sets $A_{\alpha_1}, ..., A_{\alpha_M}$ with a bounded intersection.*
Then all sets A_α, $\alpha \in \mathcal{A}$, have a point in common.

Proof. By the Helley Theorem, every finite collection of the sets A_α has a point in common, and it remains to apply the following standard fact from Analysis:
Let B_α be a family of closed sets in $\mathbb{R}^n$ such that
$\diamondsuit$ every finite collection of the sets has a nonempty intersection;
$\diamondsuit$ in the family, there exists finite collection with bounded intersection.
Then all sets from the family have a point in common.

Proof of the Standard Fact is based upon the following fundamental property of $\mathbb{R}^n$:

Every closed and bounded subset of $\mathbb{R}^n$ is a compact set.

Recall two equivalent definitions of a compact set:

- *A subset X in a metric space M is called compact, if from every sequence of points of X one can extract a sub-sequence converging to a point from X*
- *A subset X in a metric space M is called compact, if from every open covering of X* (i.e., from every family of open sets such that every point of X belongs to at least one of them) *one can extract a finite sub-covering.*

Now let B_α be a family of closed sets in $\mathbb{R}^n$ such that every finite sub-family of the sets has a nonempty intersection and at least one of these intersection, let it be B, is bounded.
Let us prove that all sets B_α have a point in common.
• Assume that it is not the case. Then for every point $x \in B$ there exists a set B_α which does not contain x. Since B_α is closed, it does not intersect an appropriate open ball V_x centered at x. Note that the system $\{V_x : x \in B\}$ forms an open covering of B.
• By its origin, B is closed (as intersection of closed sets) and bounded and *thus is a compact set.* Therefore one can find a *finite* collection $V_{x_1}, ..., V_{x_M}$ which covers B. For every $i \leq M$, there exists a set B_{α_i} in the family which does not intersect V_{x_i}; therefore $\bigcap\limits_{i=1}^{M} B_{\alpha_i}$ does not intersect B. Since B itself is the intersection of finitely many sets B_α, we see that the intersection of finitely many sets B_α (those participating in the description of B and the sets B_{α_1},...,B_{α_M}) is empty, which is a contradiction.

Exercise: We are given a function $f(x)$ on a 7,000,000-point set $X \subset \mathbb{R}$. At every 7-point subset of X, this function can be approximated, within accuracy 0.001 at every point, by appropriate polynomial of degree 5. To approximate the function on the entire X, we want to use a spline of degree 5 (a piecewise polynomial function with pieces of degree 5). How many pieces do we need to get accuracy 0.001 at every point?
Answer: Just one. Indeed, let A_x, $x \in X$, be the set of coefficients of all polynomials of degree 5 which reproduce $f(x)$ within accuracy 0.001:

$$A_x = \{p = (p_0, ..., p_5) \in \mathbb{R}^6 : \\ |f(x) - \sum_{i=0}^{5} p_i x^i| \leq 0.001\}.$$

The set A_x is polyhedral and therefore convex, and we know that every $6 + 1 = 7$ sets from the family $\{A_x\}_{x \in X}$ have a point in common. By Helley Theorem, all sets A_x, $x \in X$, have a point in common, that is, there exists a *single* polynomial of degree 5 which approximates f within accuracy 0.001 at *every* point of X.

Exercise: We should design a factory which, mathematically, is described by the following Linear Programming model:

$$\begin{array}{rcll} Ax & \geq & d & [d_1, ..., d_{1000}\text{: demands}] \\ Bx & \leq & f & [f_1 \geq 0, ..., f_{10} \geq 0\text{: amounts of resources of various types}] \\ Cx & \leq & c & [\text{other constraints}] \end{array} \qquad (F)$$

The data A, B, C, c are given in advance. We should buy in advance resources $f_i \geq 0$, $i = 1, ..., 10$, in such a way that the factory will be capable to satisfy all demand scenarios d from a given finite set D, that is, (F) should be feasible for every $d \in D$. Amount f_i of resource i costs us $a_i f_i$.

It is known that in order to be able to satisfy every single demand from D, it suffices to invest \$1 in the resources.

How large should be investment in resources in the cases when D contains

- ♢ just one scenario?
- ♢ 3 scenarios?
- ♢ 10 scenarios?
- ♢ 2004 scenarios?

Answer: $D = \{d_1\} \Rightarrow$ \$1 is enough
$D = \{d_1, d_2, d_3\} \Rightarrow$ \$3 is enough
$D = \{d_1, ..., d_{10}\} \Rightarrow$ \$10 is enough
$D = \{d_1, ..., d_{2004}\} \Rightarrow$ \$11 is enough!
Indeed, for $d \in D$ let F_d be the set of all nonnegative $f \in \mathbb{R}^{10}$, $f \geq 0$ which cost at most \$11 and result in solvable system

$$\begin{array}{rcl} Ax & \geq & d \\ Bx & \leq & f \\ Cx & \leq & c \end{array} \qquad (F[d])$$

in variables x. The set F_d is convex (why?), and *every 11 sets of this type have a common point*. Indeed, given 11 scenarios $d^1, ..., d^{11}$ from D, we can meet demand scenario d^i investing \$1 in properly selected vector of resources $f^i \geq 0$; therefore we can meet every one of 11 scenarios $d^1, ..., d^{11}$ by a *single* vector of resources $f^1 + ... + f^{11}$ at the cost of \$11, and therefore this vector belongs to every one of the sets $F_{d^1} ..., F_{d^{11}}$.
Since every 11 of 2004 convex sets $F_d \subset \mathbb{R}^{10}$, $d \in D$, have a point in common, all these sets have a point f in common; for this f, every one of the systems $(F[d])$, $d \in D$, is solvable.

Exercise: Consider an optimization program

$$c_* = \left\{c^T x : g_i(x) \leq 0,\, i = 1, ..., 2004\right\}$$

with 11 variables $x_1, ..., x_{11}$. Assume that the constraints are convex, that is, every one of the sets

$$X_i = \{x : g_i(x) \leq 0\},\, i = 1, ..., 2004$$

is convex. Assume also that the problem is solvable with optimal value 0.
Clearly, when dropping one or more constraints, the optimal value can only decrease or remain the same.
$\diamondsuit$ Is it possible to find a constraint such that dropping it, we preserve the optimal value? Two constraints which can be dropped simultaneously with no effect on the optimal value? Three of them?

<u>Answer:</u> You can drop as many as $2004 - 11 = 1993$ appropriately chosen constraints without varying the optimal value!
Assume, on the contrary, that every 11-constraint relaxation of the original problem has negative optimal value. Since there are finitely many such relaxations, there exists $\epsilon < 0$ such that every problem of the form

$$\min_x \{c^T x : g_{i_1}(x) \leq 0, ..., g_{i_{11}}(x) \leq 0\}$$

has a feasible solution with the value of the objective $< -\epsilon$. Since this problem has a feasible solution with the value of the objective equal to 0 (namely, the optimal solution of the original problem) and its feasible set is convex, the problem has a feasible solution x with $c^T x = -\epsilon$. In other words, every 11 of the 2004 sets

$$Y_i = \{x : c^T x = -\epsilon, g_i(x) \leq 0\},\ i = 1, ..., 2004$$

have a point in common.

Every 11 of the 2004 sets

$$Y_i = \{x : c^T x = -\epsilon, g_i(x) \leq 0\},\ i = 1, ..., 2004$$

have a point in common!

The sets Y_i are convex (as intersections of convex sets X_i and an affine subspace). If $c \neq 0$, then these sets belong to affine subspace of affine dimension 10, and since every 11 of them intersect, all 2004 intersect; a point x from their intersection is a feasible solution of the original problem with $c^T x < 0$, which is impossible.

When $c = 0$, the claim is evident: we can drop all 2004 constraints without varying the optimal value!

Lecture 3:
Polyhedral Sets

Theory of Systems of Linear Inequalities, 0
Polyhedrality & Fourier-Motzkin Elimination

♣ **Definition:** A polyhedral set $X \subset \mathbb{R}^n$ is a set which can be represented as

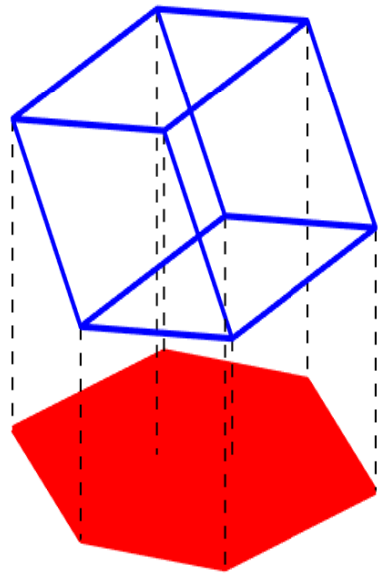
$$X = \{x : Ax \leq b\},$$

that is, as the solution set of a finite system of nonstrict linear inequalities.

♣ **Definition:** A polyhedral representation of a set $X \subset \mathbb{R}^n$ is a representation of X of the form:

$$X = \{x : \exists w : Px + Qw \leq r\},$$

that is, a representation of X as the a projection onto the space of x-variables of a polyhedral set $X^+ = \{[x; w] : Px + Qw \leq r\}$ in the space of x, w-variables.



Rotated 3D cube and its 2D projection (hexagon)

♠ Examples of polyhedral representations:

- The set $X = \{x \in \mathbb{R}^n : \sum_i |x_i| \leq 1\}$ admits the p.r.

$$X = \left\{ x \in \mathbb{R}^n : \exists w \in \mathbb{R}^n : \begin{array}{r} -w_i \leq x_i \leq w_i, \\ 1 \leq i \leq n, \\ \sum_i w_i \leq 1 \end{array} \right\}.$$

- The set

$$X = \begin{array}{l} \{x \in \mathbb{R}^6 : \max[x_1, x_2, x_3] + 2\max[x_4, x_5, x_6] \\ \quad \leq x_1 - x_6 + 5\} \end{array}$$

admits the p.r.

$$X = \left\{ x \in \mathbb{R}^6 : \exists w \in \mathbb{R}^2 : \begin{array}{l} x_1 \leq w_1, x_2 \leq w_1, x_3 \leq w_1 \\ x_4 \leq w_2, x_5 \leq w_2, x_6 \leq w_2 \\ w_1 + 2w_2 \leq x_1 - x_6 + 5 \end{array} \right\}.$$

Whether a Polyhedrally Represented Set is Polyhedral?

♣ **Question:** Let X be given by a polyhedral representation:

$$X = \{x \in \mathbb{R}^n : \exists w : Px + Qw \leq r\},$$

that is, as the *projection* of the solution set

$$Y = \{[x; w] : Px + Qw \leq r\} \qquad (*)$$

of a finite system of linear inequalities in variables x, w onto the space of x-variables.
Is it true that X is polyhedral, i.e., X is a solution set of finite system of linear inequalities *in variables x only?*

Theorem. *Every polyhedrally representable set is polyhedral.*

Proof is given by the *Fourier — Motzkin elimination scheme* which demonstrates that the projection of the set $(*)$ onto the space of x-variables is a polyhedral set.

$$Y = \{[x; w] : Px + Qw \leq r\}, \qquad (*)$$

Elimination step: eliminating a *single* slack variable. Given set $(*)$, assume that $w = [w_1; ...; w_m]$ is nonempty, and let Y^+ be the projection of Y on the space of variables $x, w_1, ..., w_{m-1}$:

$$Y^+ = \{[x; w_1; ...; w_{m-1}] : \exists w_m : Px + Qw \leq r\} \qquad (!)$$

Let us prove that Y^+ is polyhedral. Indeed, let us split the linear inequalities

$$p_i^T x + q_i^T w \leq r_i, \; 1 \leq i \leq I$$

defining Y into three groups:

- **black** – the coefficient at w_m is 0
- **red** – the coefficient at w_m is > 0
- **green** – the coefficient at w_m is < 0

Then

$$\begin{aligned} Y = \big\{[x; w] : & \\ & a_i^T x + b_i^T [w_1; ...; w_{m-1}] \leq c_i, \;\; i \text{ is black} \\ & w_m \leq a_i^T x + b_i^T [w_1; ...; w_{m-1}] + c_i, \;\; i \text{ is red} \\ & w_m \geq a_i^T x + b_i^T [w_1; ...; w_{m-1}] + c_i, \;\; i \text{ is green} \big\} \end{aligned}$$

$$
\begin{aligned}
Y = \big\{ [x; w] : \\
a_i^T x + b_i^T [w_1; ...; w_{m-1}] \le c_i, \ \ i \text{ is black} \\
w_m \le a_i^T x + b_i^T [w_1; ...; w_{m-1}] + c_i, \ \ i \text{ is red} \\
w_m \ge a_i^T x + b_i^T [w_1; ...; w_{m-1}] + c_i, \ \ i \text{ is green} \big\}
\end{aligned}
$$

$\Rightarrow$

$$
\begin{aligned}
Y^+ = \big\{ [x; w_1; ...; w_{m-1}] : \\
a_i^T x + b_i^T [w_1; ...; w_{m-1}] \le c_i, \ \ i \text{ is black} \\
a_\mu^T x + b_\mu^T [w_1; ...; w_{m-1}] + c_\mu \ge a_\nu^T x + b_\nu^T [w_1; ...; w_{m-1}] + c_\nu \\
\text{whenever } \mu \text{ is red and } \nu \text{ is green} \big\}
\end{aligned}
$$

and thus Y^+ is polyhedral.

We have seen that the projection

$$Y^+ = \{[x; w_1; ...; w_{m-1}] : \exists w_m : [x; w_1; ...; w_m] \in Y\}$$

of the polyhedral set $Y = \{[x, w] : Px + Qw \leq r\}$ is polyhedral. Iterating the process, we conclude that the set $X = \{x : \exists w : [x, w] \in Y\}$ is polyhedral, Q.E.D.

♣ Given an LO program

$$\mathrm{Opt} = \max_x \left\{ c^T x : Ax \leq b \right\}, \tag{!}$$

observe that the set of values of the objective at feasible solutions can be represented as

$$\begin{aligned} T &= \{\tau \in \mathbb{R} : \exists x : Ax \leq b, c^T x - \tau = 0\} \\ &= \{\tau \in \mathbb{R} : \exists x : Ax \leq b, c^T x \leq \tau, c^T x \geq \tau\} \end{aligned}$$

that is, *T is polyhedrally representable*. By Theorem, T is polyhedral, that is, T can be represented by a finite system of linear inequalities *in variable τ only.* It immediately follows that *if T is nonempty and is bounded from above, T has the largest element*. Thus, we have proved

Corollary. *A feasible and bounded LO program admits an optimal solution and thus is solvable.*

$$\begin{aligned} T &= \{\tau \in \mathbb{R} : \exists x : Ax \leq b, c^T x - \tau = 0\} \\ &= \{\tau \in \mathbb{R} : \exists x : Ax \leq b, c^T x \leq \tau, c^T x \geq \tau\} \end{aligned}$$

♣ Fourier-Motzkin Elimination Scheme suggests a finite algorithm for solving an LO program, where we

• first, apply the scheme to get a representation of T by a finite system S of linear inequalities in variable τ,

• second, analyze S to find out whether the solution set is nonempty and bounded from above, and when it is the case, to find out the optimal value $\mathrm{Opt} \in T$ of the program,

• third, use the Fourier-Motzkin elimination scheme in the backward fashion to find x such that $Ax \leq b$ and $c^T x = \mathrm{Opt}$, thus recovering an optimal solution to the problem of interest.

Bad news: The resulting algorithm is completely impractical, since the number of inequalities we should handle at a step usually rapidly grows with the step number and can become astronomically large when eliminating just tens of variables.

Theory of Systems of Linear Inequalities, I
Homogeneous Farkas Lemma

♣ Consider a homogeneous linear inequality

$$a^T x \geq 0 \qquad (*)$$

along with a finite system of similar inequalities:

$$a_i^T x \geq 0,\ 1 \leq i \leq m \qquad (!)$$

♣ **Question:** *When* $(*)$ *is a consequence of* $(!)$*, that is, every* x *satisfying* $(!)$ *satisfies* $(*)$ *as well?*

Observation: If a is a conic combination of $a_1, ..., a_m$:

$$\exists \lambda_i \geq 0 : a = \sum_i \lambda_i a_i, \qquad (+)$$

then $(*)$ is a consequence of $(!)$.

Indeed, $(+)$ implies that

$$a^T x = \sum_i \lambda_i a_i^T x \ \forall x,$$

and thus for every x with $a_i^T x \geq 0 \, \forall i$ one has $a^T x \geq 0$.

$$a^T x \geq 0 \tag{$*$}$$

$$a_i^T x \geq 0,\ 1 \leq i \leq m \tag{!}$$

♣ **Homogeneous Farkas Lemma:** $(*)$ *is a consequence of* (!) *if and only if* a *is a conic combination of* $a_1, ..., a_m$.

♣ **Equivalently:** *Given vectors* $a_1, ..., a_m \in \mathbb{R}^n$, *let* $K = \mathrm{Cone}\{a_1, ..., a_m\} = \{\sum_i \lambda_i a_i : \lambda \geq 0\}$ *be the conic hull of the vectors. Given a vector* a,

- *it is easy to certify that* $a \in \mathrm{Cone}\{a_1, ..., a_m\}$: *a certificate is a collection of weights* $\lambda_i \geq 0$ *such that* $\sum_i \lambda_i a_i = a$;
- *it is easy to certify that* $a \notin \mathrm{Cone}\{a_1, ..., a_m\}$: *a certificate is a vector* d *such that* $a_i^T d \geq 0\, \forall i$ *and* $a^T d < 0$.

Proof of HFL: All we need to prove is that *If a is not a conic combination of $a_1, ..., a_m$, then there exists d such that $a^T d < 0$ and $a_i^T d \geq 0$, $i = 1, ..., m$.*
Fact: The set $K = \text{Cone}\{a_1, ..., a_m\}$ is polyhedrally representable:

$$\text{Cone}\{a_1, ..., a_m\} = \left\{ x : \exists \lambda \in \mathbb{R}^m : \begin{array}{c} x = \sum_i \lambda_i a_i \\ \lambda \geq 0 \end{array} \right\}.$$

$\Rightarrow$ By Fourier-Motzkin, K is polyhedral:

$$K = \{x : d_\ell^T x \geq c_\ell, 1 \leq \ell \leq L\}.$$

Observation I: $0 \in K \Rightarrow c_\ell \leq 0 \, \forall \ell$
Observation II: $\lambda a_i \in \text{Cone}\{a_1, ..., a_m\} \, \forall \lambda > 0 \Rightarrow \lambda d_\ell^T a_i \geq c_\ell \, \forall \lambda \geq 0 \Rightarrow d_\ell^T a_i \geq 0 \, \forall i, \ell$.
Now, $a \notin \text{Cone}\{a_1, ..., a_m\} \Rightarrow \exists \ell = \ell_* : d_{\ell_*}^T a < c_{\ell_*} \leq 0 \Rightarrow d_{\ell_*}^T a < 0$.
$\Rightarrow d = d_{\ell_*}$ *satisfies* $a^T d < 0$, $a_i^T d \geq 0$, $i = 1, ..., m$, *Q.E.D.*

Theory of Systems of Linear Inequalities, II
Theorem on Alternative

♣ A general (finite!) system of linear inequalities with unknowns $x \in \mathbb{R}^n$ can be written down as

$$\begin{array}{rcl} a_i^T x & > & b_i,\ i = 1, ..., m_{\mathsf{s}} \\ a_i^T x & \geq & b_i,\ i = m_{\mathsf{s}} + 1, ..., m \end{array} \qquad (S)$$

Question: *How to certify that (S) is solvable?*
Answer: A solution is a certificate of solvability!
Question: *How to certify that S is* **not** *solvable?*
Answer: ???

$$\begin{array}{rcl} a_i^T x & > & b_i,\ i = 1, ..., m_{\mathsf{s}} \\ a_i^T x & \geq & b_i,\ i = m_{\mathsf{s}} + 1, ..., m \end{array} \qquad (S)$$

Question: *How to certify that S is* **not** *solvable?*

Conceptual **sufficient insolvability condition:**

If we can lead the assumption that x solves (S) to a contradiction, then (S) has no solutions.

Example: To certify that the system

$$\begin{array}{rrrcr} -4u & -9v & +5w & > & 2 \\ -2u & +6v & & \geq & -2 \\ 7u & & -5w & \geq & 1 \end{array}$$

has no solutions, it suffices to point out that *aggregating the inequalities of the system with weights $2, 3, 2$, we get a contradictory inequality:*

$$\begin{array}{c|rrrcr} & 2\times & -4u & -9v & +5w & > & 2 \\ + & & & & & & \\ & 3\times & -2u & +6v & & \geq & -2 \\ + & & & & & & \\ & 2\times & 7u & & -5w & \geq & 1 \\ \hline & & 0 \cdot u & +0 \cdot v & +0 \cdot w & > & 0 \end{array}$$

By how we aggregate, every solution to the system *must* solve the aggregated inequality. The latter has no solutions $\Rightarrow$ so is the system.

$$\begin{array}{rcl} a_i^T x & > & b_i,\ i = 1, ..., m_{\mathsf{s}} \\ a_i^T x & \geq & b_i,\ i = m_{\mathsf{s}} + 1, ..., m \end{array} \qquad (S)$$

"Contradiction by linear aggregation": Let us associate with inequalities of (S) *non-negative* weights λ_i and sum up the inequalities with these weights. The resulting inequality

$$\left[\sum_{i=1}^{m} \lambda_i a_i\right]^T x \begin{cases} > \sum\limits_i \lambda_i b_i, & \sum\limits_{i=1}^{m_{\mathsf{s}}} \lambda_i > 0 \\ \geq \sum\limits_i \lambda_i b_i, & \sum\limits_{i=1}^{m_{\mathsf{s}}} \lambda_i = 0 \end{cases} \qquad (C)$$

by its origin is a *consequence* of (S), that is, it is satisfied at every solution to (S). Consequently, *if there exist $\lambda \geq 0$ such that (C) has no solutions at all, then (S) has no solutions!*

<u>Question:</u> When a linear inequality

$$d^T x \begin{cases} > \\ \geq \end{cases} e$$

has no solutions at all?
<u>Answer:</u> This is the case if and only if $d = 0$ and

— either the sign is "$>$", and $e \geq 0$,

— or the sign is "$\geq$", and $e > 0$.

Conclusion: *Consider a system of linear inequalities*

$$\begin{array}{rcl} a_i^T x & > & b_i,\ i = 1, ..., m_{\mathsf{s}} \\ a_i^T x & \geq & b_i,\ i = m_{\mathsf{s}} + 1, ..., m \end{array} \qquad (S)$$

in variables x, and let us associate with it two systems of linear inequalities in variables λ:

$$\mathcal{T}_{\mathrm{I}} : \left\{ \begin{array}{rcl} \lambda & \geq & 0 \\ \sum\limits_{i=1}^{m} \lambda_i a_i & = & 0 \\ \sum\limits_{i=1}^{m_s} \lambda_i & > & 0 \\ \sum\limits_{i=1}^{m} \lambda_i b_i & \geq & 0 \end{array} \right. \qquad \mathcal{T}_{\mathrm{II}} : \left\{ \begin{array}{rcl} \lambda & \geq & 0 \\ \sum\limits_{i=1}^{m} \lambda_i a_i & = & 0 \\ \sum\limits_{i=1}^{m_s} \lambda_i & = & 0 \\ \sum\limits_{i=1}^{m} \lambda_i b_i & > & 0 \end{array} \right.$$

If *one of the systems $\mathcal{T}_{\mathrm{I}}$, $\mathcal{T}_{\mathrm{II}}$ is solvable,* **then** *(S) is unsolvable.*
Note: **If** *$\mathcal{T}_{\mathrm{II}}$ is solvable,* **then** *already the system*

$$a_i^T x \geq b_i,\ i = m_{\mathsf{s}} + 1, ..., m$$

is unsolvable!

General Theorem on Alternative: *A system of linear inequalities*

$$\begin{array}{rcl} a_i^T x & > & b_i,\ i = 1, ..., m_{\mathsf{s}} \\ a_i^T x & \geq & b_i,\ i = m_{\mathsf{s}} + 1, ..., m \end{array} \qquad (S)$$

is **un***solvable* **iff** *one of the systems*

$$\mathcal{T}_{\mathrm{I}} : \left\{ \begin{array}{rcl} \lambda & \geq & 0 \\ \sum_{i=1}^{m} \lambda_i a_i & = & 0 \\ \sum_{i=1}^{m_s} \lambda_i & > & 0 \\ \sum_{i=1}^{m} \lambda_i b_i & \geq & 0 \end{array} \right. \qquad \mathcal{T}_{\mathrm{II}} : \left\{ \begin{array}{rcl} \lambda & \geq & 0 \\ \sum_{i=1}^{m} \lambda_i a_i & = & 0 \\ \sum_{i=1}^{m_s} \lambda_i & = & 0 \\ \sum_{i=1}^{m} \lambda_i b_i & > & 0 \end{array} \right.$$

is solvable.

Note: *The subsystem*

$$a_i^T x \geq b_i,\ i = m_{\mathsf{s}} + 1, ..., m$$

of (S) *is* **un***solvable* **iff** $\mathcal{T}_{\mathrm{II}}$ *is solvable!*

Proof. We already know that solvability of one of the systems $\mathcal{T}_{\rm I}$, $\mathcal{T}_{\rm II}$ is a sufficient condition for **<u>un</u>**solvability of (S). All we need to prove is that if (S) is unsolvable, then one of the systems $\mathcal{T}_{\rm I}$, $\mathcal{T}_{\rm II}$ is solvable.
Assume that the system

$$\begin{array}{rcl} a_i^T x & > & b_i,\ i = 1, ..., m_{\rm s} \\ a_i^T x & \geq & b_i,\ i = m_{\rm s} + 1, ..., m \end{array} \qquad (S)$$

in variables x has no solutions. Then *every solution x, τ, ϵ to the homogeneous system of inequalities*

$$\begin{array}{rrrcl} & \tau & -\epsilon & \geq & 0 \\ a_i^T x & -b_i\tau & -\epsilon & \geq & 0,\ i = 1, ..., m_{\rm s} \\ a_i^T x & -b_i\tau & & \geq & 0,\ i = m_{\rm s} + 1, ..., m \end{array}$$

has $\epsilon \leq 0$.
Indeed, in a solution with $\epsilon > 0$ one would also have $\tau > 0$, and the vector $\tau^{-1}x$ would solve (S).

Situation: Every solution to the system of homogeneous inequalities

$$\begin{array}{rrrcll} \tau & -\epsilon & \geq & 0 & & [\text{weight } \nu \geq 0] \\ a_i^T x & -b_i\tau & -\epsilon & \geq & 0,\ i = 1, ..., m_{\mathsf{s}} & [\text{weight } \lambda_i \geq 0] \\ a_i^T x & -b_i\tau & & \geq & 0,\ i = m_{\mathsf{s}}+1, ..., m & [\text{weight } \lambda_i \geq 0] \end{array} \qquad (U)$$

has $\epsilon \leq 0$, i.e., the homogeneous inequality

$$-\epsilon \geq 0 \qquad (I)$$

is a consequence of system (U) of homogeneous inequalities. By Homogeneous Farkas Lemma, *the vector of coefficients in the left hand side of (I) is a conic combination of the left hand side vectors of coefficients of (U):*

$$\begin{array}{rcll} \exists \lambda \geq 0, \nu \geq 0: & & & \\ \sum\limits_{i=1}^{m} \lambda_i a_i & = & 0 & [\text{coefficients at } x] \\ -\sum\limits_{i=1}^{m} \lambda_i b_i + \nu & = & 0 & [\text{coefficient at } \tau] \\ -\sum\limits_{i=1}^{m_{\mathsf{s}}} \lambda_i - \nu & = & -1 & [\text{coefficient at } \epsilon] \end{array}$$

Assuming that $\lambda_1 = ... = \lambda_{m_{\mathsf{s}}} = 0$, we get $\nu = 1$, and therefore λ solves $\mathcal{T}_{\mathrm{II}}$. In the case of $\sum\limits_{i=1}^{m_{\mathsf{s}}} \lambda_i > 0$, λ clearly solves $\mathcal{T}_{\mathrm{I}}$.

Corollaries of GTA

♣ **Principle A:** *A finite system of linear inequalities has no solutions iff one can lead it to a contradiction by linear aggregation, i.e., an appropriate weighted sum of the inequalities with "legitimate" weights is either a contradictory inequality*

$$0^T x > a \qquad [a \geq 0]$$

or a contradictory inequality

$$0^T x \geq a \qquad [a > 0]$$

♣ **Principle B:** [Inhomogeneous Farkas Lemma] *A linear inequality*

$$a^T x \leq b$$

is a consequence of **solvable** *system of linear inequalities*

$$a_i^T x \leq b_i,\ i = 1, ..., m$$

iff the target inequality can be obtained from the inequalities of the system **and** *the identically true inequality*

$$0^T x \leq 1$$

by linear aggregation, that is, iff there exist nonnegative $\lambda_0, \lambda_1, ..., \lambda_m$ *such that*

$$\begin{array}{rcl} a & = & \sum\limits_{i=1}^m \lambda_i a_i \\ b & = & \lambda_0 + \sum\limits_{i=1}^m \lambda_i b_i \end{array} \qquad \left[\Leftrightarrow \left\{ \begin{array}{rcl} a & = & \sum\limits_{i=1}^m \lambda_i a_i \\ b & \geq & \sum\limits_{i=1}^m \lambda_i b_i \end{array} \right. \right]$$

Linear Programming Duality Theorem

♣ The origin of the LP dual of a Linear Programming program

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : Ax \geq b \right\} \qquad (P)$$

is the desire to get a systematic way to bound from below the optimal value in (P). The conceptually simplest bounding scheme is *linear aggregation of the constraints*:

Observation: *For every vector λ of nonnegative weights, the constraint*

$$[A^T\lambda]^T x \equiv \lambda^T A x \geq \lambda^T b$$

is a consequence of the constraints of (P) and as such is satisfied at every feasible solution of (P).

Corollary: *For every vector $\lambda \geq 0$ such that $A^T\lambda = c$, the quantity $\lambda^T b$ is a lower bound on $\mathrm{Opt}(P)$.*

♣ *The problem dual to (P) is nothing but the problem*

$$\mathrm{Opt}(D) = \max_\lambda \left\{ b^T\lambda : \lambda \geq 0, A^T\lambda = c \right\} \qquad (D)$$

of maximizing the lower bound on $\mathrm{Opt}(P)$ given by Corollary.

♣ The origin of (D) implies the following

Weak Duality Theorem: *The value of the primal objective at every feasible solution of the primal problem*

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : Ax \geq b \right\} \qquad (P)$$

is $\geq$ the value of the dual objective at every feasible solution to the dual problem

$$\mathrm{Opt}(D) = \max_\lambda \left\{ b^T \lambda : \lambda \geq 0, A^T \lambda = c \right\} \qquad (D)$$

that is,

$$\left.\begin{array}{l} x \text{ is feasible for } (P) \\ \lambda \text{ is feasible for } (D) \end{array}\right\} \Rightarrow c^T x \geq b^T \lambda$$

In particular,

$$\mathrm{Opt}(P) \geq \mathrm{Opt}(D).$$

♣ **<u>LP Duality Theorem:</u>** *Consider an LP program along with its dual:*

$$\begin{array}{rcll} \mathrm{Opt}(P) & = & \min\limits_{x} \left\{c^T x : Ax \geq b\right\} & (P) \\ \mathrm{Opt}(D) & = & \max\limits_{\lambda} \left\{b^T \lambda : A^T \lambda = c, \lambda \geq 0\right\} & (D) \end{array}$$

Then

♢ *Duality is symmetric: the problem dual to dual is (equivalent to) the primal*

♢ *The value of the dual objective at every dual feasible solution is* $\leq$ *the value of the primal objective at every primal feasible solution*

♢ *The following 5 properties are equivalent to each other:*

- *(i)* (P) *is feasible and bounded (below)*
- *(ii)* (D) *is feasible and bounded (above)*
- *(iii)* (P) *is solvable*
- *(iv)* (D) *is solvable*
- *(v)* *both* (P) *and* (D) *are feasible*

and whenever they take place, one has $\mathrm{Opt}(P) = \mathrm{Opt}(D)$.

$$\begin{array}{rcll} \text{Opt}(P) & = & \min\limits_{x} \left\{ c^T x : Ax \geq b \right\} & (P) \\ \text{Opt}(D) & = & \max\limits_{\lambda} \left\{ b^T \lambda : A^T \lambda = c, \lambda \geq 0 \right\} & (D) \end{array}$$

♢ *Duality is symmetric*

Proof: Rewriting (D) in the form of (P), we arrive at the problem

$$\min_{\lambda} \left\{ -b^T \lambda : \begin{bmatrix} A^T \\ -A^T \\ I \end{bmatrix} \lambda \geq \begin{bmatrix} c \\ -c \\ 0 \end{bmatrix} \right\},$$

with the dual being

$$\max_{u,v,w} \left\{ c^T u - c^T v + 0^T w : \begin{array}{c} u \geq 0, v \geq 0, w \geq 0, \\ Au - Av + w = -b \end{array} \right\}$$

$$\Updownarrow$$

$$\max_{x = v-u, w} \left\{ -c^T x : w \geq 0, Ax = b + w \right\}$$

$$\Updownarrow$$

$$\min_{x} \left\{ c^T x : Ax \geq b \right\}$$

$\diamondsuit$ *The value of the dual objective at every dual feasible solution is $\leq$ the value of the primal objective at every primal feasible solution*

This is Weak Duality

♢ *The following 5 properties are equivalent to each other:*

(P) is feasible and bounded below (i)

$$\Downarrow$$

(D) is solvable (iv)

Indeed, by origin of $\mathrm{Opt}(P)$, the inequality

$$c^T x \geq \mathrm{Opt}(P)$$

is a consequence of the (solvable!) system of inequalities

$$Ax \geq b.$$

By Principle B, the inequality is a linear consequence of the system:

$$\exists \lambda \geq 0 : A^T \lambda = c \ \& \ b^T \lambda \geq \mathrm{Opt}(P).$$

Thus, the dual problem has a feasible solution with the value of the dual objective $\geq \mathrm{Opt}(P)$. By Weak Duality, this solution is dual optimal, and $\mathrm{Opt}(D) = \mathrm{Opt}(P)$.

♢ *The following properties are equivalent to each other:*

$$\begin{array}{rr} (D) \text{ is solvable} & \text{(iv)} \\ \Downarrow & \\ (D) \text{ is feasible and bounded above} & \text{(ii)} \end{array}$$

Evident

♢ *The following 5 properties are equivalent to each other:*

(D) is feasible and bounded above (ii)

$$\Downarrow$$

(P) is solvable (iii)

Implied by already proved relation

(P) is feasible and bounded below (i)

$$\Downarrow$$

(D) is solvable (iv)

in view of primal-dual symmetry

♢ *The following 5 properties are equivalent to each other:*

$$(P) \text{ is solvable} \qquad \text{(iii)}$$
$$\Downarrow$$
$$(P) \text{ is feasible and bounded below} \qquad \text{(i)}$$

Evident

We proved that

$$\text{(i)} \Leftrightarrow \text{(ii)} \Leftrightarrow \text{(iii)} \Leftrightarrow \text{(iv)}$$

and that when these 4 equivalent properties take place, one has

$$\mathrm{Opt}(P) = \mathrm{Opt}(D)$$

It remains to prove that properties (i) – (iv) are equivalent to

$$\text{both } (P) \text{ and } (D) \text{ are feasible} \qquad \text{(v)}$$

◇ In the case of (v), (P) is feasible and below bounded (Weak Duality), so that (v)⇒(i)
◇ in the case of (i)≡(ii), both (P) and (D) are feasible, so that (i)⇒(v)

Optimality Conditions in LP

Theorem: *Consider a primal-dual pair of feasible LP programs*

$$\begin{aligned} \text{Opt}(P) &= \min_x \left\{c^T x : Ax \geq b\right\} && (P) \\ \text{Opt}(D) &= \max_\lambda \left\{b^T \lambda : A^T \lambda = c, \lambda \geq 0\right\} && (D) \end{aligned}$$

and let x, λ be **feasible** *solutions to the respective programs. These solutions are optimal for the respective problems*

$\diamondsuit$ *iff* $c^T x - b^T \lambda = 0$ ["zero duality gap"]

as well as

$\diamondsuit$ *iff* $[Ax - b]_i \cdot \lambda_i = 0$ *for all* i ["complementary slackness"]

Proof: Under Theorem's premise, $\text{Opt}(P) = \text{Opt}(D)$, so that

$$c^T x - b^T \lambda = \underbrace{c^T x - \text{Opt}(P)}_{\geq 0} + \underbrace{\text{Opt}(D) - b^T \lambda}_{\geq 0}$$

Thus, duality gap $c^T x - b^T \lambda$ is always nonnegative and is zero iff x, λ are optimal for the respective problems.

The complementary slackness condition is given by the identity

$$c^T x - b^T \lambda = (A^T \lambda)^T x - b^T \lambda = [Ax - b]^T \lambda$$

Since both $[Ax - b]$ and λ are nonnegative, duality gap is zero iff the complementary slackness

$$[Ax - b]_i \lambda_i = 0 \ \forall i$$

holds true.

Lecture 4:
Separation and Extreme Points

Separation Theorem

♣ Every linear form $f(x)$ on $\mathbb{R}^n$ is representable via inner product:

$$f(x) = f^T x$$

for appropriate vector $f \in \mathbb{R}^n$ uniquely defined by the form. Nontrivial (not identically zero) forms correspond to nonzero vectors f.

♣ A *level set*

$$M = \{x : f^T x = a\} \qquad (*)$$

of a *nontrivial* linear form on $\mathbb{R}^n$ is affine subspace of affine dimension $n-1$; vice versa, every affine subspace M of affine dimension $n-1$ in $\mathbb{R}^n$ can be represented by $(*)$ with appropriately chosen $f \neq 0$ and a; f and a are defined by M up to multiplication by a common nonzero factor.

$(n-1)$-dimensional affine subspaces in $\mathbb{R}^n$ are called *hyperplanes*.

$$M = \{x : f^T x = a\} \qquad (*)$$

♣ Level set $(*)$ of nontrivial linear form splits $\mathbb{R}^n$ into two parts:

$$\begin{array}{rcl} M_+ & = & \{x : f^T x \geq a\} \\ M_- & = & \{x : f^T x \leq a\} \end{array}$$

called *closed half-spaces* given by (f, a); the hyperplane M is the common boundary of these half-spaces. The interiors M_{++} of M_+ and M_{--} of M_- are given by

$$\begin{array}{rcl} M_{++} & = & \{x : f^T x > a\} \\ M_{--} & = & \{x : f^T x < a\} \end{array}$$

and are called *open half-spaces* given by (f, a). We have

$$\mathbb{R}^n = M_- \bigcup M_+ \quad [M_- \bigcap M_+ = M]$$

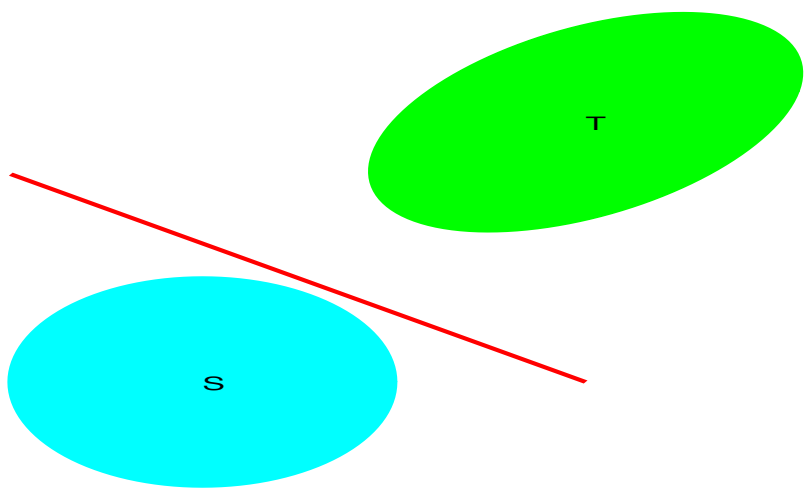
and

$$\mathbb{R}^n = M_{--} \bigcup M \bigcup M_{++}$$

♣ **<u>Definition.</u>** Let T, S be two nonempty sets in $\mathbb{R}^n$.
(i) We say that a hyperplane

$$M = \{x : f^T x = a\} \qquad (*)$$

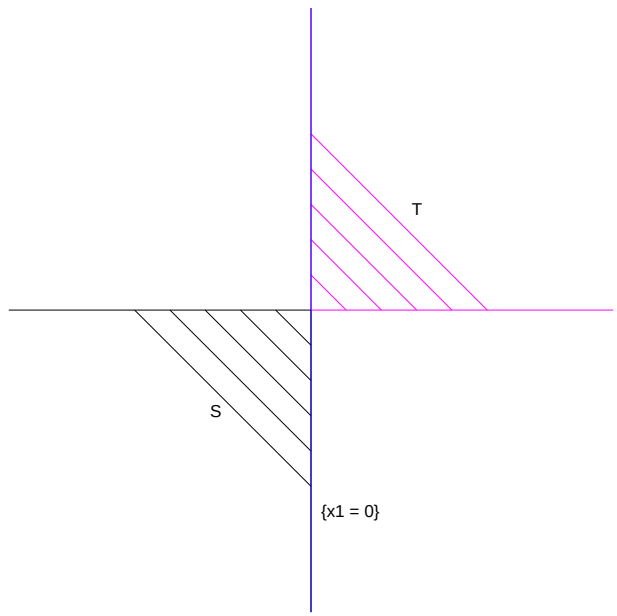
separates S and T, if
♢ $S \subset M_-$, $T \subset M_+$ ("S does not go above M, and T does not go below M")
and
♢ $S \cup T \not\subset M$.
(ii) We say that a nontrivial linear form $f^T x$ separates S and T if, for properly chosen a, the hyperplane $(*)$ separates S and T.



Red hyperplane $2x_1 + 3x_2 = 6$ separates cyan set S and green set T

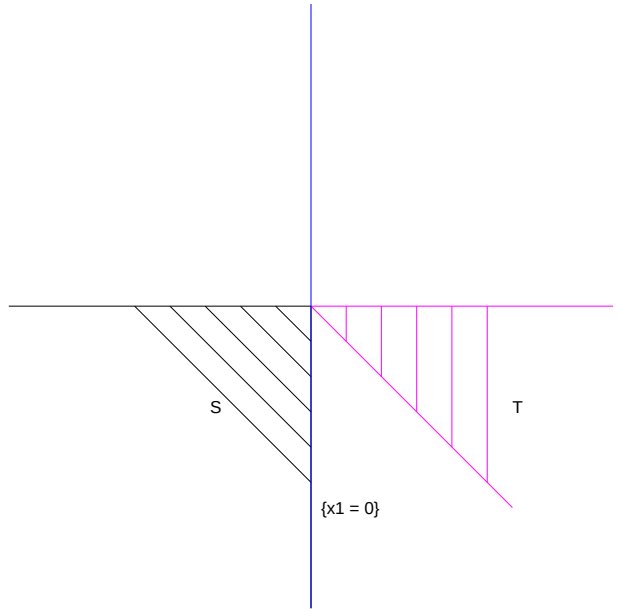
Examples: The linear form x_1 on $\mathbb{R}^2$
1) separates the sets

$$\begin{aligned} S &= \{x \in \mathbb{R}^2 : x_1 \leq 0, x_2 \leq 0\}, \\ T &= \{x \in \mathbb{R}^2 : x_1 \geq 0, x_2 \geq 0\} : \end{aligned}$$



The linear form x_1 on $\mathbb{R}^2$...
2) separates the sets

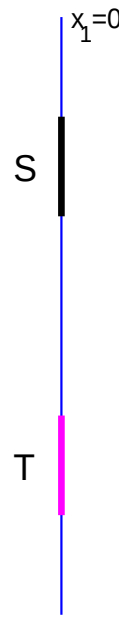
$$\begin{aligned} S &= \{x \in \mathbb{R}^2 : x_1 \leq 0, x_2 \leq 0\}, \\ T &= \{x \in \mathbb{R}^2 : x_1 + x_2 \geq 0, x_2 \leq 0\} : \end{aligned}$$



The linear form x_1 on $\mathbb{R}^2$...

3) does **<u>not</u>** separate the sets

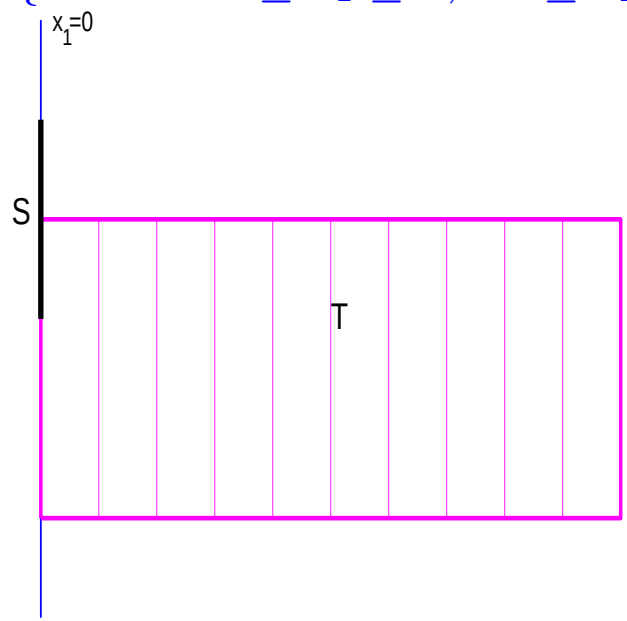
$$\begin{aligned} S &= \{x \in \mathbb{R}^2 : x_1 = 0, 1 \leq x_2 \leq 2\}, \\ T &= \{x \in \mathbb{R}^2 : x_1 = 0, -2 \leq x_2 \leq -1\} : \end{aligned}$$



The linear form x_1 on $\mathbb{R}^2$...

4) separates the sets

$$\begin{aligned} S &= \{x \in \mathbb{R}^2 : x_1 = 0, 0 \leq x_2 \leq 2\}, \\ T &= \{x \in \mathbb{R}^2 : 0 \leq x_1 \leq 1, -2 \leq x_2 \leq 1\} : \end{aligned}$$



<u>Observation:</u> *A linear form $f^T x$ separates nonempty sets S, T iff*

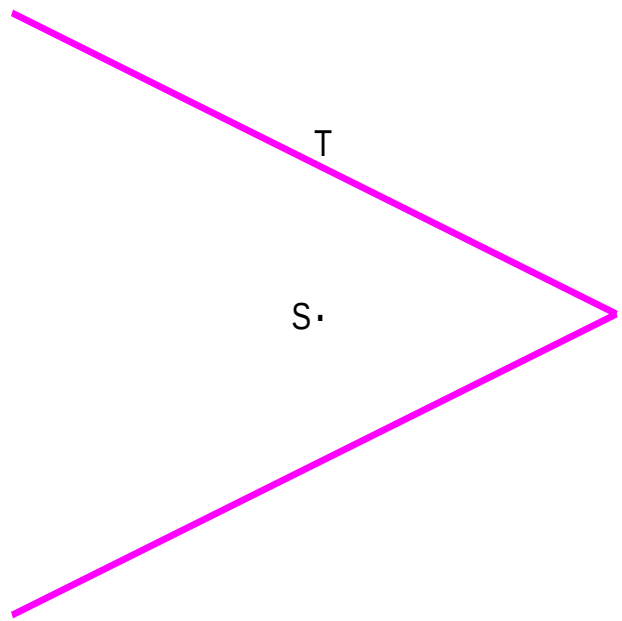
$$\begin{array}{rcl} \sup\limits_{x\in S} f^T x & \leq & \inf\limits_{y\in T} f^T y \\ \inf\limits_{x\in S} f^T x & < & \sup\limits_{y\in T} f^T y \end{array} \qquad (*)$$

In the case of $(*)$, the associated with f hyperplanes separating S and T are exactly the hyperplanes

$$\{x : f^T x = a\} \text{ with } \sup_{x\in S} f^T x \leq a \leq \inf_{y\in T} f^T y.$$

♣ **Separation Theorem:** *Two nonempty* **convex** *sets S, T can be separated iff their relative interiors do not intersect.*

Note: In this statement, convexity of both S and T is crucial!



Proof, $\Rightarrow$: (!) **<u>If</u>** *nonempty convex sets S, T can be separated,* **<u>then</u>** $\operatorname{rint} S \bigcap \operatorname{rint} T = \emptyset$

<u>Lemma.</u> *Let X be a convex set, $f(x) = f^T x$ be a linear form and $a \in \operatorname{rint} X$. Then*

$$f^T a = \max_{x \in X} f^T x \Leftrightarrow f(\cdot)\Big|_X = \text{const}.$$

♣ **<u>Lemma $\Rightarrow$ (!):</u>** Let $a \in \operatorname{rint} S \cap \operatorname{rint} T$. Assume, on contrary to what should be proved, that $f^T x$ separates S, T, so that

$$\sup_{x \in S} f^T x \leq \inf_{y \in T} f^T y.$$

♢ Since $a \in T$, we get $f^T a \geq \sup_{x \in S} f^T x$, that is, $f^T a = \max_{x \in S} f^T x$. By Lemma, $f^T x = f^T a$ for all $x \in S$.

♢ Since $a \in S$, we get $f^T a \leq \inf_{y \in T} f^T y$, that is, $f^T a = \min_{y \in T} f^T y$. By Lemma, $f^T y = f^T a$ for all $y \in T$.

Thus,

$$z \in S \cup T \Rightarrow f^T z \equiv f^T a,$$

so that f does **<u>not</u>** separate S and T, which is a contradiction.

Lemma. *Let X be a convex set, $f(x) = f^T x$ be a linear form and $a \in \operatorname{rint} X$. Then*

$$f^T a = \max_{x \in X} f^T x \Leftrightarrow f(\cdot)\Big|_X = \text{const}.$$

Proof. Shifting X, we may assume $a = 0$. Let, on the contrary to what should be proved, $f^T x$ be non-constant on X, so that there exists $y \in X$ with $f^T y \neq f^T a = 0$. The case of $f^T y > 0$ is impossible, since $f^T a = 0$ is the maximum of $f^T x$ on X. Thus, $f^T y < 0$. The line $\{ty : t \in \mathbb{R}\}$ passing through 0 and through y belongs to $\operatorname{Aff}(X)$; since $0 \in \operatorname{rint} X$, all points $z = -\epsilon y$ on this line belong to X, provided that $\epsilon > 0$ is small enough. At every point of this type, $f^T z > 0$, which contradicts the fact that $\max_{x \in X} f^T x = f^T a = 0$.

Proof, $\Leftarrow$: Assume that S, T are nonempty convex sets such that $\operatorname{rint} S \cap \operatorname{rint} T = \emptyset$, and let us prove that S, T can be separated.

Step 1: Separating a point and a convex hull of a finite set. Let $S = \mathrm{Conv}(\{b_1, ..., b_m\})$ and $T = \{b\}$ with $b \notin S$, and let us prove that S and T can be separated.

Indeed,

$$S = \mathrm{Conv}(b_1, ..., b_m) = \left\{ x : \exists \lambda : \begin{array}{l} \lambda \geq 0, \sum_i \lambda_i = 1 \\ x = \sum_i \lambda_i b_i \end{array} \right\}$$

is polyhedrally representable and thus is polyhedral:

$$S = \{x : a_\ell^T x \leq \alpha_\ell, \ell \leq L\}.$$

Since $b \notin S$, for some $\bar{\ell}$ we have

$$a_{\bar{\ell}}^T b > \alpha_{\bar{\ell}} \geq \sup_{x \in S} a_{\bar{\ell}}^T x$$

which is the desired separation.

Step 2: Separating a point and a convex set which does not contain the point. Let S be a nonempty convex set and $T = \{b\}$ with $b \notin S$, and let us prove that S and T can be separated.

1^0. Shifting S and T by $-b$ (which clearly does not affect the possibility of separating the sets), we can assume that $T = \{0\} \not\subset S$.

2^0. Replacing, if necessary, $\mathbb{R}^n$ with $\mathrm{Lin}(S)$, we may further assume that $\mathbb{R}^n = \mathrm{Lin}(S)$.

Lemma: *Every nonempty subset S in $\mathbb{R}^n$ is separable: one can find a sequence $\{x_i\}$ of points from S which is dense in S, i.e., is such that every point $x \in S$ is the limit of an appropriate subsequence of the sequence.*

Lemma $\Rightarrow$ Separation: Let $\{x_i \in S\}$ be a sequence which is dense in S. Since S is convex and does not contain 0, we have

$$0 \notin \mathrm{Conv}(\{x_1, ..., x_i\}) \ \forall i$$

whence

$$\exists f_i : 0 = f_i^T 0 > \max_{1 \le j \le i} f_i^T x_j. \qquad (*)$$

By scaling, we may assume that $\|f_i\|_2 = 1$.
The sequence $\{f_i\}$ of unit vectors possesses a converging subsequence $\{f_{i_s}\}_{s=1}^{\infty}$; the limit f of this subsequence is, of course, a unit vector. By $(*)$, for every fixed j and all large enough s we have $f_{i_s}^T x_j < 0$, whence

$$f^T x_j \le 0 \ \forall j. \qquad (**)$$

Since $\{x_j\}$ is dense in S, $(**)$ implies that $f^T x \le 0$ for all $x \in S$, whence

$$\sup_{x \in S} f^T x \le 0 = f^T 0.$$

<u>Situation:</u> (a) $\mathrm{Lin}(S) = \mathbb{R}^n$
(b) $T = \{0\}$
(c) We have built a unit vector f such that

$$\sup_{x \in S} f^T x \leq 0 = f^T 0. \tag{!}$$

By (!), all we need to prove that f separates $T = \{0\}$ and S is to verify that

$$\inf_{x \in S} f^T x < f^T 0 = 0.$$

Assuming the opposite, (!) would say that $f^T x = 0$ for all $x \in S$, which is impossible, since $\mathrm{Lin}(S) = \mathbb{R}^n$ and f is nonzero.

Lemma: *Every nonempty subset S in $\mathbb{R}^n$ is separable: one can find a sequence $\{x_i\}$ of points from S which is dense in S, i.e., is such that every point $x \in S$ is the limit of an appropriate subsequence of the sequence.*

Definition: A set X is called *countable*, if one can arrange all its elements into a (finite or infinite) sequence:

$$X = \{x_1, x_2, x_3, ...\}$$

First preliminary fact: *The set $\mathbf{Q}^n$ of vectors $r \in \mathbb{R}^n$ with rational coordinates is countable: one can arrange all these vectors in a sequence $r_1, r_2, ...$* Indeed, representing rational numbers as fractions with numerator and denominator without common factors, for every integer $N \geq 0$ there are finitely many rational n-dimensional vectors with the total sum of magnitudes of numerators and denominators in the coordinates not exceeding N, let the set of these vectors be R_N. We have

$$R_0 \subset R_1 \subset R_2 \subset ..$$

Now let us arrange all rational vectors from $\mathbb{R}^n$ into a single sequence as follows:

- first, we write down, in a whatever order, all vectors from the (finite!) set R_0
- next, we add to the resulting finite sequence all vectors from the (finite) set $R_1 \backslash R_0$, again in a whatever order
- next, we add to the finite sequence we have built so far all vectors from the (finite!) set $R_2 \backslash R_1$, and so on.

As a result, all rational n-dimensional vectors will be arranged into a sequence $r_1, r_2, ...$

Second preliminary fact: *The union* $X = \bigcup_{i=1,2,\ldots} X_i$ *of countably many countable sets* $X_i = \{x_{ij}\}_{j=1,2,\ldots}$ *is countable.*

Indeed, $X = \{x_{ij} : i, j = 1, 2, ..\}$, and we can arrange all the elements x_{ij} into a single sequence writing down,

- first, all x_{ij} with $i + j \leq 1$ (in a whatever order),
- next, all x_{ij} with $1 < i + j \leq 2$ (in a whatever order),
- next, all x_{ij} with $2 < i + j \leq 3$, and so on.

If the element we are about to write down was already written down (it was met, with different pair of indexes, before), we skip writing it down.

As a result, all x_{ij}'s will be arranged into a (finite or infinite) sequence, complying countability of X.

Lemma: *Every nonempty subset S in $\mathbb{R}^n$ is separable: one can find a sequence $\{x_i\}$ of points from S which is dense in S, i.e., is such that every point $x \in S$ is the limit of an appropriate subsequence of the sequence.*

Proof. Let $r_1, r_2, ...$ be the sequence comprised of all rational vectors in $\mathbb{R}^n$. For every positive integer t, let $X_t \subset S$ be the countable set given by the following construction:

> We look, one after another, at the points $r_1, r_2, ...$ and for every point r_s check whether there is a point z in S which is at most at the distance $1/t$ away from r_s. If points z with this property exist, we take one of them and add it to X_t and then pass to r_{s+1}, otherwise directly pass to r_{s+1}.

Is is clear that

> (*) Every point $x \in S$ is at the distance at most $2/t$ from certain point of X_t.

Indeed, since the rational vectors are dense in $\mathbb{R}^n$, there exists s such that r_s is at the distance $\leq \frac{1}{t}$ from x. Therefore, when processing r_s, we definitely add to X_t a point z which is at the distance $\leq 1/t$ from r_s and thus is at the distance $\leq 2/t$ from x.

- The countable union $\bigcup\limits_{t=1}^{\infty} X_t$ of countable sets $X_t \subset S$ is a countable set in S, and by (*) this set is dense in S.

Step 3: Separating two non-intersecting nonempty convex sets. Let S, T be nonempty convex sets which do not intersect; let us prove that S, T can be separated. Let $\widehat{S} = S - T$ and $\widehat{T} = \{0\}$. The set $\widehat{S}$ clearly is convex and does not contain 0 (since $S \cap T = \emptyset$). By Step 2, $\widehat{S}$ and $\{0\} = \widehat{T}$ can be separated: there exists f such that

$$\left\{\begin{array}{rcl} \overbrace{\sup\limits_{x\in S,y\in T}[f^Tx - f^Ty]}^{\sup\limits_{x\in S} f^Ts - \inf\limits_{y\in T} f^Ty} & \leq & 0 = \inf\limits_{z\in\{0\}} f^Tz \\ \underbrace{\inf\limits_{x\in S,y\in T}[f^Tx - f^Ty]}_{\inf\limits_{x\in S} f^Tx - \sup\limits_{y\in T} f^Ty} & < & 0 = \sup\limits_{z\in\{0\}} f^Tz \end{array}\right.$$

whence

$$\begin{array}{rcl} \sup\limits_{x\in S} f^Tx & \leq & \inf\limits_{y\in T} f^Ty \\ \inf\limits_{x\in S} f^Tx & < & \sup\limits_{y\in T} f^Ty \end{array}$$

Step 4: Completing the proof of Separation Theorem. Finally, let S, T be nonempty convex sets with non-intersecting relative interiors, and let us prove that S, T can be separated.

As we know, the sets $S' = \operatorname{rint} S$ and $T' = \operatorname{rint} T$ are convex and nonempty; we are in the situation when these sets do not intersect. By Step 3, S' and T' can be separated: for properly chosen f, one has

$$\begin{array}{rcl} \sup\limits_{x \in S'} f^T x & \leq & \inf\limits_{y \in T'} f^T y \\ \inf\limits_{x \in S'} f^T x & < & \sup\limits_{y \in T'} f^T y \end{array} \qquad (*)$$

Since S' is dense in S and T' is dense in T, inf's and sup's in $(*)$ remain the same when replacing S' with S and T' with T. Thus, f separates S and T.

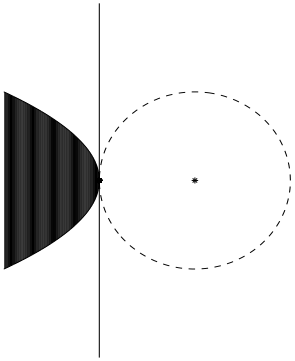
♣ Alternative proof of Separation Theorem starts with separating a point $T = \{a\}$ and a *closed* convex set S, $a \notin S$, and is based on the following fact:

Let S be a nonempty closed convex set and let $a \notin S$. There exists a unique closest to a point in S:

$$\mathrm{Proj}_S(a) = \operatorname*{argmin}_{x \in S} \|a - x\|_2$$

and the vector $e = a - \mathrm{Proj}_S(a)$ separates a and S:

$$\max_{x \in S} e^T x = e^T \mathrm{Proj}_S(a) = e^T a - \|e\|_2^2 < e^T a.$$



Proof: 1^0. The closest to a point in S does exist. Indeed, let $x_i \in S$ be a sequence such that

$$\|a - x_i\|_2 \to \inf_{x \in S} \|a - x\|_2, \ , \ i \to \infty$$

The sequence $\{x_i\}$ clearly is bounded; passing to a subsequence, we may assume that $x_i \to \bar{x}$ as $i \to \infty$. *Since S is closed*, we have $\bar{x} \in S$, and

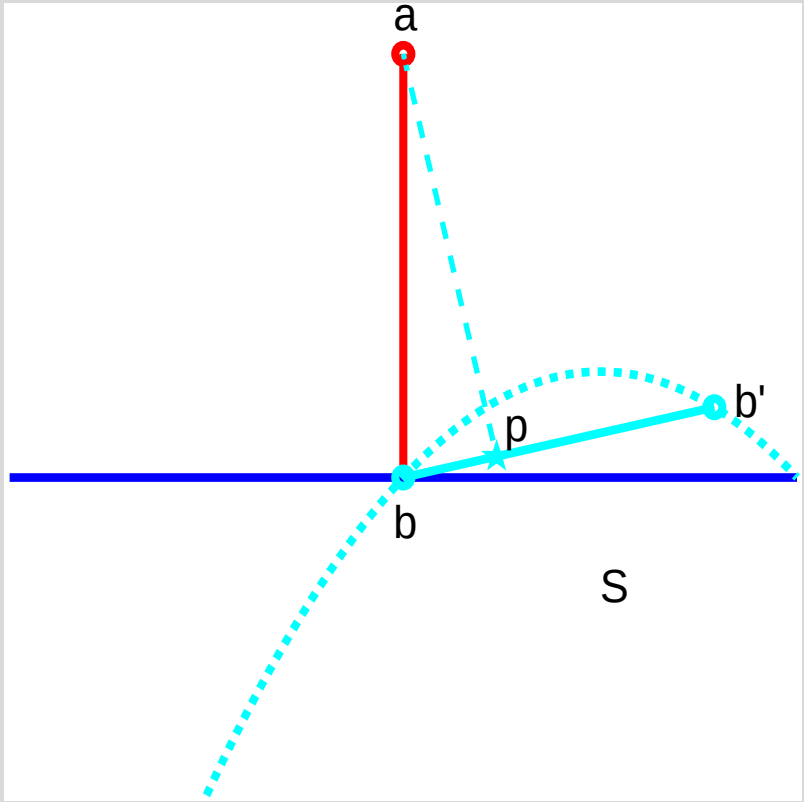
$$\|a - \bar{x}\|_2 = \lim_{i \to \infty} \|a - x_i\|_2 = \inf_{x \in S} \|a - x\|_2.$$

2^0. The closest to a point in S is unique. Indeed, let x, y be two closest to a points in S, so that $\|a - x\|_2 = \|a - y\|_2 = d$. *Since S is convex*, the point $z = \frac{1}{2}(x + y)$ belongs to S; therefore $\|a - z\|_2 \geq d$. We now have

$$\begin{aligned} &\overbrace{\|[a - x] + [a - y]\|_2^2}^{=\|2(a-z)\|_2^2 \geq 4d^2} + \overbrace{\|[a - x] - [a - y]\|_2^2}^{=\|x-y\|^2} \\ &= \underbrace{2\|a - x\|_2^2 + 2\|a - y\|_2^2}_{4d^2} \end{aligned}$$

whence $\|x - y\|_2 = 0$.

3^0. Thus, the closest to a point $b = \mathrm{Proj}_S(a)$ in S exists, is unique and differs from a (since $a \notin S$). The hyperplane passing through b and orthogonal to $a - b$ separates a and S:



Indeed, if there were a point $b' \in S$ "above" the hyperplane, the entire segment $[b, b']$ would be contained in S by convexity of S. Since the angle $\angle abb'$ is $< \pi/2$, performing a small step from b towards b' we stay in S and become closer to a, which is impossible!

With $e = a - \text{Proj}_S(a)$, we have

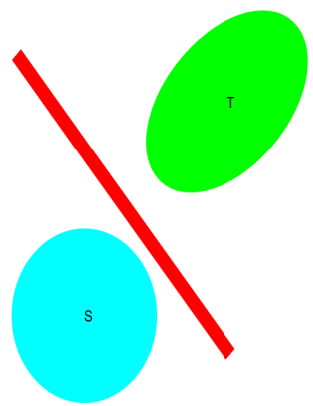
$$x \in S, f = x - \text{Proj}_S(a)$$
$$\Downarrow$$
$$\begin{aligned} \phi(t) &\equiv \|e - tf\|_2^2 \\ &= \|a - [\text{Proj}_S(a) + t(x - \text{Proj}_S(a))]\|_2^2 \\ &\geq \|a - \text{Proj}_S(a)\|_2^2 \\ &= \phi(0), 0 \leq t \leq 1 \end{aligned}$$
$$\Downarrow$$
$$0 \leq \phi'(0) = -2e^T(x - \text{Proj}_S(a))$$
$$\Downarrow$$
$$\forall x \in S : e^T x \leq e^T \text{Proj}_S(a) = e^T a - \|e\|_2^2.$$

♣ Separation of sets S, T by linear form $f^T x$ is called *strict*, if

$$\sup_{x \in S} f^T x < \inf_{y \in T} f^T y$$

Geometrically: For properly selected $\delta > 0$ and a, S and T are separated by the stripe $\{x : a - \delta \leq f^T x \leq a + \delta\}$:

$$\sup_{x \in S} f^T x \leq a - \delta < a + \delta \leq \inf_{y \in T} f^T y$$



Theorem: *Let S, T be nonempty convex sets. These sets can be strictly separated iff they are at positive distance:*

$$\mathrm{dist}(S, T) = \inf_{x \in S, y \in T} \|x - y\|_2 > 0.$$

♣ Separation of sets S, T by linear form $f^T x$ is called *strict*, if

$$\sup_{x \in S} f^T x < \inf_{y \in T} f^T y$$

Claim: *Two nonempty convex sets S, T can be strictly separated iff they are at positive distance:*

$$\mathrm{dist}(S, T) = \inf_{x \in S, y \in T} \|x - y\|_2 > 0.$$

Proof, $\Rightarrow$: Let f strictly separate S, T; let us prove that S, T are at positive distance. Otherwise we could find sequences $x_i \in S$, $y_i \in T$ with $\|x_i - y_i\|_2 \to 0$ as $i \to \infty$, whence $f^T(y_i - x_i) \to 0$ as $i \to \infty$. It follows that the sets on the axis

$$\widehat{S} = \{a = f^T x : x \in S\}, \widehat{T} = \{b = f^T y : y \in T\}$$

are at zero distance, which is a contradiction with

$$\sup_{a \in \widehat{S}} a < \inf_{b \in \widehat{T}} b.$$

Proof, $\Leftarrow$: Let T, S be nonempty convex sets which are at positive distance 2δ:

$$2\delta = \inf_{x \in S, y \in T} \|x - y\|_2 > 0.$$

Let

$$S^+ = S + \{z : \|z\|_2 \leq \delta\}$$

The sets S^+ and T are convex and do not intersect, and thus can be separated:

$$\sup_{x_+ \in S^+} f^T x_+ \leq \inf_{y \in T} f^T y \qquad [f \neq 0]$$

Since

$$\begin{aligned} \sup_{x_+ \in S^+} f^T x_+ &= \sup_{x \in S, \|z\|_2 \leq \delta} [f^T x + f^T z] \\ &= [\sup_{x \in S} f^T x] + \delta \|f\|_2, \end{aligned}$$

we arrive at

$$\sup_{x \in S} f^T x < \inf_{y \in T} f^T y$$

Exercise Below S is a nonempty convex set and $T = \{a\}$.

Statement	True?
If T and S can be separated then $a \notin S$	
If $a \notin S$, then T and S can be separated	
If T and S can be strictly separated, then $a \notin S$	
If $a \notin S$, then T and S can be strictly separated	
If S is closed and $a \notin S$, then T and S can be strictly separated	

Supporting Planes and Extreme Points

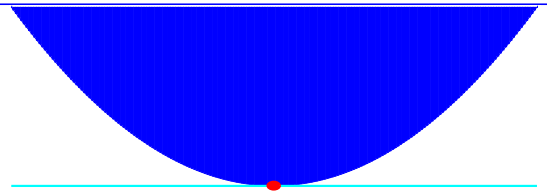
♣ **Definition.** Let Q be a *closed* convex set in $\mathbb{R}^n$ and $\bar{x}$ be a point from the relative boundary of Q. A hyperplane

$$\Pi = \{x : f^T x = a\} \qquad [f \neq 0]$$

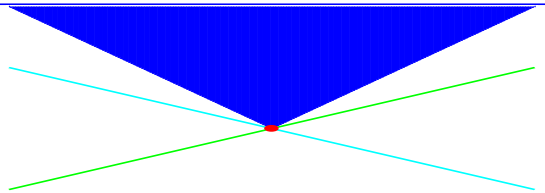
is called *supporting to Q at the point $\bar{x}$*, if the hyperplane separates Q and $\{\bar{x}\}$:

$$\begin{array}{lll} \sup\limits_{x \in Q} f^T x & \leq a & \leq f^T \bar{x} \; [\Leftrightarrow \sup_{x \in Q} f^T x = a = f^T \bar{x} \text{ due to } \bar{x} \in Q] \\ \inf\limits_{x \in Q} f^T x & < & f^T \bar{x} \end{array}$$

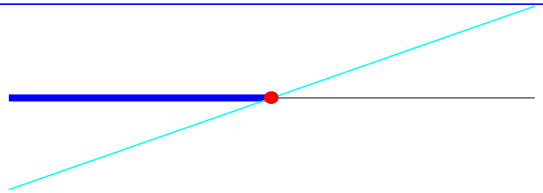
Equivalently: Hyperplane $\Pi = \{x : f^T x = a\}$ supports Q at $\bar{x}$ iff the linear form $f^T x$ attains its maximum on Q, equal to a, at the point $\bar{x}$ and the form is non-constant on Q.



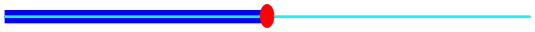
cyan: supporting hyperplane



cyan and green: supporting hyperplanes



cyan: supporting hyperplane



cyan: **NOT** a supporting hyperplane

Q: blue set in 2D; a: red point

Proposition: *Let Q be a convex closed set in $\mathbb{R}^n$ and $\bar{x}$ be a point from the relative boundary of Q. Then*

$\diamondsuit$ *There exist at least one hyperplane Π which supports Q at $\bar{x}$;*

$\diamondsuit$ *For every such hyperplane Π, the set $Q \cap \Pi$ has dimension less than the one of Q.*

Proof: Existence of supporting plane is given by Separation Theorem. This theorem is applicable since

$$\bar{x} \notin \operatorname{rint} Q \Rightarrow \{\bar{x}\} \equiv \operatorname{rint}\{\bar{x}\} \cap \operatorname{rint} Q = \emptyset.$$

Further,

$$Q \not\subset \Pi \Rightarrow \operatorname{Aff}(Q) \not\subset \Pi \Rightarrow \operatorname{Aff}(\Pi \cap Q) \subset \operatorname{Aff}(Q) \cap \Pi \subsetneqq \operatorname{Aff}(Q),$$

and if two *distinct* affine subspaces (in our case, $\operatorname{Aff}(\Pi \cap Q)$ and $\operatorname{Aff}(Q)$) are embedded one into another, then the dimension of the embedded subspace is *strictly less* than the dimension of the embedding one.

Extreme Points

♣ **<u>Definition.</u>** Let Q be a convex set in $\mathbb{R}^n$ and $\bar{x}$ be a point of Q. The point is called *extreme*, if it is not a convex combination, with positive weights, of two points of X distinct from $\bar{x}$:

$$\bar{x} \in \mathrm{Ext}(Q)$$
$$\Updownarrow$$
$$\left\{\bar{x} \in Q\right\} \ \& \ \left\{ \begin{array}{c} u, v \in Q, \lambda \in (0,1) \\ \bar{x} = \lambda u + (1-\lambda) v \end{array} \right\} \Rightarrow u = v = \bar{x} \Bigg\}$$

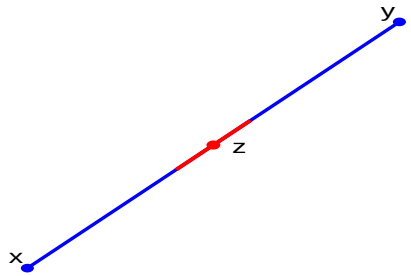
<u>Equivalently:</u> A point $\bar{x} \in Q$ is extreme iff it is **<u>not</u>** the midpoint of a nontrivial segment in Q:

$$\bar{x} \pm h \in Q \Rightarrow h = 0.$$

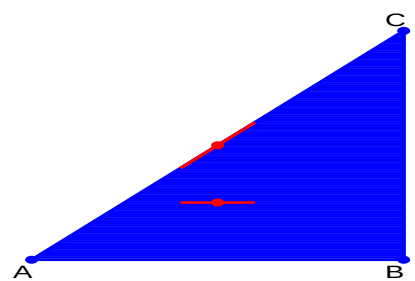
<u>Equivalently:</u> A point $\bar{x} \in Q$ is extreme iff the set $Q \backslash \{\bar{x}\}$ is convex.

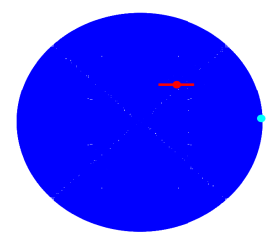
Examples:

1. Extreme points of $[x, y]$ are ...



2. Extreme points of $\triangle ABC$ are ...





3. Extreme points of the ball $\{x : \|x\|_2 \leq 1\}$ are ...

Examples:

1. Extreme points of $[x, y]$ are the endpoints x and y

2. Extreme points of $\triangle ABC$ are the vertices A, B, C

3. Extreme points of the ball $\{x : \|x\|_2 \leq 1\}$ are the points $\{x : \|x\|_2 = 1\}$ on the boundary of the ball.

<u>Theorem</u> [Krein-Milman] *Let Q be a closed convex and nonempty set in $\mathbb{R}^n$. Then*

◇ *Q possess extreme points iff Q does not contain lines;*

◇ *If Q is bounded, then Q is the convex hull of its extreme points:*

$$Q = \mathrm{Conv}(\mathrm{Ext}(Q))$$

so that every point of Q is convex combination of extreme points of Q.

<u>Note:</u> **<u>If</u>** $Q = \mathrm{Conv}(A)$, **<u>then</u>** $\mathrm{Ext}(Q) \subset A$. Thus, extreme points of a *closed convex bounded* set Q give the *minimal* representation of Q as Conv(...).

Proof. 1^0: **<u>If</u>** closed convex set Q does not contain lines, **<u>then</u>** $\mathrm{Ext}(Q) \neq \emptyset$

<u>Important lemma:</u> *Let S be a closed convex set and $\Pi = \{x : f^T x = a\}$ be a hyperplane which supports S at certain point. Then*

$$\mathrm{Ext}(\Pi \cap S) \subset \mathrm{Ext}(S).$$

Proof of Lemma. Let $\bar{x} \in \mathrm{Ext}(\Pi \cap S)$; we should prove that $\bar{x} \in \mathrm{Ext}(S)$. Assume, on the contrary, that $\bar{x}$ is a midpoint of a nontrivial segment $[u, v] \subset S$. Then $f^T \bar{x} = a = \max\limits_{x \in S} f^T x$, whence $f^T \bar{x} = \max\limits_{x \in [u,v]} f^T x$. A linear form can attain its maximum on a segment at the midpoint of the segment iff the form is constant on the segment; thus, $a = f^T \bar{x} = f^T u = f^T v$, that is, $[u, v] \subset \Pi \cap S$. But $\bar{x}$ is an extreme point of $\Pi \cap S$ – contradiction!

Let Q be a nonempty closed convex set which does not contain lines. In order to build an extreme point of Q, apply the *Purification algorithm*. It generates a sequence $Q = S_0 \supset S_1 \supset S_2 \supset ...$ of shrinking closed convex nonempty sets which starts from $S_0 = Q$, along with points $x_t \in S_t$, and is such that

A: *all extreme points of S_t, if any, are extreme points of S_{t-1}* (and therefore are extreme points of $S_0 = Q$), and

B: *whenever S_t is not a singleton, S_{t+1} is well defined and is of dimension strictly less than the dimension of S_t.*

Taking for granted that there is an algorithm capable to produce sequence with these properties, observe that the sequence $S_0 \supset S_t \supset ...$ is finite by **B** (dimension of S_t strictly decreases when passing from S_t to S_{t+1}, and this cannot last indefinitely) and the concluding set S_K in this sequence is a singleton (again by **B**). In particular, S_K has extreme point:

$$S_K = \{\bar{x}\} \Rightarrow \mathrm{Ext}(S_K) = \{\bar{x}\}$$

and by **A** this extreme point is an extreme point of $Q \Rightarrow \mathrm{Ext}(Q) \neq \emptyset$, Q.E.D.

♠ This is how Purification works:

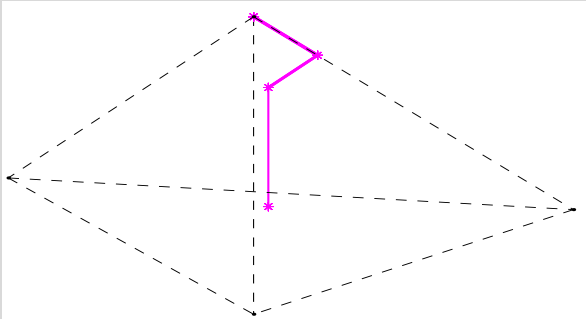
• We start with $S_0 = Q$ and select as x_0 an arbitrary point of S_0

• Given S_t, and $x_t \in S_t$ we check whether S_t is a singleton; if yes, we terminate, otherwise we

— find a point x_{t+1} on the relative boundary of S_t

— build a hyperplane Π_t supporting S_t at x_{t+1}, and set $S_{t+1} = S_t \cap \Pi_t$

Note: By construction, S_{t+1}, when defined, is a nonempty closed convex subset of S_t, with $\dim(S_{t+1}) < \dim(S_t)$ (by Proposition on Supporting Plane) and $\mathrm{Ext}(S_{t+1}) \subset \mathrm{Ext}(S_t)$ (by Important Lemma), so that we do ensure **A** and **B**.



• To find a point x_{t+1} on the relative boundary of a *non-singleton* closed convex set $S_t \ni x_t$, we take a direction $h \neq 0$ parallel to Aff(S_t).
Since $S_t \subset Q$, S_t does not contain lines
$\Rightarrow$ replacing if necessary h with $-h$, we can assume that the ray

$$\{x_t + sh : s \geq 0\}$$

is not contained in S_t, which combines with closedness of S_t to imply that the largest $s = \bar{s}$ such that $x_t + sh \in S_t$ is well defined
$\Rightarrow x_{t+1} = x_t + \bar{s}h$ is a point from the relative boundary of S_t

<u>Note:</u> Assume you are given a linear form $g^T x$ which is bounded from above on Q. Then in the Purification algorithm one can easily ensure that $g^T x_{t+1} \geq g^T x_t$. Thus,

If Q is a nonempty convex closed set in $\mathbb{R}^n$ which does not contain lines and $g^T x$ is a linear form which is bounded above on Q, then for every point $x_0 \in Q$ there exists (and can be found by Purification) a point $\bar{x} \in \mathrm{Ext}(Q)$ such that $g^T \bar{x} \geq g^T x_0$. In particular, if $g^T x$ attains its maximum on Q, then a maximizer can be found among extreme points of Q.

Proof, 2^0 **<u>If</u>** a closed convex set Q contains lines, it has no extreme points.

<u>Another Important Lemma:</u> *Let S be a closed convex set and h be such that for some $x \in S$ the ray $\{x + th : t \geq 0\}$ belongs to S. Then*

$$\{y + th : t \geq 0\} \subset S \ \forall y \in S.$$

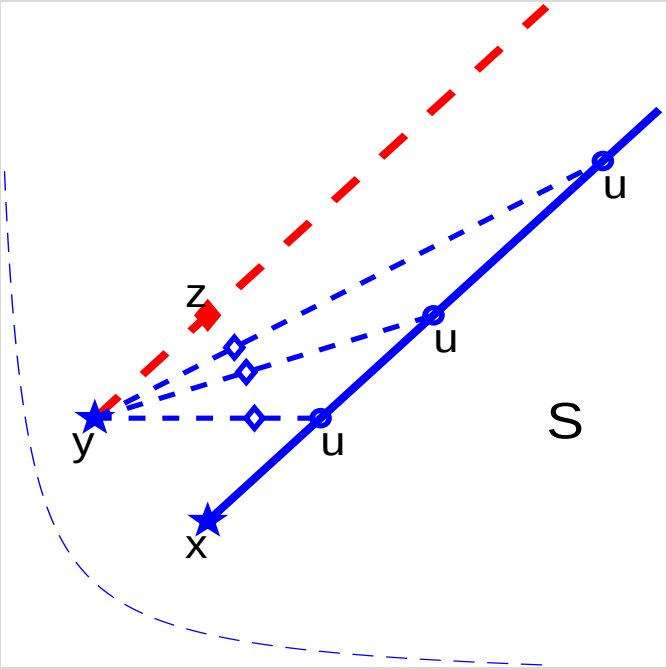
<u>Note:</u> The set of all directions $h \in \mathbb{R}^n$ such that $\{x + th : t \geq 0\} \subset S$ for some (and then, for all) $x \in S$, is called the *recessive cone* $\mathrm{Rec}(S)$ of closed convex set S. $\mathrm{Rec}(S)$ indeed is a cone, and

$$S + \mathrm{Rec}(S) = S.$$

Geometrically: *Nonzero recessive directions of S are exactly the directions of rays contained in S.*

<u>Corollary:</u> *If a closed convex set Q contains a line ℓ, then the parallel lines, passing through points of Q, also belong to Q. In particular, Q possesses no extreme points.*

Proof of Another Important Lemma: For every $s > 0$ and $y \in S$ we have $y + sh = \lim_{i\to\infty} \underbrace{[(1 - s/i)y + (s/i)[x + (i/s)h]]}_{\in S}$. $\square$



Geometrically: Given that S contains blue ray and point y, we want to prove that S contains the red ray.

Let z be a point on the red ray, and let variable point u run to ∞ along the blue ray. The segments $[y, u]$ belong to S by convexity, and the points on these segments which are at the distance $\|z - y\|_2$ from y (points $\diamond$) converge to z. *Since S is closed, $z \in S$.*

Proof, 3^0: If a nonempty closed convex set Q is bounded, then $Q = \mathrm{Conv}(\mathrm{Ext}(Q))$.

The inclusion $\mathrm{Conv}(\mathrm{Ext}(Q)) \subset Q$ is evident. Let us prove the opposite inclusion, i.e., prove that every point of Q is a convex combination of extreme points of Q.

Induction in $k = \dim Q$. Base $k = 0$ (Q is a singleton) is evident.

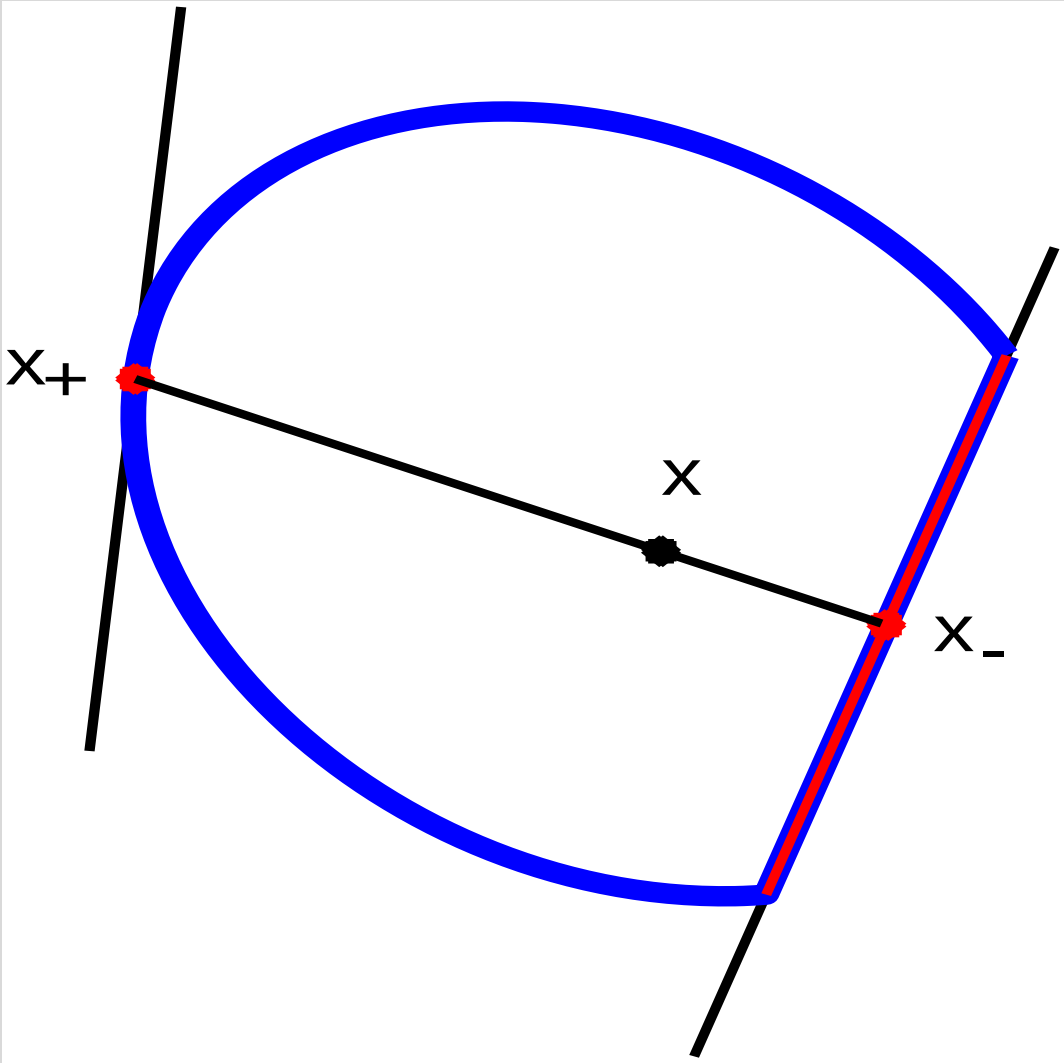
Step $k \mapsto k+1$: Given $(k+1)$-dimensional closed and bounded convex set Q and a point $x \in Q$, we can use the construction for finding a relative boundary point from the Purification algorithm to represent x as a convex combination of two points x_+ and x_- *from the relative boundary of* Q. Let Π_+ be a hyperplane which supports Q at x_+, and let $Q_+ = \Pi_+ \cap Q$. As we know, Q_+ is a closed convex set such that

$$\dim Q_+ < \dim Q,\ \mathrm{Ext}(Q_+) \subset \mathrm{Ext}(Q),\ x_+ \in Q_+.$$

Invoking inductive hypothesis,

$$x_+ \in \mathrm{Conv}(\mathrm{Ext}(Q_+)) \subset \mathrm{Conv}(\mathrm{Ext}(Q)).$$

Similarly, $x_- \in \mathrm{Conv}(\mathrm{Ext}(Q))$. Since $x \in [x_-, x_+]$, we get $x \in \mathrm{Conv}(\mathrm{Ext}(Q))$.


x+
x
x-

Structure of Polyhedral Sets

♣ **Definition:** A *polyhedral* set Q in $\mathbb{R}^n$ is a subset in $\mathbb{R}^n$ which is a solution set of a finite system of nonstrict linear inequalities:

$$Q \text{ is polyhedral} \Leftrightarrow Q = \{x : Ax \geq b\}.$$

♠ *Every polyhedral set is convex and closed.*

In the sequel, the polyhedral sets in question are assumed to be nonempty.

Question: When a polyhedral set $Q = \{x : Ax \geq b\}$ contains lines? What are these lines, if any?
Answer: Q contains lines iff A has a nontrivial nullspace:

$$\mathrm{Null}(A) \equiv \{h : Ah = 0\} \neq \{0\}.$$

Indeed, a line $\ell = \{x = \bar{x} + th : t \in \mathbb{R}\}$, $h \neq 0$, belongs to Q iff

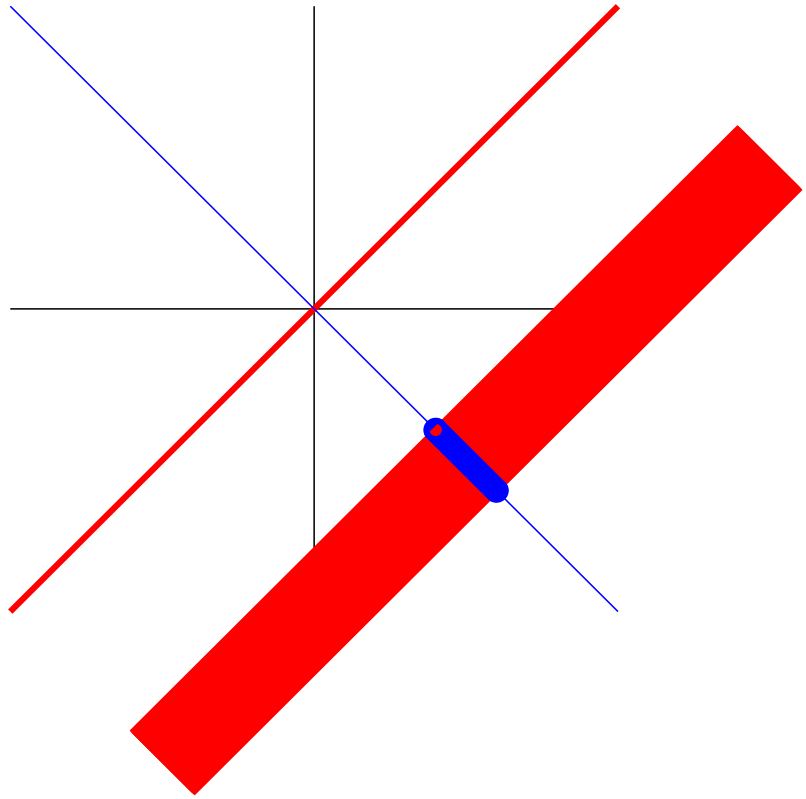
$$\begin{array}{rl} & \forall t : A(\bar{x} + th) \geq b \\ \Leftrightarrow & \forall t : tAh \geq b - A\bar{x} \\ \Leftrightarrow & Ah = 0 \ \& \ \bar{x} \in Q. \end{array}$$

Fact: *A polyhedral set $Q = \{x : Ax \geq b\}$ always can be represented as*

$$Q = Q_* + L,$$

where Q_* is a polyhedral set which does not contain lines and L is a linear subspace. In this representation,
◇ *L is uniquely defined by Q and coincides with* $\mathrm{Null}(A)$,
◇ *Q_* can be chosen, e.g., as*

$$Q_* = Q \cap L^{\perp}$$



- Red stripe Q: polyhedral set containing lines
- red line L: the recessive subspace of Q
- Blue segment: $Q_* = Q \cap L^\perp$

♠ *Red stripe Q = blue segment Q_* + red subspace L*

♠ *Blue segment Q_*: polyhedral set not containing lines*

Structure of polyhedral set which does <u>not</u> contain lines

♣ **<u>Theorem:</u>** *Let*

$$Q = \{x : Ax \geq b\} \neq \emptyset$$

be a polyhedral set which does not contain lines (or, which is the same, $\mathrm{Null}(A) = \{0\}$). *Then the set* $\mathrm{Ext}(Q)$ *of extreme points of* Q *is nonempty and finite, and*

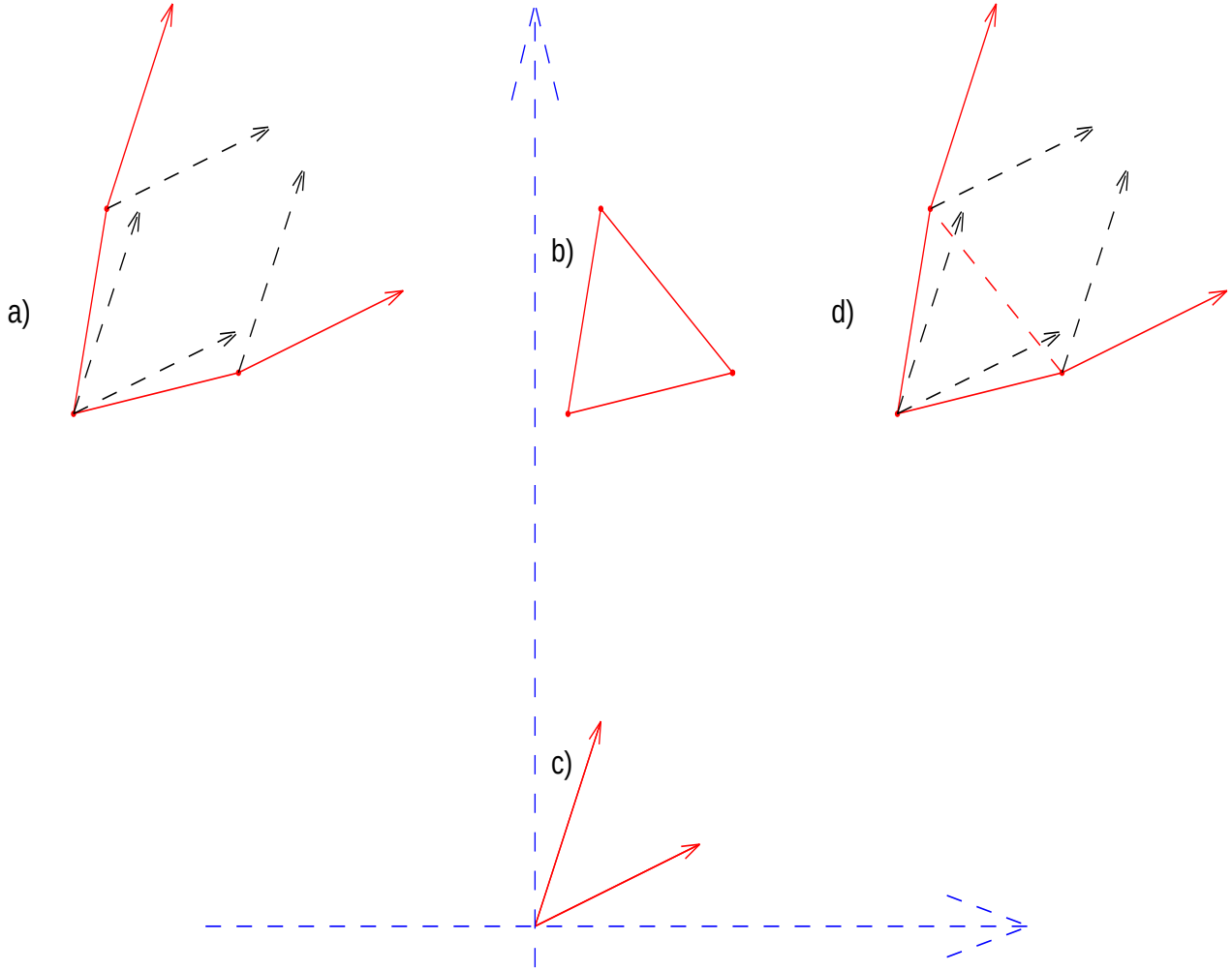
$$\begin{array}{rcl} Q & = & \mathrm{Conv}(\mathrm{Ext}(Q)) + \mathrm{Cone}\,\{r_1, ..., r_S\} \\ & = & \mathrm{Conv}\{v_1, ..., v_T\} + \mathrm{Cone}\,\{r_1, ..., r_S\} \\ & = & \left\{ x = \sum_r \lambda_t v_t + \sum_s \mu_s r_s : \begin{array}{l} \lambda_t \geq 0, \sum_t \lambda_t = 1 \\ \mu_s \geq 0 \end{array} \right\} \end{array} \qquad (*)$$

for properly chosen vectors $r_1, ..., r_S$.

<u>Note:</u> $\mathrm{Cone}\,\{r_1, ..., r_s\}$ is exactly the recessive cone of Q:

$$\begin{array}{rl} & \mathrm{Cone}\,\{r_1, ..., r_S\} \\ = & \{r : x + tr \in Q \ \forall (x \in Q, t \geq 0)\} \\ = & \{r : Ar \geq 0\}. \end{array}$$

This cone is the trivial cone $\{0\}$ iff Q is a *bounded* polyhedral set (called *polytope*).



a): a polyhedral set

b): $\{\sum_{i=1}^{3} \lambda_i v_i : \lambda_i \geq 0, \sum_{i=1}^{3} \lambda_i = 1\}$

c): $\{\sum_{j=1}^{2} \mu_j r_j : \mu_j \geq 0\}$

d): The set a) is the sum of sets b) and c)

Note: shown are the boundaries of the sets.

♣ Combining the above theorems, we come to the following results:

A (nonempty) polyhedral set Q always can be represented in the form

$$Q = \left\{ x = \sum_{i=1}^{I} \lambda_i v_i + \sum_{j=1}^{J} \mu_j w_j : \begin{array}{l} \lambda \geq 0, \mu \geq 0 \\ \sum\limits_i \lambda_i = 1 \end{array} \right\} \qquad (!)$$

where I, J are positive integers and $v_1, ..., v_I$, $w_1, ..., w_J$ are appropriately chosen points and directions.

<u>Vice versa,</u> *every set Q of the form* (!) *is a polyhedral set.*

<u>Note:</u> Polytopes (nonempty bounded polyhedral sets) are *exactly* the sets of form (!) with "trivial w-part": $w_1 = ... = w_J = 0$.

$$Q \neq \emptyset, \ \& \ \exists A, b : Q = \{x : Ax \geq b\}$$

$$\Updownarrow$$

$$\exists (I, J, v_1, ..., v_I, w_1, ..., w_J) :$$

$$Q = \left\{ x = \sum_{i=1}^{I} \lambda_i v_i + \sum_{j=1}^{J} \mu_j w_j : \begin{array}{l} \lambda \geq 0, \mu \geq 0 \\ \sum_i \lambda_i = 1 \end{array} \right\}$$

<u>Exercise 1:</u> Is it true that the intersection of two polyhedral sets is a polyhedral set?

<u>Exercise 2:</u> Is it true that the affine image $\{y = Px + p : x \in Q\}$ of a polyhedral set Q is a polyhedral set?

Applications to Linear Programming

♣ Consider a *feasible* Linear Programming program

$$\min_x c^T x \text{ s.t. } x \in Q = \{x : Ax \geq b\} \qquad \text{(LP)}$$

Observation: We lose nothing when assuming that Null$(A) = \{0\}$.
Indeed, we have

$$Q = Q_* + \text{Null}(A),$$

where Q_* is a polyhedral set not containing lines. If c is not orthogonal to Null(A), then (LP) clearly is unbounded. If c is orthogonal to Null(A), then (LP) is equivalent to the LP program

$$\min_x c^T x \text{ s.t. } x \in Q_*,$$

and now the matrix in a representation $Q_* = \{x : \widetilde{A}x \geq \widetilde{b}\}$ has trivial nullspace.

Assuming Null$(A) = \{0\}$, let (LP) be bounded (and thus solvable). Since Q is convex, closed and does not contain lines, in the (nonempty!) set of minimizers of the objective on Q there is an extreme point of Q.

$$\min_x c^T x \text{ s.t. } x \in Q = \{x : Ax \geq b\} \qquad \text{(LP)}$$

We have proved

Proposition: *Assume that (LP) is feasible and bounded (and thus is solvable) and that* $\text{Null}(A) = \{0\}$. *Then among optimal solutions to (LP) there exists at least one which is an extreme point of* Q.

Question: How to characterize extreme points of the set

$$Q = \{x \in \mathbb{R}^n : Ax \geq b\} \text{ ?}$$

Answer [Algebraic Characterization of Extreme Points of Polyhedral Set]: Extreme points $\bar{x}$ of Q are fully characterized by the following two properties:

$\diamondsuit$ $\bar{x} \in Q$, that is, $A\bar{x} \geq b$

$\diamondsuit$ Among constraints $Ax \geq b$ which are *active* at $\bar{x}$ (i.e., are satisfied as equalities), there are n linearly independent (i.e., with linearly independent vectors of coefficients).

Justification of the answer, $\Rightarrow$: **<u>If</u>** $\bar{x}$ is an extreme point of Q, **<u>then</u>** among the constraints $Ax \geq b$ active at $\bar{x}$ there are n linearly independent.
W.l.o.g., assume that the constraints active at $\bar{x}$ are the first k constraints

$$a_i^T x \geq b_i,\ i = 1, ..., k.$$

We should prove that among n-dimensional vectors $a_1, ..., a_k$, there are n linearly independent. Assuming otherwise, there exists a nonzero vector h such that $a_i^T h = 0$, $i = 1, ..., k$, that is,

$$a_i^T[\bar{x} \pm \epsilon h] = a_i^T \bar{x} = b_i,\ i = 1, ..., k$$

for all $\epsilon > 0$. Since the remaining constraints $a_i^T x \geq b_i$, $i > k$, are strictly satisfied at $\bar{x}$, we conclude that

$$a_i^T[\bar{x} \pm \epsilon h] \geq b_i,\ i = k+1, ..., m$$

for all small enough values of $\epsilon > 0$.
We conclude that $\bar{x} \pm \epsilon h \in Q = \{x : Ax \geq b\}$ for all small enough $\epsilon > 0$.
Since $h \neq 0$ and $\bar{x}$ is an extreme point of Q, we get a contradiction.

Justification of the answer, $\Leftarrow$: **<u>If</u>** $\bar{x} \in Q$ makes equalities n of the constraints $a_i^T x \geq b_i$ with linearly independent vectors of coefficients, **<u>then</u>** $\bar{x} \in \mathrm{Ext}(Q)$.

W.l.o.g., assume that n active at $\bar{x}$ constraints with linearly independent vectors of coefficients are the first n constraints

$$a_i^T x \geq b_i,\ i = 1, ..., n.$$

We should prove that if h is such that $\bar{x} \pm h \in Q$, then $h = 0$. Indeed, we have

$$\bar{x} \pm h \in Q \Rightarrow a_i^T [\bar{x} \pm h] \geq b_i,\ i = 1, ..., n;$$

since $a_i^T \bar{x} = b_i$ for $i \leq n$, we get

$$a_i^T \bar{x} \pm a_i^T h = a_i^T [\bar{x} \pm h] \geq a_i^T \bar{x},\ i = 1, ..., n,$$

whence

$$a_i^T h = 0,\ i = 1, ..., n. \qquad (*)$$

Since n-dimensional vectors $a_1, ..., a_n$ are linearly independent, $(*)$ implies that $h = 0$, Q.E.D.

Example: Given integer $k \leq n$, let us list extreme point of the set

$$\Delta_{k,n} = \{x \in \mathbb{R}^n : 0 \leq x_i \leq 1 \, \forall i, \sum_i x_i = k\}$$

• At an extreme point v, n of the constraints should become active. One of these constraints is $\sum_i x_i = k$, and $n-1$ of the remaining active constraints should be among the bounds $1 \geq x_i \geq 0$

$\Rightarrow$ at least $n-1$ of entries in v are zeros and ones

$\Rightarrow$ all entries in v are integers (since all but one are so, and the sum of all entries is integer)

$\Rightarrow$ all entries are zeros and ones

$\Rightarrow$ *all nonzero entries are equal to 1, and there are k of them.*

• Reasoning can be reversed, implying that *every 0/1 vector with exactly k entries equal to 1 is an extreme point of $\Delta_{k,n}$.*

Question: *What are extreme points of the set*

$$\{x \in \mathbb{R}^n : 0 \leq x_i \leq 1 \, \forall i, \sum_i x_i = 2.5\} \qquad ?$$

Polyhedral sets with MUST TO KNOW extreme points

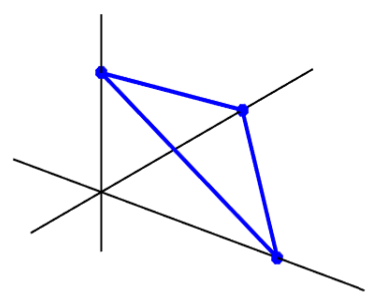
A. Let $k \leq n$ be positive integers.

A.1. *The extreme points of the set*

$$\left\{x \in \mathbb{R}^n : 0 \leq x_i \leq 1 \, \forall i, \sum_i x_i = k\right\}$$

are exactly Boolean vectors from the set, that is, 0/1 vectors with exactly k entries equal to 1.

In particular, the extreme points of the "flat (a.k.a. *probabilistic*) simplex"



$$\{x \in \mathbb{R}^n : x \geq 0, \sum_i x_i = 1\}$$

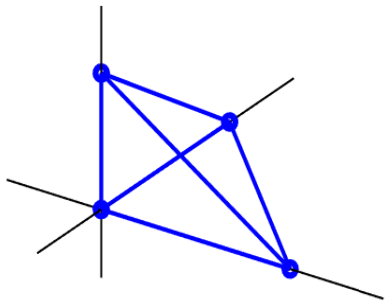
are the standard basic orths (set $k = 1$).

A.2. *The extreme points of the set*

$$\left\{x \in \mathbb{R}^n : 0 \leq x_i \leq 1 \, \forall i, \sum_i x_i \leq k\right\}$$

are exactly Boolean vectors from the set, that is, 0/1 vectors with at most k entries equal to 1.

In particular, the extreme points of the "full-dimensional simplex"



$$\{x \in \mathbb{R}^n : x \geq 0, \sum_i x_i \leq 1\}$$

are the standard basic orths *and* the origin (set $k = 1$).

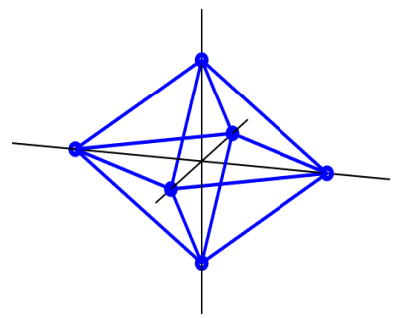
A.3. *The extreme points of the set*

$$\left\{x \in \mathbb{R}^n : |x_i| \le 1\,\forall i, \sum_i |x_i| \le k\right\}$$

are exactly the vectors with k nonzero entries equal to ± 1 each.

In particular,

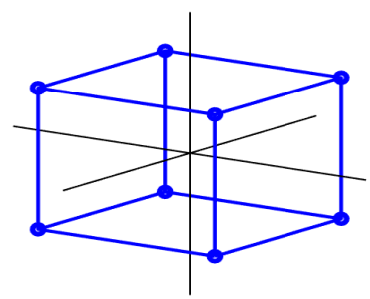
- the extreme points of the unit ℓ_1-ball



$$\{x \in \mathbb{R}^n : \|x\|_1 \le 1\} = \{x \in \mathbb{R}^n : \sum_i |x_i| \le 1\}$$

are the plus-minus standard basic orths (set $k = 1$).

- the extreme points of the unit ℓ_∞-ball



$$\{x \in \mathbb{R}^n : \|x\|_\infty \le 1\} = \{x \in \mathbb{R}^n : -1 \le x_i \le 1\,\forall i\}$$

are ± 1 vectors (set $k = n$).

B. A double-stochastic matrix is a *square matrix with nonnegative entries and all row and column sums equal to 1*. $n \times n$ double-stochastic matrices form a polytope $\mathcal{P}_n$ in the space $\mathbb{R}^{n\times n}$ of $n \times n$ matrices:

$$\mathcal{P}_n = \{[x_{ij}] \in \mathbb{R}^{n\times n} : x_{ij} \geq 0 \,\forall (i,j), \sum_j x_{ij} = 1 \,\forall i, \sum_i x_{ij} = 1 \,\forall j\}$$

Birkhoff's Theorem: *The extreme points of $\mathcal{P}_n$ are exactly the Boolean matrices from the set, that is, permutation matrices – those with exactly one nonzero entry, equal to 1, in every row and in every column.*

Note: Permutation matrices P are exactly the matrices of linear transformations $x \mapsto Px$ which permute the entries in the argument. Such a matrix is specified by the corresponding permutation, and there are $n!$ of them.

Essence of the proof is in the following fact: *If x is an extreme point of $\mathcal{P}$, then matrix x has an entry equal to 1*

$\Rightarrow$ all other entries in the row and the column of the unit entry are zeros

$\Rightarrow$ eliminating from x the row and the column of the unit entry, we get an $(n-1) \times (n-1)$ double-stochastic matrix.

Claim: *If x is an extreme point of the polytope $\mathcal{P}$ of double stochastic matrices, then matrix x has an entry equal to 1*

Proof: "As is", $\mathcal{P}$ is given by $2n$ linear equalities stating that all row and all column sums in matrix x are equal to 1 plus n^2 inequalities $x_{ij} \geq 0$.

• In fact, we can drop one of the equalities without changing $\mathcal{P}$: if all column sums and all but one row sums are equal to 1, then all row and column sums are equal to 1.

Indeed, the total of all n row sums is equal to the total of all n column sums – both these totals are the sums of all entries in the matrix, and "In fact" follows.

$\Rightarrow$ We lose nothing when assuming that $\mathcal{P}$ is given by n^2 inequalities $x_{ij} \geq 0$ and $2n-1$ linear equalities.

• By algebraic characterization of extreme points, *at an extreme point $\bar{x}$ of $\mathcal{P}$ n^2 of the above constraints should become active*

$\Rightarrow$ *at least $n^2 - 2n + 1$ entries in $\bar{x}$ are zeros*

$\Rightarrow$ *there is a column in $\bar{x}$ with at least $n-1$ zero entries*, since otherwise the total # of zero entries would be at most $n(n-2) < n^2 - 2n + 1$

In the column with at least $n-1$ zero entries the sum of entries is 1, implying that in this column there is exactly one nonzero entry, and this entry is equal to 1.

Lecture 5:
Convex Functions

Convex Functions

Definition: Let f be a real-valued function defined on a nonempty subset $\mathrm{Dom} f$ in $\mathbb{R}^n$. f is called *convex*, if

$\diamondsuit$ $\mathrm{Dom} f$ is a convex set

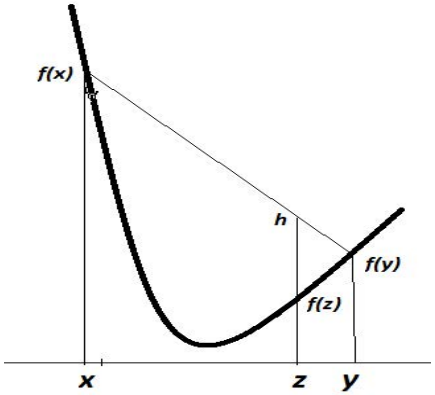
$\diamondsuit$ for all $x, y \in \mathrm{Dom} f$ and $\lambda \in [0, 1]$ one has

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda) f(y)$$

Equivalent definition: Let f be a real-valued function defined on a nonempty subset $\mathrm{Dom} f$ in $\mathbb{R}^n$. The function is called convex, if its *epigraph* – the set

$$\mathrm{Epi}\{f\} = \{(x, t) \in \mathbb{R}^{n+1} : f(x) \leq t\}$$

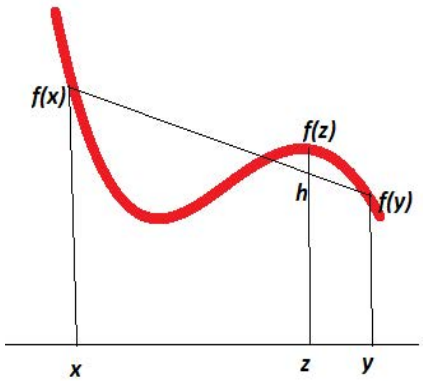
is a convex set in $\mathbb{R}^{n+1}$.



convex function: graph on $[x; y]$ is below secant

$z = \lambda x + (1 - \lambda)y$

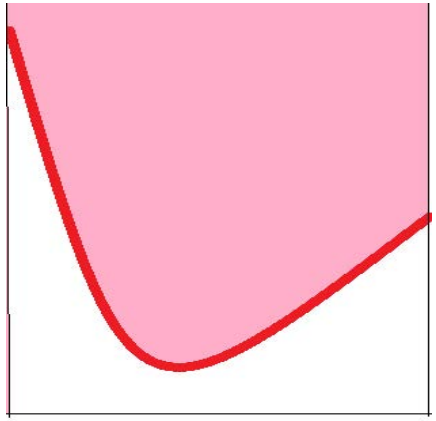
$f(z) \leq h = \lambda f(x) + (1 - \lambda)f(y)$



nonconvex function: graph on $[x; y]$ is *not* entirely below secant

$z = \lambda x + (1 - \lambda)y$

$f(z) > h = \lambda f(x) + (1 - \lambda)f(y)$



epigraph of convex function

What does the definition of convexity actually mean?

The inequality

$$f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda) f(y) \qquad (*)$$

where $x, y \in \mathrm{Dom} f$ and $\lambda \in [0,1]$ is automatically satisfied when $x = y$ or when $\lambda = 0/1$. Thus, it says something only when the points x, y are distinct from each other and the point $z = \lambda x + (1-\lambda)y$ is a (relative) interior point of the segment $[x, y]$. What does $(*)$ say in this case?

$\diamondsuit$ Observe that $z = \lambda x + (1-\lambda)y = x + (1-\lambda)(y-x)$, whence

$$\|y-x\| : \|y-z\| : \|z-x\| = 1 : \lambda : (1-\lambda)$$

Therefore

$$f(z) \le \lambda f(x) + (1-\lambda) f(y) \qquad (*)$$

$$\Updownarrow$$

$$f(z) - f(x) \le \underbrace{(1-\lambda)}_{\frac{\|z-x\|}{\|y-x\|}}(f(y) - f(x))$$

$$\Updownarrow$$

$$\frac{f(z)-f(x)}{\|z-x\|} \le \frac{f(y)-f(x)}{\|y-x\|}$$

Similarly,

$$f(z) \leq \lambda f(x) + (1-\lambda) f(y) \qquad (*)$$

$$\Updownarrow$$

$$\underbrace{\lambda}_{\frac{\|y-z\|}{\|y-x\|}} (f(y) - f(x)) \leq f(y) - f(z)$$

$$\Updownarrow$$

$$\frac{f(y)-f(x)}{\|y-x\|} \leq \frac{f(y)-f(z)}{\|y-z\|}$$

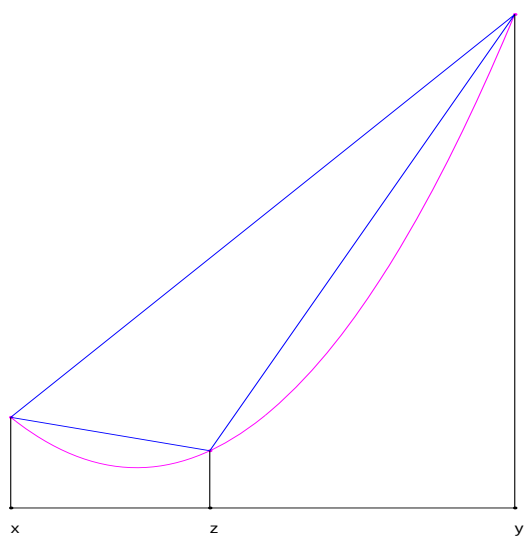
Conclusion: f is convex iff for every three distinct points x, y, z such that $x, y \in \mathrm{Dom} f$ and $z \in [x, y]$, we have $z \in \mathrm{Dom} f$ and

$$\frac{f(z) - f(x)}{\|z - x\|} \leq \frac{f(y) - f(x)}{\|y - x\|} \leq \frac{f(y) - f(z)}{\|y - z\|} \qquad (*)$$

Note: *From 3 inequalities in* $(*)$*:*

$$\frac{f(z)-f(x)}{\|z-x\|} \leq \frac{f(y)-f(x)}{\|y-x\|}, \qquad \frac{f(y)-f(x)}{\|y-x\|} \leq \frac{f(y)-f(z)}{\|y-z\|}, \qquad \frac{f(z)-f(x)}{\|z-x\|} \leq \frac{f(y)-f(z)}{\|y-z\|}$$

every single one implies the other two.



Jensen's Inequality: *Let $f(x)$ be a convex function. Then*

$$x_i \in \mathrm{Dom} f, \lambda_i \geq 0, \sum_i \lambda_i = 1 \Rightarrow$$
$$f(\sum_i \lambda_i x_i) \leq \sum_i \lambda_i f(x_i)$$

Proof: The points $(x_i, f(x_i))$ belong to $\mathrm{Epi}\{f\}$. Since this set is convex, the point

$$(\sum_i \lambda_i x_i, \sum_i \lambda_i f(x_i)) \in \mathrm{Epi}\{f\}.$$

By definition of the epigraph, it follows that

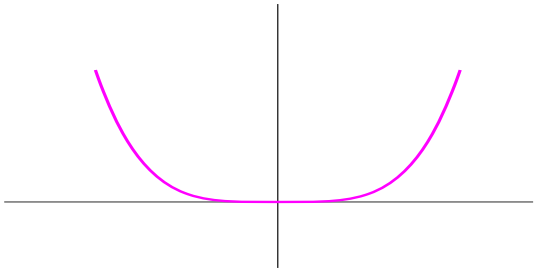
$$f(\sum_i \lambda_i x_i) \leq \sum_i \lambda_i f(x_i).$$

Extension: Let f be convex, $\mathrm{Dom} f$ be closed and f be continuous on $\mathrm{Dom} f$. Consider a probability distribution $\pi(dx)$ supported on $\mathrm{Dom} f$. Then

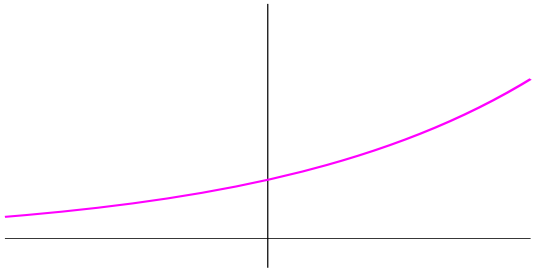
$$f(\mathbf{E}_\pi\{x\}) \leq \mathbf{E}_\pi\{f(x)\}.$$

Examples:

◇ **Functions convex on $\mathbb{R}$:**

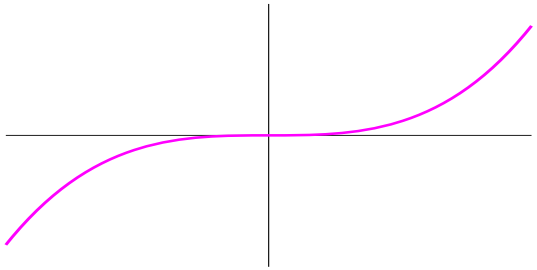
- 

  $x^2, x^4, x^6, \ldots$

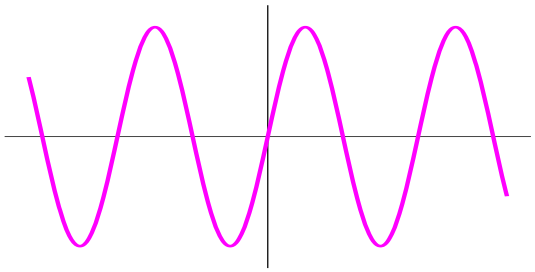
- 

  $\exp\{x\}$

◇ **Nonconvex functions on $\mathbb{R}$:**

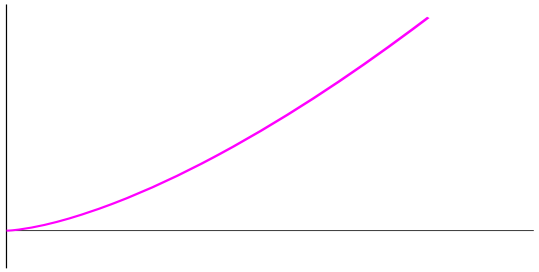
- 

  x^3

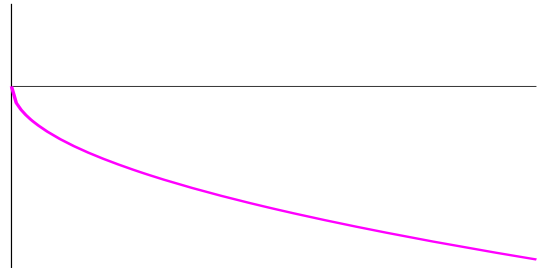
- 

  $\sin(x)$

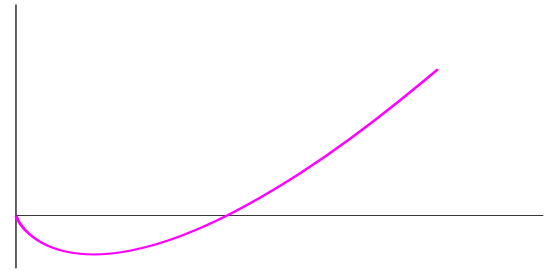
◇ **Functions convex on $\mathbb{R}_+$:**

- 

  x^p, $p \geq 1$

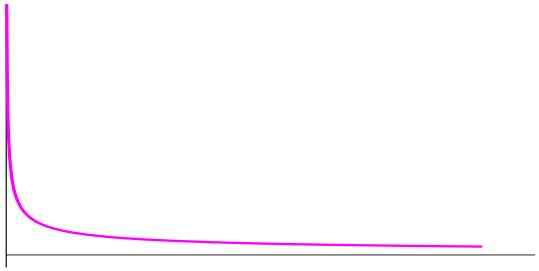
- 

  $-x^p$, $0 \leq p \leq 1$

- 

  $x \ln x$

◇ **Functions convex on $\mathbb{R}_{++} = \operatorname{int} \mathbb{R}_+ = \{x > 0\}$:**

- 

  $1/x^p$, $p > 0$

♢ Functions convex on $\mathbb{R}^n$:

- affine function $f(x) = f^T x$
- A norm $\|\cdot\|$ on $\mathbb{R}^n$ is a convex function:

$$\begin{aligned} \|\lambda x + (1-\lambda)y\| &\leq \|\lambda x\| + \|(1-\lambda)y\| \\ &\quad \text{[Triangle inequality]} \\ &= \lambda\|x\| + (1-\lambda)\|y\| \\ &\quad \text{[homogeneity]} \end{aligned}$$

Application of Jensen's Inequality: Let $p = \{p_i > 0\}_{i=1}^n$, $q = \{q_i > 0\}_{i=1}^n$ be two discrete probability distributions.

Claim: The *Kullback-Liebler distance*

$$\sum_i p_i \ln \frac{p_i}{q_i}$$

between the distributions is ≥ 0.

Indeed, the function $f(x) = -\ln x$, $\mathrm{Dom} f = \{x > 0\}$, is convex. Setting $x_i = q_i/p_i$, $\lambda_i = p_i$ we have

$$\begin{aligned} 0 &= -\ln\left(\sum_i q_i\right) = f(\sum_i p_i x_i) \\ &\leq \sum_i p_i f(x_i) = \sum_i p_i(-\ln q_i/p_i) \\ &= \sum_i p_i \ln(p_i/q_i) \end{aligned}$$

What is the value of a convex function outside its domain?

Convention. To save words, it is convenient to think that a convex function f is defined *everywhere* on $\mathbb{R}^n$ and takes real values *and value* $+\infty$. With this interpretation, f "remembers" its domain:

$$\mathrm{Dom} f = \{x : f(x) \in \mathbb{R}\}$$
$$x \notin \mathrm{Dom} f \Rightarrow f(x) = +\infty$$

and the definition of convexity becomes

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \ \forall \begin{array}{c} x, y \in \mathbb{R}^n \\ \lambda \in [0,1] \end{array}$$

where the arithmetics of $+\infty$ and reals is given by the rules

$$\begin{array}{l} +\infty \leq +\infty \\ a \in \mathbb{R} \Rightarrow a + (+\infty) = (+\infty) + (+\infty) = +\infty \\ 0 \cdot (+\infty) = 0 \\ \lambda > 0 \Rightarrow \lambda \cdot (+\infty) = +\infty \end{array}$$

Note: Operations like $(+\infty) - (+\infty)$ or $(-5) \cdot (+\infty)$ are undefined!

♣ **Convexity-preserving operations:**

◇ **Taking conic combinations:** If $f_i(x)$ are convex function on $\mathbb{R}^n$ and $\lambda_i \geq 0$, then the function $\sum_i \lambda_i f_i(x)$ is convex

◇ **Affine substitution of argument:** If $f(x)$ is convex function on $\mathbb{R}^n$ and $x = Ay + b$ is an affine mapping from $\mathbb{R}^k$ to $\mathbb{R}^n$, then the function $g(y) = f(Ay + b)$ is convex on $\mathbb{R}^k$

◇ **Taking supremum:** If $f_\alpha(x)$, $\alpha \in \mathcal{A}$, is a family of convex function on $\mathbb{R}^n$, then the function $\sup_{\alpha \in \mathcal{A}} f_\alpha(x)$ is convex.

Proof: $\mathrm{Epi}\{\sup_\alpha f_\alpha(\cdot)\} = \bigcap_\alpha \mathrm{Epi}\{f_\alpha(\cdot)\}$, and intersections of convex sets are convex.

◇ **Superposition Theorem:** *Let $f_i(x)$ be convex functions on $\mathbb{R}^n$, $i = 1, ..., m$, and $F(y_1, ..., y_m)$ be a convex and monotone function on $\mathbb{R}^m$. Then the function*

$$g(x) = \begin{cases} F(f_1(x), ..., f_m(x)) & , x \in \mathrm{Dom} f_i, \forall i \\ +\infty & , \textit{otherwise} \end{cases}$$

is convex.

$\diamondsuit$ **Projective transformation:** Let $f(x)$ be a convex function of $x \in \mathbb{R}^n$. Then the function $F(\alpha, x) = \alpha f(x/\alpha) : \{\alpha > 0\} \times \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is convex.

Indeed, we need to verify that if $x, x' \in \mathbb{R}^n$, $\alpha, \alpha' > 0$ and $\lambda \in (0, 1)$, then

$$[\lambda\alpha + (1-\lambda)\alpha']f([\lambda x + (1-\lambda)x']/[\lambda\alpha + (1-\lambda)\alpha']) \leq \lambda\alpha f(x/\alpha) + (1-\lambda)\alpha' f(x/\alpha'),$$

or, which is the same,

$$f\left(\frac{\lambda x + (1-\lambda)x'}{\lambda\alpha + (1-\lambda)\alpha'}\right) \leq \underbrace{\left[\frac{\lambda\alpha}{\lambda\alpha + (1-\lambda)\alpha'}\right]}_{p} f(x/\alpha) + \underbrace{\left[\frac{(1-\lambda)\alpha'}{\lambda\alpha + (1-\lambda)\alpha'}\right]}_{q} f(x'/\alpha') \qquad (??)$$

Note that $p, q > 0$ and $p + q = 1$, so that by convexity of f we have

$$pf(x/\alpha) + qf(x'/\alpha') \geq f(\underbrace{px/\alpha + qx'/\alpha'}_{=\frac{\lambda x + (1-\lambda)x'}{\lambda\alpha + (1-\lambda)\alpha'}}),$$

as required in (??).

Illustration: The function $\alpha \ln(\alpha/\beta)$ is convex in the quadrant $\{\alpha > 0, \beta > 0\}$.

Indeed, the function is projective transformation of the convex function

$$f(\beta) = \begin{cases} -\ln(\beta) & , \beta > 0 \\ +\infty & , \beta \leq 0 \end{cases}$$

◇ **Partial minimization:** Let $f(x,y)$ be a convex function of $z=(x,y)\in\mathbb{R}^n$, and let

$$g(x)=\inf_y f(x,y).$$

Then the function $g(x)$ is convex on every convex set Q on which g does not take value $-\infty$.

Proof: Let Q be a convex set such that g does not take value $-\infty$ on Q. Let us check the Convexity Inequality

$$g(\lambda x'+(1-\lambda)x'')\le\lambda g(x')+(1-\lambda)g(x'') \qquad [\lambda\in[0,1],x',x''\in Q]$$

There is nothing to check when $\lambda=0$ or $\lambda=1$, so let $0<\lambda<1$. In this case, there is nothing to check when $g(x')$ or $g(x'')$ is $+\infty$, so let $g(x')<+\infty$, $g(x'')<+\infty$. Since $g(x')<+\infty$, for every $\epsilon>0$ there exists y' such that $f(x',y')\le g(x')+\epsilon$. Similarly, there exists y'' such that $f(x'',y'')\le g(x'')+\epsilon$. Now,

$$\begin{array}{rl} & g(\lambda x'+(1-\lambda)x'') \\ \le & f(\lambda x'+(1-\lambda)x'',\lambda y'+(1-\lambda)y'') \\ \le & \lambda f(x',y')+(1-\lambda)f(x'',y'') \\ \le & \lambda(g(x')+\epsilon)+(1-\lambda)(g(x'')+\epsilon) \\ = & \lambda g(x')+(1-\lambda)g(x'')+\epsilon \end{array}$$

Since $\epsilon>0$ is arbitrary, we get

$$g(\lambda x'+(1-\lambda)x'')\le\lambda g(x')+(1-\lambda)g(x'').$$

How to detect convexity?

Convexity is one-dimensional property:

- A set $X \subset \mathbb{R}^n$ is convex iff the set

$$\{t : a + th \in X\}$$

is, for every (a, h), a convex set on the axis

- A function f on $\mathbb{R}^n$ is convex iff the function

$$\phi(t) = f(a + th)$$

is, for every (a, h), a convex function on the axis.

♣ When a function ϕ on the axis is convex?

Let ϕ be convex and finite on (a, b). This is exactly the same as

$$\frac{\phi(z) - \phi(x)}{z - x} \leq \frac{\phi(y) - \phi(x)}{y - x} \leq \frac{\phi(y) - \phi(z)}{y - z}$$

when $a < x < z < y < b$. Assuming that $\phi'(x)$ and $\phi'(y)$ exist and passing to limits as $z \to x + 0$ and $z \to y - 0$, we get

$$\phi'(x) \leq \frac{\phi(y) - \phi(x)}{y - x} \leq \phi'(y)$$

that is, $\phi'(x)$ *is nondecreasing on the set of points from* (a, b) *where it exists*.

The following conditions are necessary and sufficient for convexity of a univariate function:

$\diamondsuit$ <u>The domain</u> of the function ϕ should be an open interval $\Delta = (a, b)$, possibly with added endpoint(s) (provided that the corresponding endpoint(s) is/are finite)

$\diamondsuit$ ϕ should be continuous on (a, b) and differentiable everywhere, except, perhaps, a countable set, *and the derivative should be monotonically non-decreasing*

$\diamondsuit$ at an endpoint of (a, b) which belongs to $\mathrm{Dom}\,\phi$, ϕ is allowed to "jump up", but not to jump down:

$$a \in \mathrm{Dom} f \Rightarrow f(a) \geq \lim_{x \to a+0} f(x); \quad b \in \mathrm{Dom} f \Rightarrow f(b) \geq \lim_{x \to b-0} f(x)$$

♣ **<u>Sufficient condition for convexity</u>** of a univariate function ϕ: $\mathrm{Dom}\phi$ is convex, ϕ is continuous on $\mathrm{Dom}\phi$ and is twice continuously differentiable, *with nonnegative* ϕ'', on $\mathrm{int}\,\mathrm{Dom}\phi$.

Indeed, we should prove that under the condition, if $x < z < y$ are in $\mathrm{Dom}\phi$, then

$$\frac{\phi(z) - \phi(x)}{z - x} \leq \frac{\phi(y) - \phi(z)}{y - z}$$

By Lagrange Theorem, the left ratio is $\phi'(\xi)$ for certain $\xi \in (x, z)$, and the right ratio is $\phi'(\eta)$ for certain $\eta \in (z, y)$. Since $\phi''(\cdot) \geq 0$ and $\eta > \xi$, we have $\phi'(\eta) \geq \phi'(\xi)$, Q.E.D.

♣ **<u>Sufficient condition for convexity</u>** of a multivariate function f: $\mathrm{Dom} f$ is convex and with a nonempty interior, f is continuous on $\mathrm{Dom} f$ and is twice continuously differentiable, *with positive semidefinite Hessian matrix* f'', on $\mathrm{int}\,\mathrm{Dom} f$.

Recall: A symmetric matrix H is called *positive semidefinite*, if $h^T H h \geq 0$ for all h. *Positive semidefiniteness of* f'' *on* $\mathrm{int}\,\mathrm{Dom} f$ *is the same as nonnegativity of the second order directional derivative of* f *taken at any point* $x \in \mathrm{int}\,\mathrm{Dom} f$ *along every direction* $h \in \mathbb{R}^n$

$$\frac{d^2}{dt^2}\Big|_{t=0} f(x+th) \geq 0$$

Instructive example: The function

$$f(x) = \ln(\sum_{i=1}^{n} \exp\{x_i\})$$

is convex on $\mathbb{R}^n$.

Indeed,

$$\left.\frac{d}{dt}\right|_{t=0} f(x+th) = h^T f'(x) \;=\; \frac{\sum_i \exp\{x_i\} h_i}{\sum_i \exp\{x_i\}}$$

$$\left.\frac{d^2}{dt^2}\right|_{t=0} f(x+th) = h^T f''(x) h \;=\; -\frac{\left(\sum_i \exp\{x_i\} h_i\right)^2}{\left(\sum_i \exp\{x_i\}\right)^2} + \frac{\sum_i \exp\{x_i\} h_i^2}{\sum_i \exp\{x_i\}}$$

$$\frac{d^2}{dt^2}\Big|_{t=0} f(x+th) = h^T f''(x) h = -\left(\frac{\sum_i \exp\{x_i\} h_i}{\sum_i \exp\{x_i\}}\right)^2 + \frac{\sum_i \exp\{x_i\} h_i^2}{\sum_i \exp\{x_i\}}$$

Setting $p_i = \frac{\exp\{x_i\}}{\sum_j \exp\{x_j\}}$, we have

$$\begin{aligned}
h^T f''(x) h &= \sum_i p_i h_i^2 - \left(\sum_i p_i h_i\right)^2 \\
&= \sum_i p_i h_i^2 - \left(\sum_i \sqrt{p_i}(\sqrt{p_i} h_i)\right)^2 \\
&\geq \sum_i p_i h_i^2 - \left(\sum_i (\sqrt{p_i})^2\right)\left(\sum_i (\sqrt{p_i} h_i)^2\right) \\
&= \sum_i p_i h_i^2 - \left(\sum_i p_i h_i^2\right) = 0
\end{aligned}$$

(note that $\sum_i p_i = 1$)

Note: For many years I thought that $\ln(\sum_i \exp\{x_i\})$ is one of very small family of multi-dimensional functions for which convexity is established "by bare hands" – by checking positive semidefiniteness of the Hessian. Recently I realized that convexity of this function can be established by "Convexity Calculus" with no computations:

$$s > 0 \Rightarrow \ln(s) = \min_z [s\exp\{z\} - z - 1] \text{ [straightforward computation]}$$

$$\Rightarrow \ln(\sum_i \exp\{x_i\}) = \min_z \left[\underbrace{\sum_i \exp\{z\}\exp\{x_i\} - z - 1}_{\text{convex function of } [x;z]}\right]$$

and it remains to use the rule on preserving convexity by partial minimization.

Corollary: *When $c_i > 0$, the function*

$$g(y) = \ln\left(\sum_i c_i \exp\{a_i^T y\}\right)$$

is convex.

Indeed,

$$g(y) = \ln\left(\sum_i \exp\{\ln c_i + a_i^T y\}\right)$$

is obtained from the convex function

$$\ln\left(\sum_i \exp\{x_i\}\right)$$

by affine substitution of argument.

Gradient Inequality

Proposition: *Let f be a function, x be a point of the domain* Dom$f$ *of f and Q, $x \in Q$, be a convex set such that f is convex on Q. Assume that f is differentiable at x: there exists a vector $f'(x)$ such that*

$$\forall \epsilon > 0 \exists \delta > 0 : y \in \mathrm{Dom} f \ \& \ \|y - x\| \leq \delta \Rightarrow |f(y) - f(x) - (y - x)^T f'(x)| \leq \epsilon \|y - x\|.$$

Then

$$\forall y \in Q : f(y) \geq f(x) + (y - x)^T f'(x). \qquad (*)$$

Proof. Let $y \in Q$. There is nothing to prove when $y = x$ or $f(y) = +\infty$, thus, assume that $f(y) < \infty$ and $y \neq x$. Let is set $z_\epsilon = x + \epsilon(y - x)$, $0 < \epsilon < 1$. Then z_ϵ is an interior point of the segment $[x, y]$. Since f is convex, we have

$$\frac{f(y) - f(x)}{\|y - x\|} \geq \frac{f(z_\epsilon) - f(x)}{\|z_\epsilon - x\|} = \underbrace{\frac{f(x + \epsilon(y - x)) - f(x)}{\epsilon}}_{\to (y-x)^T f'(x) \text{ as } \epsilon \to +0} \cdot \frac{1}{\|y - x\|}$$

Passing to limit as $\epsilon \to +0$, we arrive at

$$\frac{f(y) - f(x)}{\|y - x\|} \geq \frac{(y - x)^T f'(x)}{\|y - x\|},$$

as required by $(*)$.

Lipschitz continuity of a convex function

Proposition: *Let f be a convex function, and let K be a closed and bounded set belonging to relative interior of the domain of f. Then f is Lipschitz continuous on K, that is, there exists a constant $L < \infty$ such that*

$$|f(x) - f(y)| \leq L\|x - y\|_2 \quad \forall x, y \in K.$$

Note: All three assumptions on K are essential, as is shown by the following examples:

$\diamondsuit$ $f(x) = -\sqrt{x}$, $\mathrm{Dom} f = \{x \geq 0\}$, $K = [0, 1]$. Here $K \subset \mathrm{Dom} f$ is closed and bounded, but is not contained in the relative interior of $\mathrm{Dom} f$, and f is *not* Lipschitz continuous on K

$\diamondsuit$ $f(x) = x^2$, $\mathrm{Dom} f = K = \mathbb{R}$. Here K is closed and belongs to $\mathrm{rint}\, \mathrm{Dom} f$, but is unbounded, and f is *not* Lipschitz continuous on K

$\diamondsuit$ $f(x) = \frac{1}{x}$, $\mathrm{Dom} f = \{x > 0\}$, $K = (0, 1]$. Here K is bounded and belongs to $\mathrm{rint}\, Dom f$, but is not closed, and f is *not* Lipschitz continuous on K

Maxima and Minima of Convex Functions

(!) **Proposition** ["unimodality"] *Let f be a convex function and x_* be a local minimizer of f:*

$$x_* \in \mathrm{Dom} f \ \& \ \exists r > 0 : f(x) \geq f(x_*) \ \forall (x : \|x - x_*\| \leq r).$$

Then x_* is a global minimizer of f:

$$f(x) \geq f(x_*) \ \forall x.$$

Proof: All we need to prove is that if $x \neq x_*$ and $x \in \mathrm{Dom} f$, then $f(x) \geq f(x_*)$. To this end let $z \in (x_*, x)$. By convexity we have

$$\frac{f(z) - f(x_*)}{\|z - x_*\|} \leq \frac{f(x) - f(x_*)}{\|x - x_*\|}.$$

When $z \in (x_*, x)$ is close enough to x_*, we have $\frac{f(z)-f(x_*)}{\|z-x_*\|} \geq 0$, whence $\frac{f(x)-f(x_*)}{\|x-x_*\|} \geq 0$, that is, $f(x) \geq f(x_*)$.

Proposition *Let f be a convex function. The set of X_* of global minimizers is convex.*

Proof: This is an immediate corollary of important

Lemma: *Let f be a convex function. Then the level sets of f, that is, the sets*

$$X_a = \{x : f(x) \leq a\}$$

where a is a real, are convex.

Proof of Lemma: If $x, y \in X_a$ and $\lambda \in [0, 1]$, then

$$\begin{aligned} f(\lambda x + (1-\lambda)y) &\leq \lambda f(x) + (1-\lambda) f(y) \\ &\leq \lambda a + (1-\lambda) a = a. \end{aligned}$$

Thus, $[x, y] \subset X_a$.

♣ *When the minimizer of a convex function is unique?*

Definition: A convex function is called *strictly convex*, if

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y)$$

whenever $x \neq y$ and $\lambda \in (0,1)$.

Note: *If a convex function f has open domain and is twice continuously differentiable on this domain with*

$$h^T f''(x) h > 0 \quad \forall (x \in \mathrm{Dom} f, h \neq 0),$$

then f is strictly convex.

Proposition: *For a strictly convex function f a minimizer, if it exists, is unique.*

Proof. Assume that $X_* = \mathrm{Argmin}\, f$ contains two distinct points x', x''. By strong convexity,

$$f(\frac{1}{2}x' + \frac{1}{2}x'') < \frac{1}{2}\left[f(x') + f(x'')\right] = \inf_x f,$$

which is impossible.

Theorem [Optimality conditions in convex minimization] *Let f be a function which is differentiable at a point x_* and is convex on a convex set $Q \subset \mathrm{Dom} f$ which contains x_*. A necessary and sufficient condition for f to attain its minimum on Q at x_* is*

$$(x - x_*)^T f'(x_*) \geq 0 \quad \forall x \in Q. \qquad (*)$$

Proof, $\Leftarrow$: Assume that $(*)$ is valid. Applying Gradient inequality, for $x \in Q$ we have

$$f(x) \geq f(x_*) + (x - x_*)^T f'(x_*),$$

and $(x - x_*)^T f'(x_*) \geq 0$ by $(*)$, implying that $f(x) \geq f(x_*)$ whenever $x \in Q$.

"Let f be a function which is differentiable at a point x_* and is convex on a convex set $Q \subset \mathrm{Dom} f$ which contains x_*. A necessary and sufficient condition for f to attain its minimum on Q at x_* is

$$(x - x_*)^T f'(x_*) \geq 0 \quad \forall x \in Q."$$

Proof, $\Rightarrow$: Given that $x_* \in \mathrm{Argmin}_{y \in Q} f(y)$, let $x \in Q$. Then

$$0 \leq \frac{f(x_* + \lambda[x - x_*]) - f(x_*)}{\lambda} \quad \forall \lambda \in (0, 1),$$

whence $(x - x_*)^T f'(x_*) \geq 0$.

♣ **Equivalent reformulation:** *Let f be a function which is differentiable at a point x_* and is convex on a convex set $Q \subset \mathrm{Dom} f$, $x_* \in Q$. Consider the radial cone of Q at x_*:*

$$T_Q(x_*) = \{h : \exists t > 0 : x_* + th \in Q\}$$

Note: *$T_Q(x_*)$ is indeed a cone which is comprised of all vectors of the form $s(x - x_*)$, where $x \in Q$ and $s \geq 0$.*
f attains its minimum on Q at x_* iff

$$h^T f'(x_*) \geq 0 \ \forall h \in T_Q(x_*),$$

or, which is the same, iff

$$f'(x_*) \in \underbrace{N_Q(x_*) = \{g : g^T h \geq 0 \forall h \in T_Q(x_*)\}}_{\text{normal cone of } Q \text{ at } x_*}. \qquad (*)$$

Example I: $x_* \in \mathrm{int}\, Q$. Here $T_Q(x_*) = \mathbb{R}^n$, whence $N_Q(x_*) = \{0\}$, and $(*)$ becomes the Fermat equation

$$f'(x_*) = 0$$

Example II: $x_* \in \operatorname{rint} Q$**.** Let $\mathrm{Aff}(Q) = x_* + L$, where L is a linear subspace in $\mathbb{R}^n$. Here $T_Q(x_*) = L$, whence $N_Q(x_*) = L^\perp$. $(*)$ becomes the condition

$$f'(x_*) \text{ is orthogonal to } L.$$

Equivalently: Let $\mathrm{Aff}(Q) = \{x : Ax = b\}$. Then $L = \{x : Ax = 0\}$, $L^\perp = \{y = A^T\lambda\}$, and the optimality condition becomes

$$\exists \lambda^* : \quad \begin{array}{c} \nabla\Big|_{x=x_*}[f(x) + (\lambda^*)^T(Ax - b)] = 0 \\ \Updownarrow \\ f'(x_*) + \sum_i \lambda_i^* \nabla(a_i^T x - b_i) = 0 \end{array} \qquad [A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix}]$$

Example III: $Q = \{x : Ax - b \leq 0\}$ is polyhedral. Here

$$T_Q(x_*) = \left\{h : a_i^T h \leq 0 \ \forall i \in I(x_*) = \{i : a_i^T x_* - b_i = 0\}\right\}.$$

By Homogeneous Farkas Lemma,

$$\begin{array}{rcl} N_Q(x_*) & \equiv & \{y : a_i^T h \leq 0,\ i \in I(x_*) \Rightarrow y^T h \geq 0\} \\ & = & \{y = -\sum\limits_{i \in I(x_*)} \lambda_i a_i : \lambda_i \geq 0\} \end{array}$$

and the optimality condition becomes

$$\exists(\lambda_i^* \geq 0, i \in I(x_*)) : f'(x_*) + \sum_{i \in I(x_*)} \lambda_i^* a_i = 0$$

or, which is the same:

$$\exists \lambda^* \geq 0 : \left\{ \begin{array}{rcl} f'(x_*) + \sum\limits_{i=1}^m \lambda_i^* a_i & = & 0 \\ \lambda_i^*(a_i^T x_* - b_i) & = & 0,\ i = 1, ..., m \end{array} \right.$$

The point is that in the *convex* case these conditions are necessary *and sufficient* for x_* to be a minimizer of f on Q.

Example: Let us solve the problem

$$\min_x \left\{ c^T x + \sum_{i=1}^m x_i \ln x_i : x \geq 0, \sum_i x_i = 1 \right\}.$$

The objective is convex, the domain $Q = \{x \geq 0, \sum_i x_i = 1\}$ is convex (and even polyhedral). *Assuming* that the minimum is achieved at a point $x_* \in \operatorname{rint} Q$, the optimality condition becomes

$$\nabla \left[c^T x + \sum_i x_i \ln x_i + \lambda [\sum_i x_i - 1] \right] = 0$$

$$\Updownarrow$$

$$\ln x_i = -c_i - \lambda - 1 \ \forall i$$

$$\Updownarrow$$

$$x_i = \exp\{1 - \lambda\} \exp\{-c_i\}$$

Since $\sum_i x_i$ should be 1, we arrive at

$$x_i = \frac{\exp\{-c_i\}}{\sum_j \exp\{-c_j\}}.$$

At this point, the optimality condition is satisfied, so that the point indeed is a minimizer.

Maxima of convex functions

Proposition. *Let f be a convex function. Then*

$\diamondsuit$ *If f attains its maximum over* $\mathrm{Dom} f$ *at a point $x^* \in \mathrm{rint}\,\mathrm{Dom} f$, then f is constant on* $\mathrm{Dom} f$

Indeed, assuming that $f(x) < f(x^*)$ for some $x \in \mathrm{Dom} f$, $y = x^* + \alpha[x^* - x] \in \mathrm{Dom} f$ for small $\alpha > 0 \Rightarrow x^*$ is in the relative interior of segment $[x, y] \subset \mathrm{Dom} f$

$\Rightarrow f(x^*) \leq \lambda \underbrace{f(x)}_{<f(x^*)} + (1-\lambda) f(y)$ for some $\lambda \in (0,1) \Rightarrow f(y) > f(x^*)$ – contradiction!

$\diamondsuit$ *If* $\mathrm{Dom} f$ *is closed and does not contain lines and f attains its maximum on* $\mathrm{Dom} f$*, then among the maximizers there is an extreme point of* $\mathrm{Dom} f$

$\diamondsuit$ *If* $\mathrm{Dom} f$ *is polyhedral and f is bounded from above on* $\mathrm{Dom} f$*, then f attains its maximum on* $\mathrm{Dom} f$*.*

• **Good news:** Maximizing convex function f over a bounded polyhedral set $X \neq \emptyset$ reduces to computing the function at finitely many extreme points of the set. For example, problem $\max_x \{f(x) : \|x\|_1 \leq 1\}$ is easy

• **Bad news:** For a bounded polyhedral X, the number of extreme points usually is astronomically large, as is the case for the box $X = \{x : \|x\|_\infty \leq 1\}$, making maximizing over extreme points by looking at them one by one intractable. *In general, maximizing convex function is a computationally intractable task.*

Subgradients of convex functions

♣ Let f be a convex function and $\bar{x} \in \operatorname{int} \operatorname{Dom} f$. If f differentiable at $\bar{x}$, then, by Gradient Inequality, there exists an affine function, specifically,

$$h(x) = f(\bar{x}) + [\nabla f(\bar{x})]^T (x - \bar{x}),$$

which underestimates f everywhere and coincides with f at $\bar{x}$:

$$f(x) \geq h(x) \forall x \ \& \ f(\bar{x}) = h(\bar{x}) \qquad (*)$$

Affine function with property $(*)$ may exist also in the case when f is *not* differentiable at $\bar{x} \in \operatorname{Dom} f$. $(*)$ implies that

$$h(x) = f(\bar{x}) + g^T (x - \bar{x}) \qquad (**)$$

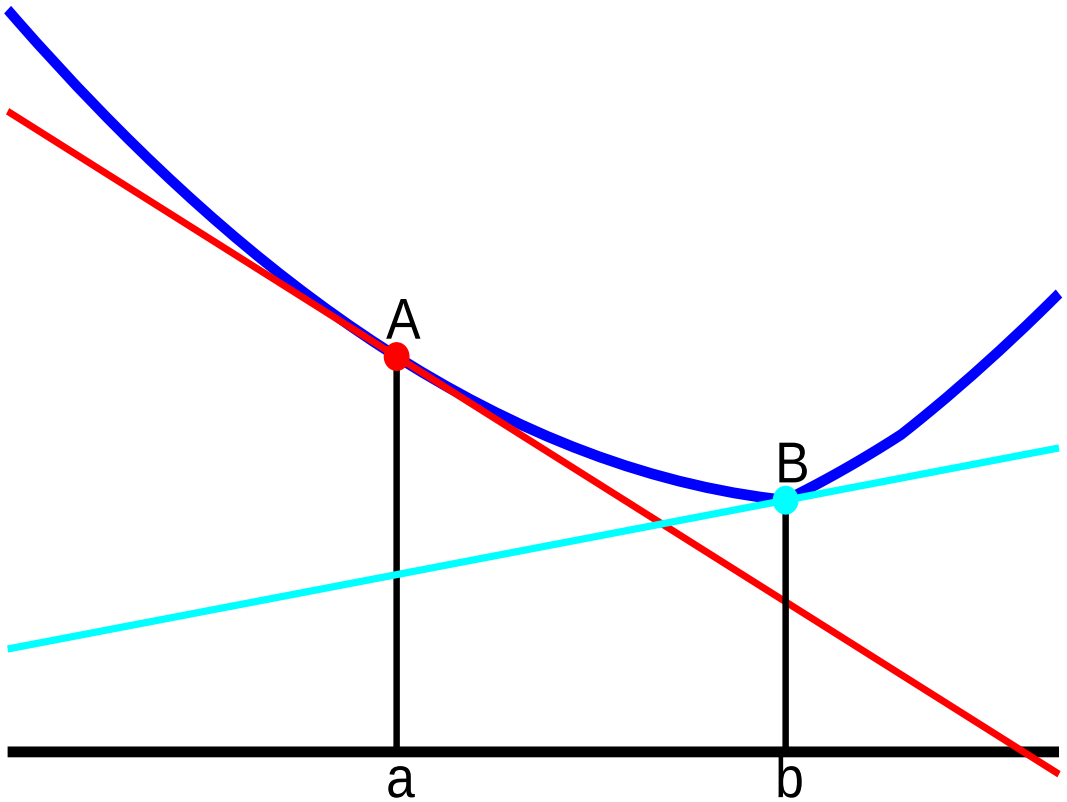
for certain g. Function $(**)$ indeed satisfies $(*)$ if and only if g is such that

$$f(x) \geq f(\bar{x}) + g^T (x - \bar{x}) \quad \forall x \qquad (!)$$

<u>Definition.</u> Let f be a convex function and $\bar{x} \in \mathrm{Dom} f$. Every vector g satisfying

$$f(x) \geq f(\bar{x}) + g^T(x - \bar{x}) \quad \forall x \tag{!}$$

is called a *subgradient* of f at $\bar{x}$. The set of all subgradients, if any, of f at $\bar{x}$ is called *subdifferential* $\partial f(\bar{x})$ of f at $\bar{x}$.



Geometrically: A hyperplane supporting the epigraph Epi$\{f\}$ of f at a point $(\bar{x}, f(\bar{x}))$ is, at least for $\bar{x} \in \operatorname{int} \operatorname{Dom} f$, the graph of an affine function $h(x) = f(\bar{x}) + g^T(x - \bar{x})$ which underestimates f everywhere and is equal to f at the point $x = \bar{x}$.

The slope g of this affine function is a subgradient of f at x.

Definition. Let f be a convex function and $\bar{x} \in \mathrm{Dom} f$. Every vector g satisfying

$$f(x) \geq f(\bar{x}) + (x - \bar{x})^T g \quad \forall x \tag{!}$$

is called a *subgradient* of f at $\bar{x}$. The set of all subgradients, if any, of f at $\bar{x}$ is called *subdifferential* $\partial f(\bar{x})$ of f at $\bar{x}$.

Example I: By Gradient Inequality, if convex function f is differentiable at $\bar{x}$, then $\nabla f(\bar{x}) \in \partial f(\bar{x})$. If, in addition, $\bar{x} \in \mathrm{int}\,\mathrm{Dom} f$, then $\nabla f(\bar{x})$ is the *unique* element of $\partial f(\bar{x})$.

Example II: Let $f(x) = |x|$ $(x \in \mathbb{R})$. When $\bar{x} \neq 0$, f is differentiable at $\bar{x}$, whence $\partial f(\bar{x}) = f'(\bar{x})$. When $\bar{x} = 0$, subgradients g are given by

$$|x| \geq 0 + gx = gx \ \forall x,$$

that is, $\partial f(0) = [-1, 1]$.

Note: In the case in question, f has directional derivative

$$Df(x)[h] = \lim_{t \to +0} \frac{f(x + th) - f(x)}{t}$$

at every point $x \in \mathbb{R}$ along every direction $h \in \mathbb{R}$, and this derivative is nothing but

$$Df(x)[h] = \max_{g \in \partial f(x)} g^T h$$

Proposition: *Let f be convex, $\mathrm{Dom} f$ be nonempty, and let $L = \mathrm{Aff}(\mathrm{Dom} f) - \mathrm{Aff}(\mathrm{Dom} f)$ be the linear subspace parallel to $\mathrm{Aff}(\mathrm{Dom} f)$. Then*

◇ *For every $x \in \mathrm{Dom} f$, the subdifferential $\partial f(x)$ is a closed convex set*

◇ *If $x \in \mathrm{rint}\,\mathrm{Dom} f$, then $\partial f(x)$ is nonempty.*

◇ *If $x \in \mathrm{rint}\,\mathrm{Dom} f$, then, for every $h \in L$,*

$$\exists Df(x)[h] \equiv \lim_{t \to +0} \frac{f(x+th) - f(x)}{t} = \max_{g \in \partial f(x)} g^T h.$$

◇ *Assume that $\bar{x} \in \mathrm{Dom} f$ is represented as $\lim_{i \to \infty} x_i$ with $x_i \in \mathrm{Dom} f$ and that*

$$f(\bar{x}) \leq \lim\inf_{i \to \infty} f(x_i)$$

If a sequence $g_i \in \partial f(x_i)$ converges to certain vector g, then $g \in \partial f(\bar{x})$.

◇ *The multi-valued mapping $x \mapsto \partial f(x)$ is locally bounded at every point $\bar{x} \in \mathrm{int}\,\mathrm{Dom} f$, that is, whenever $\bar{x} \in \mathrm{int}\,\mathrm{Dom} f$, there exist $r > 0$ and $R < \infty$ such that*

$$\|x - \bar{x}\|_2 \leq r, g \in \partial f(x) \Rightarrow \|g\|_2 \leq R.$$

Selected proof: "If $\bar{x} \in \operatorname{rint} \operatorname{Dom} f$, then $\partial f(\bar{x})$ is nonempty."
W.l.o.g. let $\operatorname{Dom} f$ be full-dimensional, so that $\bar{x} \in \operatorname{int} \operatorname{Dom} f$. Consider the convex set

$$T = \operatorname{Epi}\{f\} = \{(x,t) : t \geq f(x)\}.$$

Since f is convex, it is continuous on $\operatorname{int} \operatorname{Dom} f$, whence T has a nonempty interior. The point $(\bar{x}, f(\bar{x}))$ clearly does not belong to this interior, whence $S = \{(\bar{x}, f(\bar{x}))\}$ can be separated from T: there exists $(\alpha, \beta) \neq 0$ such that

$$\alpha^T \bar{x} + \beta f(\bar{x}) \leq \alpha^T x + \beta t \quad \forall (x, t \geq f(x)) \qquad (*)$$

Clearly $\beta \geq 0$ (otherwise $(*)$ will be impossible when $x = \bar{x}$ and $t > f(\bar{x})$ is large).
<u>Claim:</u> $\beta > 0$. Indeed, with $\beta = 0$, (*) implies

$$\alpha^T \bar{x} \leq \alpha^T x \ \forall x \in \operatorname{Dom} f \qquad (**)$$

Since $(\alpha, \beta) \neq 0$ and $\beta = 0$, we have $\alpha \neq 0$; but then $(**)$ contradicts $\bar{x} \in \operatorname{int} \operatorname{Dom} f$.
$\diamondsuit$ Since $\beta > 0$, $(*)$ implies that if $g = \beta^{-1}\alpha$, then

$$g^T \bar{x} + f(\bar{x}) \leq g^T x + f(x) \ \forall x \in \operatorname{Dom} f,$$

that is,

$$f(x) \geq f(\bar{x}) + (x - \bar{x})^T g \ \forall x.$$

Elementary Calculus of Subgradients

◇ If $g_i \in \partial f_i(x)$ and $\lambda_i \geq 0$, then

$$\sum_i \lambda_i g_i \in \partial(\sum_i \lambda_i f_i)(x)$$

◇ If $g_\alpha \in \partial f_\alpha(x)$, $\alpha \in \mathcal{A}$,

$$f(\cdot) = \sup_{\alpha \in \mathcal{A}} f_\alpha(\cdot)$$

and

$$f(x) = f_\alpha(x), \ \alpha \in \mathcal{A}_*(x) \neq \emptyset,$$

then every convex combination of vectors g_α, $\alpha \in \mathcal{A}_*(x)$, is a subgradient of f at x

◇ If $g_i \in \partial f_i(x)$, $i = 1, ..., m$, and $F(y_1, ..., y_m)$ is convex and monotone and $0 \leq d \in \partial F(f_1(x), ..., f_m(x))$, then the vector

$$\sum_i d_i g_i$$

is a subgradient of $F(f_1(\cdot), ..., f_m(\cdot))$ at x.

Lecture 6:

Convex Programming

Lagrange Duality

Saddle Points

Convex Programming
Lagrange Duality
Saddle Points

♣ **Mathematical Programming program** is

$$f_* = \min_x \left\{ f(x) : \begin{array}{rl} g(x) \equiv (g_1(x), ..., g_m(x))^T & \leq 0 \\ h(x) = (h_1(x), ..., h_k(x))^T & = 0 \\ x & \in X \end{array} \right\} \qquad (P)$$

◇ x is the *design vector*. Values of x are called *solutions* to (P)

◇ $f(x)$ is the *objective*

◇ $g(x) \equiv (g_1(x), ..., g_m(x))^T \leq 0$ – *inequality constraints*

◇ $h(x) = (h_1(x), ..., h_k(x))^T = 0$ – *equality constraints*

◇ $X \subset \mathbb{R}^n$ – *domain*. We always assume that the objective and the constraints are well-defined on X.

$$f_* = \min_x \left\{ f(x) : \begin{array}{rl} g(x) \equiv (g_1(x), ..., g_m(x))^T & \leq 0 \\ h(x) = (h_1(x), ..., h_k(x))^T & = 0 \\ x & \in X \end{array} \right\} \qquad (P)$$

♣ Solution x is called *feasible*, if it satisfies all the constraints. Problem which has feasible solutions is called *feasible*.

♣ If the objective is (below) bounded on the set of feasible solutions, (P) is called *bounded*.

♣ The *optimal value* f_* is

$$f_* = \begin{cases} \inf_x \{f(x) : \ x \text{ is feasible}\}, & (P) \text{ is feasible} \\ +\infty, & \text{otherwise} \end{cases}$$

f_* is a real for feasible and bounded problem, is $-\infty$ for feasible unbounded problem, and is $+\infty$ for infeasible problem.

♣ *Optimal solution* of (P) is a feasible solution x_* such that $f(x_*) = f_*$. Problem which has optimal solutions is called *solvable*.

$$f_* = \min_x \left\{ f(x) : \begin{array}{rl} g(x) \equiv (g_1(x), ..., g_m(x))^T & \leq 0 \\ h(x) = (h_1(x), ..., h_k(x))^T & = 0 \\ x & \in X \end{array} \right\} \qquad (P)$$

♣ Problem (P) is called *convex*, if

♢ X is a convex subset of $\mathbb{R}^n$

♢ $f(\cdot)$, $g_1(\cdot)$,...,$g_m(\cdot)$ are *convex real-valued* functions on X

♢ There are no equality constraints

[we could allow *linear* equality constraints, but this does not add generality]

Preparing tools for Lagrange Duality: Convex Theorem on Alternative

♣ **Question:** *How to certify* **insolvability** *of the system*

$$\begin{array}{rcl} f(x) & < & c \\ g_j(x) & \leq & 0,\ j = 1, ..., m \\ x & \in & X \end{array} \qquad (I)$$

♣ **Answer:** *Assume that there exist* **nonnegative** *weights* λ_j, $j = 1, ..., m$, *such that the inequality*

$$f(x) + \sum_{j=1}^{m} \lambda_j g_j(x) < c$$

has no solutions in X*:*

$$\exists \lambda_j \geq 0 : \quad \inf_{x \in X} [f(x) + \sum_{j=1}^{m} \lambda_j g_j(x)] \geq c.$$

Then (I) *is insolvable.*

♣ **<u>Convex Theorem on Alternative:</u>** *Consider a system of constraints on* x

$$\begin{array}{rcl} f(x) & < & c \\ g_j(x) & \leq & 0,\ j = 1, ..., m \\ x & \in & X \end{array} \qquad (I)$$

along with system of constraints on λ*:*

$$\begin{array}{rcl} \inf\limits_{x \in X} [f(x) + \sum\limits_{j=1}^{m} \lambda_j g_j(x)] & \geq & c \\ \lambda_j & \geq & 0,\ j = 1, ..., m \end{array} \qquad (II)$$

◇ *[Trivial part]* **<u>If</u>** *(II) is solvable,* **<u>then</u>** *(I) is insolvable*

◇ *[Nontrivial part]* **<u>If</u>** *(I) is insolvable* **<u>and</u>** *system (I) is convex:*

— X *is convex set*

— f, $g_1, ..., g_m$ *are real-valued convex functions on* X

<u>and</u> *the subsystem*

$$\begin{array}{rcl} g_j(x) & < & 0,\ j = 1, ..., m, \\ x & \in & X \end{array}$$

is solvable [Slater condition], **<u>then</u>** (II) *is solvable.*

$\diamondsuit$ [Nontrivial part] **<u>If</u>** (I) is insolvable **<u>and</u>** system (I) is convex:

— X is convex set

— f, $g_1, ..., g_m$ are real-valued convex functions on X

<u>and</u> the subsystem

$$\begin{array}{rcl} g_j(x) & < & 0,\ j = 1, ..., m, \\ x & \in & X \end{array}$$

is solvable [Slater condition], **<u>then</u>** the system of constraints on λ

$$\begin{array}{rcl} \inf\limits_{x \in X} [f(x) + \sum\limits_{j=1}^{m} \lambda_j g_j(x)] & \geq & c \\ \lambda_j & \geq & 0,\ j = 1, ..., m \end{array} \qquad (II)$$

is solvable.

Fact: Nontrivial part remains valid when Slater condition is replaced with **Relaxed Slater Condition:** *There exists* $\bar{x} \in \operatorname{rint} X$ *such that* $g_i(\bar{x}) \leq 0$ *for all* i *<u>and</u>* $g_i(\bar{x}) < 0$ *for those* i *for which* $g_i(\cdot)$ *are not affine functions.*

$$\begin{array}{rcl} f(x) & < & c \\ g_j(x) & \leq & 0,\ j = 1, ..., m \\ x & \in & X \end{array} \qquad (I)$$

Proof of Nontrivial part (under Slater condition): Assume that (I) has no solutions. Consider two sets in $\mathbb{R}^{m+1}$:

$$\overbrace{\left\{ u \in \mathbb{R}^{m+1} : \exists x \in X : \begin{array}{rcl} f(x) & \leq & u_0 \\ g_1(x) & \leq & u_1 \\ \multicolumn{3}{c}{\cdots\cdots\cdots\cdots} \\ g_m(x) & \leq & u_m \end{array} \right\}}^{T}$$

$$\underbrace{\left\{ u \in \mathbb{R}^{m+1} : u_0 < c, u_1 \leq 0, ..., u_m \leq 0 \right\}}_{S}$$

<u>Observations:</u> ♢ S, T are convex (since X, f, and g_i are so) and nonempty

♢ S, T do not intersect (otherwise (I) would have a solution)

<u>Conclusion:</u> S and T can be separated:

$$\exists (a_0, ..., a_m) \neq 0 : \inf_{u \in T} a^T u \geq \sup_{u \in S} a^T u$$

$$\overbrace{\left\{ u \in \mathbb{R}^{m+1} : \exists x \in X : \begin{array}{rcl} f(x) & \leq & u_0 \\ g_1(x) & \leq & u_1 \\ \multicolumn{3}{c}{\dots\dots\dots\dots} \\ g_m(x) & \leq & u_m \end{array} \right\}}^{T}$$

$$\underbrace{\left\{ u \in \mathbb{R}^{m+1} : u_0 < c, u_1 \leq 0, \dots, u_m \leq 0 \right\}}_{S}$$

$$\begin{aligned} &\exists (a_0, \dots, a_m) \neq 0 : \\ &\quad \inf_{x \in X} \inf_{\substack{u_0 \geq f(x) \\ u_i \geq g_i(x),\ i \leq m}} [a_0 u_0 + a_1 u_1 + \dots + a_m u_m] \\ &\geq \sup_{u_0 < c,\ u_i \leq 0,\ i \leq m} [a_0 u_0 + a_1 u_1 + \dots + a_m u_m] \end{aligned}$$

<u>Conclusion:</u> $a \geq 0$, whence

$$\inf_{x \in X} [a_0 f(x) + a_1 g_1(x) + \dots + a_m g_m(x)] \geq a_0 c.$$

Summary:

$$\exists a \geq 0, a \neq 0 :$$
$$\inf_{x \in X} [a_0 f(x) + a_1 g_1(x) + ... + a_m g_m(x)] \geq a_0 c$$

Observation: $a_0 > 0$.

Indeed, otherwise $0 \neq (a_1, ..., a_m) \geq 0$ and

$$\inf_{x \in X} [a_1 g_1(x) + ... + a_m g_m(x)] \geq 0,$$

while $\exists \bar{x} \in X : g_j(\bar{x}) < 0$ for all j.

Conclusion: $a_0 > 0$, whence

$$\inf_{x \in X} \left[f(x) + \sum_{j=1}^{m} \underbrace{\left[\frac{a_j}{a_0} \right]}_{\lambda_j \geq 0} g_j(x) \right] \geq c.$$

Lagrange Function

♣ Consider optimization program

$$\mathrm{Opt}(P) = \min\left\{f(x) : g_j(x) \leq 0,\ j \leq m,\ x \in X\right\}. \qquad (P)$$

and associate with it *Lagrange function*

$$L(x,\lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x)$$

along with the *Lagrange Dual problem*

$$\mathrm{Opt}(D) = \max_{\lambda \geq 0} \underline{\mathbf{L}}(\lambda),\ \underline{\mathbf{L}}(\lambda) = \inf_{x \in X} L(x,\lambda) \qquad (D)$$

♣ **Convex Programming Duality Theorem:**

♢ *[Weak Duality] For every $\lambda \geq 0$, $\underline{\mathbf{L}}(\lambda) \leq \mathrm{Opt}(P)$. In particular,*

$$\mathrm{Opt}(D) \leq \mathrm{Opt}(P)$$

♢ *[Strong Duality] If (P) is convex and below bounded and satisfies Relaxed Slater condition, then (D) is solvable, and*

$$\mathrm{Opt}(D) = \mathrm{Opt}(P).$$

$$\mathrm{Opt}(P) = \min\{f(x) : g_j(x) \le 0,\ j \le m,\ x \in X\} \quad (P)$$

$$\Downarrow$$

$$L(x,\lambda) = f(x) + \sum_j \lambda_j g_j(x)$$

$$\Downarrow$$

$$\mathrm{Opt}(D) = \max_{\lambda \ge 0} \underbrace{\left[\inf_{x \in X} L(x,\lambda)\right]}_{\underline{\mathbf{L}}(\lambda)} \qquad (D)$$

Weak Duality: "$\mathrm{Opt}(D) \le \mathrm{Opt}(P)$": There is nothing to prove when (P) is infeasible, that is, when $\mathrm{Opt}(P) = \infty$. If x is feasible for (P) and $\lambda \ge 0$, then $L(x,\lambda) \le f(x)$, whence

$$\begin{aligned}
\lambda \ge 0 \Rightarrow \underline{\mathbf{L}}(\lambda) &\equiv \inf_{x \in X} L(x,\lambda) \\
&\le \inf_{x \in X \text{ is feasible}} L(x,\lambda) \\
&\le \inf_{x \in X \text{ is feasible}} f(x) \\
&= \mathrm{Opt}(P) \\
\Rightarrow \mathrm{Opt}(D) &= \sup_{\lambda \ge 0} \underline{\mathbf{L}}(\lambda) \le \mathrm{Opt}(P).
\end{aligned}$$

$$\mathrm{Opt}(P) = \min\{f(x) : g_j(x) \le 0,\ j \le m,\ x \in X\} \quad (P)$$
$$\Rightarrow L(x,\lambda) = f(x) + \sum_j \lambda_j g_j(x)$$
$$\Rightarrow \mathrm{Opt}(D) = \max_{\lambda \ge 0} \underbrace{\left[\inf_{x \in X} L(x,\lambda)\right]}_{\underline{\mathbf{L}}(\lambda)} \qquad (D)$$

Strong Duality: "If (P) is convex and below bounded and satisfies Relaxed Slater condition, then (D) is solvable and $\mathrm{Opt}(D) = \mathrm{Opt}(P)$":
The system

$$f(x) < \mathrm{Opt}(P),\ g_j(x) \le 0,\ j = 1, ..., m,\ x \in X$$

has no solutions. Since the Relaxed Slater condition holds true, we can apply CTA. By CTA,

$$\exists \lambda^* \ge 0 : f(x) + \sum_j \lambda_j^* g_j(x) \ge \mathrm{Opt}(P)\ \forall x \in X,$$

whence

$$\underline{\mathbf{L}}(\lambda^*) \ge \mathrm{Opt}(P). \qquad (*)$$

Combined with Weak Duality, $(*)$ says that

$$\mathrm{Opt}(D) = \underline{\mathbf{L}}(\lambda^*) = \mathrm{Opt}(P).$$

$$\mathrm{Opt}(P) = \min\{f(x) : g_j(x) \le 0,\ j \le m,\ x \in X\} \quad (P)$$

$$\Downarrow$$

$$L(x,\lambda) = f(x) + \sum_j \lambda_j g_j(x)$$

$$\Downarrow$$

$$\mathrm{Opt}(D) = \max_{\lambda \ge 0} \underbrace{\left[\inf_{x \in X} L(x,\lambda)\right]}_{\underline{\mathbf{L}}(\lambda)} \qquad (D)$$

<u>Note:</u> The Lagrange function "remembers", up to equivalence, both (P) and (D).
Indeed,

$$\mathrm{Opt}(D) = \sup_{\lambda \ge 0} \inf_{x \in X} L(x,\lambda)$$

is given by the Lagrange function. Now consider the function

$$\overline{L}(x) = \sup_{\lambda \ge 0} L(x,\lambda) = \begin{cases} f(x), & g_j(x) \le 0,\ j \le m \\ +\infty, & \text{otherwise} \end{cases}$$

(P) clearly is equivalent to the problem of minimizing $\overline{L}(x)$ over $x \in X$:

$$\mathrm{Opt}(P) = \inf_{x \in X} \sup_{\lambda \ge 0} L(x,\lambda)$$

Saddle Points

♣ Let $X \subset \mathbb{R}^n$, $\Lambda \subset \mathbb{R}^m$ be nonempty sets, and let $F(x, \lambda)$ be a real-valued function on $X \times \Lambda$. This function gives rise to two optimization problems

$$\begin{aligned} \mathrm{Opt}(P) &= \inf_{x \in X} \overbrace{\sup_{\lambda \in \Lambda} F(x, \lambda)}^{\overline{F}(x)} \quad (P) \\ \mathrm{Opt}(D) &= \sup_{\lambda \in \Lambda} \underbrace{\inf_{x \in X} F(x, \lambda)}_{\underline{F}(\lambda)} \quad (D) \end{aligned}$$

$$\mathrm{Opt}(P) = \inf_{x\in X} \overbrace{\sup_{\lambda\in\Lambda} F(x,\lambda)}^{\overline{F}(x)} \quad (P)$$
$$\mathrm{Opt}(D) = \sup_{\lambda\in\Lambda} \underbrace{\inf_{x\in X} F(x,\lambda)}_{\underline{\mathbf{F}}(\lambda)} \quad (D)$$

Game interpretation: Player I chooses $x \in X$, player II chooses $\lambda \in \Lambda$. With choices of the players x, λ, player I pays to player II the sum of $F(x,\lambda)$. What should the players do to optimize their wealth?

$\diamondsuit$ If Player I chooses x first, and Player II knows this choice when choosing λ, II will maximize his profit, and the loss of I will be $\overline{F}(x)$. To minimize his loss, I should solve (P), thus ensuring himself loss $\mathrm{Opt}(P)$ or less.

$\diamondsuit$ If Player II chooses λ first, and Player I knows this choice when choosing x, I will minimize his loss, and the profit of II will be $\underline{\mathbf{F}}(\lambda)$. To maximize his profit, II should solve (D), thus ensuring himself profit $\mathrm{Opt}(D)$ or more.

$$\mathrm{Opt}(P) = \inf_{x\in X} \overbrace{\sup_{\lambda\in\Lambda} F(x,\lambda)}^{\overline{F}(x)} \quad (P)$$
$$\mathrm{Opt}(D) = \sup_{\lambda\in\Lambda} \underbrace{\inf_{x\in X} F(x,\lambda)}_{\underline{\mathbf{F}}(\lambda)} \quad (D)$$

Observation: For Player I, second situation seems better, so that it is natural to guess that his anticipated loss in this situation is $\leq$ his anticipated loss in the first situation:

$$\mathrm{Opt}(D) \equiv \sup_{\lambda\in\Lambda} \inf_{x\in X} F(x,\lambda) \leq \inf_{x\in X} \sup_{\lambda\in\Lambda} F(x,\lambda) \equiv \mathrm{Opt}(P).$$

This indeed is true: assuming $\mathrm{Opt}(P) < \infty$ (otherwise the inequality is evident),

$$\begin{aligned}
&\forall(\epsilon > 0): \quad \exists x_\epsilon \in X : \sup_{\lambda\in\Lambda} F(x_\epsilon,\lambda) \leq \mathrm{Opt}(P) + \epsilon \\
&\Rightarrow \forall \lambda \in \Lambda : \underline{\mathbf{F}}(\lambda) = \inf_{x\in X} F(x,\lambda) \leq F(x_\epsilon,\lambda) \leq \mathrm{Opt}(P) + \epsilon \\
&\Rightarrow \mathrm{Opt}(D) \equiv \sup_{\lambda\in\Lambda} \underline{\mathbf{F}}(\lambda) \leq \mathrm{Opt}(P) + \epsilon \\
&\Rightarrow \mathrm{Opt}(D) \leq \mathrm{Opt}(P).
\end{aligned}$$

$$\mathrm{Opt}(P) = \inf_{x\in X} \overbrace{\sup_{\lambda\in\Lambda} F(x,\lambda)}^{\overline{F}(x)} \quad (P)$$
$$\mathrm{Opt}(D) = \sup_{\lambda\in\Lambda} \underbrace{\inf_{x\in X} F(x,\lambda)}_{\underline{\mathbf{F}}(\lambda)} \quad (D)$$

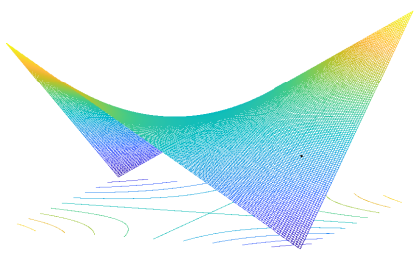
♣ What should the players do when making their choices simultaneously?

A "good case" when we can answer this question – F has a *saddle point*.

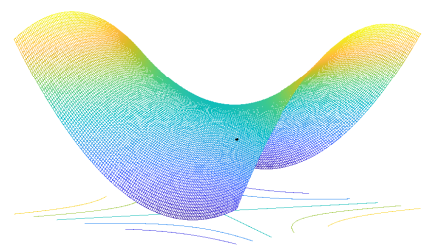
Definition: *We call a point* $(x_*, \lambda_*) \in X \times \Lambda$ *a saddle point of* F*, if*

$$F(x,\lambda_*) \geq F(x_*,\lambda_*) \geq F(x_*,\lambda) \ \forall (x \in X, \lambda \in \Lambda).$$

In game terms, a saddle point is an *equilibrium* – no one of the players can improve his wealth, provided the adversary keeps his choice unchanged.



$F(x,\lambda) = -x\lambda$



$F(x,\lambda) = x^2 - \lambda^2 + x\lambda$

In both cases, $F(x,0) \geq F(0,0) \geq F(0,\lambda) \Rightarrow (0,0)$ is a saddle point

$$\mathrm{Opt}(P) = \inf_{x\in X} \overbrace{\sup_{\lambda\in\Lambda} F(x,\lambda)}^{\overline{F}(x)} \quad (P)$$
$$\mathrm{Opt}(D) = \sup_{\lambda\in\Lambda} \underbrace{\inf_{x\in X} F(x,\lambda)}_{\underline{\mathbf{F}}(\lambda)} \quad (D)$$

Definition: *We call a point* $(x_*, \lambda_*) \in X \times \Lambda$ *a saddle point of* $F(x,\lambda) : X \times \Lambda \to \mathbb{R}$, *if*

$$F(x,\lambda_*) \geq F(x_*,\lambda_*) \geq F(x_*,\lambda) \ \forall (x \in X, \lambda \in \Lambda).$$

Proposition [Existence and Structure of saddle points]: *F has a saddle point if and only if both (P) and (D) are solvable with equal optimal values. In this case, the saddle points of F are exactly the pairs (x_*, λ_*), where x_* is an optimal solution to (P), and λ_* is an optimal solution to (D).*

At every saddle point, (x_*, λ_*), $F(x_*, \lambda_*)$ equals to the common value of $\mathrm{Opt}(P)$ *and* $\mathrm{Opt}(D)$.

$$\mathrm{Opt}(P) = \inf_{x\in X} \overbrace{\sup_{\lambda\in\Lambda} F(x,\lambda)}^{\overline{F}(x)} \quad (P)$$
$$\mathrm{Opt}(D) = \sup_{\lambda\in\Lambda} \underbrace{\inf_{x\in X} F(x,\lambda)}_{\underline{\mathbf{F}}(\lambda)} \quad (D)$$

Proof, $\Rightarrow$: Assume that (x_*, λ_*) is a saddle point of F, and let us prove that x_* solves (P), λ_* solves (D), and $\mathrm{Opt}(P) = \mathrm{Opt}(D)$.
Indeed, we have

$$F(x,\lambda_*) \geq F(x_*,\lambda_*) \geq F(x_*,\lambda) \ \forall (x\in X, \lambda\in\Lambda)$$

whence

$$\mathrm{Opt}(P) \leq \overline{F}(x_*) = \sup_{\lambda\in\Lambda} F(x_*,\lambda) = F(x_*,\lambda_*)$$
$$\mathrm{Opt}(D) \geq \underline{\mathbf{F}}(\lambda_*) = \inf_{x\in X} F(x,\lambda_*) = F(x_*,\lambda_*)$$

Since $\mathrm{Opt}(P) \geq \mathrm{Opt}(D)$, we see that all inequalities in the chain

$$\mathrm{Opt}(P) \leq \overline{F}(x_*) = F(x_*,\lambda_*) = \underline{\mathbf{F}}(\lambda_*) \leq \mathrm{Opt}(D)$$

are equalities. Thus, x_* solves (P), λ_* solves (D) and $\mathrm{Opt}(P) = \mathrm{Opt}(D)$.

$$\mathrm{Opt}(P) = \inf_{x\in X} \overbrace{\sup_{\lambda\in\Lambda} F(x,\lambda)}^{\overline{F}(x)} \quad (P)$$
$$\mathrm{Opt}(D) = \sup_{\lambda\in\Lambda} \underbrace{\inf_{x\in X} F(x,\lambda)}_{\underline{\mathbf{F}}(\lambda)} \quad (D)$$

Proof, $\Leftarrow$. Assume that (P), (D) have optimal solutions x_*, λ_* and $\mathrm{Opt}(P) = \mathrm{Opt}(D)$, and let us prove that (x_*, λ_*) is a saddle point. We have

$$\begin{aligned} \mathrm{Opt}(P) &= \overline{F}(x_*) = \sup_{\lambda\in\Lambda} F(x_*,\lambda) \geq F(x_*,\lambda_*) \\ \mathrm{Opt}(D) &= \underline{\mathbf{F}}(\lambda_*) = \inf_{x\in X} F(x,\lambda_*) \leq F(x_*,\lambda_*) \end{aligned} \qquad (*)$$

Since $\mathrm{Opt}(P) = \mathrm{Opt}(D)$, all inequalities in $(*)$ are equalities, so that

$$\sup_{\lambda\in\Lambda} F(x_*,\lambda) = F(x_*,\lambda_*) = \inf_{x\in X} F(x,\lambda_*).$$

$$\mathrm{Opt}(P) = \min_x \left\{ f(x) : g_j(x) \leq 0, j \leq m, x \in X \right\} \quad (P)$$
$$\Rightarrow L(x, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x)$$

Theorem [Saddle Point form of Optimality Conditions in Convex Programming]

Let $x_* \in X$.

$\diamondsuit$ *[Sufficient optimality condition]* **If** *x_* can be extended, by a $\lambda^* \geq 0$, to a saddle point of the Lagrange function on $X \times \{\lambda \geq 0\}$:*

$$L(x, \lambda^*) \geq L(x_*, \lambda^*) \geq L(x_*, \lambda) \ \forall (x \in X, \lambda \geq 0),$$

then *x_* is optimal for (P).*

$\diamondsuit$ *[Necessary optimality condition]* **If** *x_* is optimal for (P)* **and** *(P) is convex and satisfies the Relaxed Slater condition,* **then** *x_* can be extended, by a $\lambda^* \geq 0$, to a saddle point of the Lagrange function on $X \times \{\lambda \geq 0\}$.*

$$\mathrm{Opt}(P) = \min_x \left\{ f(x) : g_j(x) \le 0, j \le m, x \in X \right\} \quad (P)$$
$$\Rightarrow L(x,\lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x)$$

Proof, $\Rightarrow$: "**<u>Assume</u>** $x_* \in X$ and $\exists \lambda^* \ge 0$:

$$L(x,\lambda^*) \ge L(x_*,\lambda^*) \ge L(x_*,\lambda) \ \forall (x \in X, \lambda \ge 0). \qquad (*)$$

<u>Then</u> x_* is optimal for (P)."

Clearly, $\sup_{\lambda \ge 0} L(x,\lambda) = \begin{cases} +\infty, & x \in X \text{ is infeasible} \\ f(x), & \text{otherwise} \end{cases}$

We have $\lambda^* \ge 0$ & $+\infty > L(x_*,\lambda^*) \ge L(x_*,\lambda) \ \forall \lambda \ge 0$ whence

$$g_j(x_*) \le 0 \forall j \quad (\text{otherwise } \sup_{\lambda \ge 0} L(x_*,\lambda) = \infty)$$
$$\& \ L(x_*,\lambda^*) = f(x_*) + \sum_j \lambda_j^* g_j(x_*) = \max_{\lambda \ge 0} L(x_*,\lambda) = f(x_*) + \max_{\lambda \ge 0} \sum_j \lambda_j g_j(x_*),$$

$\Rightarrow \lambda_j^* g_j(x_*) = 0 \, \forall j$ and $L(x_*,\lambda^*) = f(x_*)$

$\Rightarrow$ the left inequality in $(*)$ reads

$$L(x,\lambda^*) \ge f(x_*) \ \forall x \in X. \qquad (!)$$

Since $\lambda^* \ge 0$, one has $f(x) \ge L(x,\lambda^*)$ for all feasible x, and therefore (!) implies that

$$x \text{ is feasible } \Rightarrow f(x) \ge f(x_*).$$

$$\mathrm{Opt}(P) = \min_x \left\{ f(x) : g_j(x) \le 0, j \le m, x \in X \right\} \quad (P)$$
$$\Rightarrow L(x,\lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x)$$

Proof, $\Leftarrow$: **Assume** x_* is optimal for convex problem (P) satisfying the Relaxed Slater condition. Then $\exists \lambda^* \ge 0$:

$$L(x,\lambda^*) \ge L(x_*,\lambda^*) \ge L(x_*,\lambda) \ \forall (x \in X, \lambda \ge 0).$$

As we have already seen, the primal and the dual problems stemming from the Lagrange function are

$$\begin{array}{rcll} \mathrm{Opt}(P_{\mathrm{Lag}}) & = & \min_{x\in X} \left[\overline{\mathbf{L}}(x) = \left\{ \begin{array}{ll} f(x), & x \text{ is feasible} \\ +\infty, & \text{otherwise} \end{array} \right. \right] & (P_{\mathrm{Lag}}) \\ \mathrm{Opt}(D) & = & \max_{\lambda \ge 0} \underline{\mathbf{L}}(\lambda) & (D) \end{array}$$

By Lagrange Duality Theorem, in the case under consideration the dual problem has an optimal solution λ^* and $\mathrm{Opt}(D) = \mathrm{Opt}(P_{\mathrm{Lag}})$. By the origin of x_*, x_* is an optimal solution to (P_{Lag}). Consequently, (x_*, λ^*) is a saddle point of the Lagrange function by Proposition on Existence and Structure of saddle points.

$$\mathrm{Opt}(P) = \min_x \left\{ f(x) : g_j(x) \le 0, j \le m, x \in X \right\} \quad (P)$$

$$\Downarrow$$

$$L(x, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x)$$

Theorem [Karush-Kuhn-Tucker Optimality Conditions in Convex Programming] *Let (P) be a convex program, let x^* be its feasible solution, and let the functions f, g_1,...,g_m be differentiable at x^*. Then*

$\diamondsuit$ *The Karush-Kuhn-Tucker condition:*

Exist Lagrange multipliers $\lambda^* \ge 0$ such that

$$\nabla f(x_*) + \sum_{j=1}^{m} \lambda_j^* \nabla g_j(x_*) \in N_X^*(x_*) := \{g : g^T(x - x_*) \ge 0 \, \forall x \in X\}$$

$$\lambda_j^* g_j(x_*) = 0, \; j \le m \quad \text{[complementary slackness]}$$

is sufficient for x_* to be optimal.

$\diamondsuit$ *If (P) satisfies Relaxed Slater condition:*

$\exists \bar{x} \in \mathrm{rint}\, X : g_j(\bar{x}) \le 0$ *for all constraints and* $g_j(\bar{x}) < 0$ *for all nonlinear constraints,*

then the KKT is necessary and sufficient for x_* to be optimal.

$$\mathrm{Opt}(P) = \min_x \left\{ f(x) : g_j(x) \le 0, j \le m, x \in X \right\} \quad (P)$$

$$\Downarrow$$

$$L(x, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x)$$

Proof, $\Rightarrow$: Let (P) be convex, x_* be feasible, and f, g_j be differentiable at x_*. Assume also that the KKT holds:
Exist Lagrange multipliers $\lambda^* \ge 0$ such that

$$(a) \quad \nabla f(x_*) + \sum_{j=1}^{m} \lambda_j^* \nabla g_j(x_*) \in N_X^*(x_*)$$

$$(b) \quad \lambda_j^* g_j(x_*) = 0, \; j \le m \;\; \text{[complementary slackness]}$$

Then x_* is optimal.
Indeed, complementary slackness plus $\lambda^* \ge 0$ ensure that

$$L(x_*, \lambda^*) \ge L(x_*, \lambda) \quad \forall \lambda \ge 0.$$

Further, $L(x, \lambda^*)$ is convex in $x \in X$ and differentiable at $x_* \in X$, so that (a) implies that

$$L(x, \lambda^*) \ge L(x_*, \lambda^*) \quad \forall x \in X.$$

Thus, x_* can be extended to a saddle point of the Lagrange function and therefore is optimal for (P).

$$\mathrm{Opt}(P) = \min_x \left\{ f(x) : g_j(x) \le 0, j \le m, x \in X \right\} \quad (P)$$
$$\Downarrow$$
$$L(x,\lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x)$$

Proof, $\Leftarrow$ Let (P) be convex and satisfy the Relaxed Slater condition, let x_* be optimal and f, g_j be differentiable at x_*. Then
Exist Lagrange multipliers $\lambda^* \ge 0$ such that

$$(a) \quad \nabla f(x_*) + \sum_{j=1}^{m} \lambda_j^* \nabla g_j(x_*) \in N_X^*(x_*)$$
$$(b) \quad \lambda_j^* g_j(x_*) = 0,\ j \le m \ \text{[complementary slackness]}$$

By Saddle Point Optimality condition, from optimality of x_* it follows that $\exists \lambda^* \ge 0$ such that (x_*, λ^*) is a saddle point of $L(x,\lambda)$ on $X \times \{\lambda \ge 0\}$. This is equivalent to

$$\lambda_j^* g_j(x_*) = 0 \ \forall j \ \& \ \underbrace{\min_{x \in X} L(x,\lambda^*) = L(x_*,\lambda^*)}_{(*)}$$

Since the function $L(x,\lambda^*)$ is convex in $x \in X$ and differentiable at $x_* \in X$, relation $(*)$ implies (a).

♣ **Application example:** Assuming $a_i > 0$, $p \geq 1$, let us solve the problem

$$\min_x \left\{ \sum_i \frac{a_i}{x_i} : x > 0, \sum_i x_i^p \leq 1 \right\}$$

Assuming $x_* > 0$ is a solution such that $\sum_i (x_i^*)^p = 1$, the KKT conditions read

$$\begin{array}{rcl} \nabla_x \left\{ \sum_i \frac{a_i}{x_i} + \lambda(\sum_i x_i^p - 1) \right\} & = & 0 \Leftrightarrow \frac{a_i}{x_i^2} = p\lambda x_i^{p-1} \\ \sum_i x_i^p & = & 1 \end{array}$$

whence $x_i = c(\lambda) a_i^{\frac{1}{p+1}}$. Since $\sum_i x_i^p$ should be 1, we get

$$x_i^* = \frac{a_i^{\frac{1}{p+1}}}{\left(\sum_j a_j^{\frac{p}{p+1}} \right)^{\frac{1}{p}}}.$$

This point is feasible, problem is convex, KKT at the point is satisfied $\Rightarrow x^*$ is optimal!

Existence of Saddle Points

♣ **Theorem** [Sion-Kakutani] *Let $X \subset \mathbb{R}^n$, $\Lambda \subset \mathbb{R}^m$ be nonempty convex closed sets and $F(x, \lambda) : X \times \Lambda \to \mathbb{R}$ be a continuous function which is convex in $x \in X$ and concave in $\lambda \in \Lambda$.*
Assume that X is compact, and that there exists $\bar{x} \in X$ such that for every $a \in \mathbb{R}$ the set

$$\Lambda_a : \{\lambda \in \Lambda : F(\bar{x}, \lambda) \geq a\}$$

is bounded (e.g., Λ is bounded). Then F possesses a saddle point on $X \times \Lambda$.

♠ The key role in the proof of Sion-Kakutani Theorem is played by

<u>MiniMax Lemma:</u> *Let $f_i(x)$, $i = 1, ..., m$, be convex continuous functions on a convex compact set $X \subset \mathbb{R}^n$. Then there exists $\mu^* \geq 0$ with $\sum\limits_i \mu_i^* = 1$ such that*

$$\min_{x \in X} \max_{1 \leq i \leq m} f_i(x) = \min_{x \in X} \sum_i \mu_i^* f_i(x)$$

<u>Note:</u> Setting $\Delta = \{\mu \in \mathbb{R}^m : \mu \geq 0, \sum_i \mu_i = 1\}$, consider the convex-concave saddle point problem

$$\min_{x \in X} \max_{\mu \in \Delta} \sum_i \mu_i f_i(x) \Rightarrow \begin{cases} \mathrm{Opt}(P) & = & \min_{x \in X} \overline{F}(x) := \underbrace{\max\limits_{\mu \in \Delta} \sum\limits_i \mu_i f_i(x)}_{\equiv \max_i f_i(x)} & (P) \\ \mathrm{Opt}(D) & = & \max_{\mu \in \Delta} \underline{F}(\mu) := \min_{x \in X} \sum_i \mu_i f_)i(x) & (D) \end{cases}$$

MinMax Lemma states that $\mathrm{Opt}(D) = \mathrm{Opt}(P)$, or (since (P) and (D) under the premise of MinMax lemma clearly are solvable) that *the convex-concave function $\sum_i \mu_i f_i(x)$ has a saddle point on $X \times \Delta$.*

$\Rightarrow$ *MinMax Lemma is a special case of Sion-Kakutani Theorem. After this special case is proved, the general result follows easily.*

Proof of MinMax Lemma: Consider the optimization program

$$\begin{gathered}\min_{t,x}\left\{t : f_i(x) - t \le 0,\ i \le m, (t,x) \in X_+\right\}, \\ X_+ = \{(t,x) : x \in X\}\end{gathered} \tag{P}$$

The optimal value in this problem clearly is

$$t_* = \min_{x\in X} \max_i f_i(x).$$

The program clearly is convex, solvable and satisfies the Slater condition, whence there exists $\lambda^* \ge 0$ and an optimal solution (x_*, t_*) to (P) such that $(x_*, t_*; \lambda^*)$ is the saddle point of the Lagrange function on $X^+ \times \{\lambda \ge 0\}$:

$$\begin{array}{rcll} \min\limits_{x\in X,t}\left\{t + \sum\limits_i \lambda_i^*(f_i(x) - t)\right\} & = & t_* + \sum\limits_i \lambda_i^*(f_i(x_*) - t_*) & (a) \\ \max\limits_{\lambda\ge 0}\left\{t_* + \sum\limits_i \lambda_i(f_i(x_*) - t_*)\right\} & = & t_* + \sum\limits_i \lambda_i^*(f_i(x_*) - t_*) & (b) \end{array}$$

(b) implies that $t_* + \sum\limits_i \lambda_i^*(f_i(x_*) - t_*) = t_*$.

(a) implies that $\sum\limits_i \lambda_i^* = 1$. Thus, $\lambda^* \ge 0, \sum_i \lambda_i^* = 1$ and

$$\begin{aligned} \min_{x\in X} \sum_i \lambda_i^* f_i(x) &= \min_{x\in X,t}\left\{t + \sum_i \lambda_i^*(f_i(x) - t)\right\} \\ &= t_* + \sum_i \lambda_i^*(f_i(x_*) - t_*) = t_* = \min_{x\in X} \max_i f_i(x). \end{aligned}$$

Proof of Sion-Kakutani Theorem: We should prove that problems

$$\begin{array}{rcll} \mathrm{Opt}(P) & = & \inf\limits_{x\in X} \overbrace{\sup\limits_{\lambda\in\Lambda} F(x,\lambda)}^{\overline{F}(x)} & (P) \\ \mathrm{Opt}(D) & = & \sup\limits_{\lambda\in\Lambda} \underbrace{\inf\limits_{x\in X} F(x,\lambda)}_{\underline{F}(\lambda)} & (D) \end{array}$$

are solvable with equal optimal values.

1^0. Since X is compact and $F(x,\lambda)$ is continuous on $X\times\Lambda$, the function $\underline{F}(\lambda)$ is continuous on Λ. Besides this, the sets

$$\Lambda^a = \{\lambda\in\Lambda : \underline{F}(\lambda)\geq a\}$$

are contained in the sets

$$\Lambda_a = \{\lambda\in\Lambda : F(\overline{x},\lambda)\geq a\}$$

and therefore are bounded. Finally, Λ is closed, so that the *continuous* function $\underline{F}(\cdot)$ with *bounded* level sets Λ^a attains it maximum on a *closed* set Λ. Thus, (D) is solvable.

2^0. Consider the sets

$$X(\lambda) = \{x \in X : F(x, \lambda) \leq \mathrm{Opt}(D)\}.$$

These are closed convex subsets of a compact set X. Let us prove that every finite collection of these sets has a nonempty intersection. Indeed, assume that $X(\lambda^1) \cap ... \cap X(\lambda^N) = \emptyset$, so that

$$\max_{j=1,...,N} F(x, \lambda^j) > \mathrm{Opt}(D) \ \forall x \in X$$

$$\Rightarrow \min_{x \in X} \max_j F(x, \lambda^j) > \mathrm{Opt}(D)$$

by compactness of X and continuity of F.

By MinMax Lemma, there exist weights $\mu_j \geq 0, \sum\limits_j \mu_j = 1$, such that

$$\min_{x \in X} \underbrace{\sum\nolimits_j \mu_j F(x, \lambda^j)}_{\substack{\leq F(x, \sum_j \mu_j \lambda_j) \\ \text{since } F \text{ is concave in } \lambda}} > \mathrm{Opt}(D),$$

that is,

$$\underline{F}(\sum\nolimits_j \mu_j \lambda_j) := \min_{x \in X} F(x, \sum\nolimits_j \mu_j \lambda_j) \geq \min_{x \in X} \sum_j \mu_j F(x, \lambda_j) > \mathrm{Opt}(D),$$

which is impossible.

3^0. Since every finite collection of closed convex subsets $X(\lambda)$ of the compact set X has a nonempty intersection, all those sets have a nonempty intersection:

$$\exists x_* \in X : F(x_*, \lambda) \leq \mathrm{Opt}(D) \ \forall \lambda.$$

Due to $\mathrm{Opt}(P) \geq \mathrm{Opt}(D)$, this is possible iff x_* is optimal for (P) and $\mathrm{Opt}(P) = \mathrm{Opt}(D)$.

Extension: Conic Form of Convex Program/Lagrange Duality/Optimality Conditions

♠ Traditionally, when passing from Linear Programming problem

$$\min_x \{c^T x : [g_1^T x - b_1; g_2^T x - b_2; ...; g_m^T x - b_m] \leq [0; 0; ...; 0]\}$$

to a nonlinear convex problem, one replaces linear objective $c^T x$ and linear left hand sides $g_i^T x - b_i$ of the constraints with nonlinear convex functions.

♠ There exists less straightforward and in many aspects essentially better suited for convex optimization way to introduce nonlineariy – *to make "nonlinear" the inequality* $\leq$

Vector Inequalities

♣ Let $\mathbf{K} \subset \mathbb{R}^\nu$ be a *regular cone* – closed convex pointed ($\mathbf{K} \cap [-\mathbf{K}] = \{0\}$) cone with nonempty interior in $\mathbb{R}^\nu$.

♠ **$\mathbf{K}$-ordering:** $\mathbf{K}$ defines *ordering* on $E = \mathbb{R}^\nu$: we say that $a \in E$ is $\mathbf{K}$-*greater than or equal* to $b \in E$ (synonym: b is $\mathbf{K}$-less than or equal to a, notation: $a \geq_{\mathbf{K}} b \Leftrightarrow b \leq_{\mathbf{K}} a$) when $a - b \in \mathbf{K}$:

$$\{a \geq_{\mathbf{K}} b \Leftrightarrow b \leq_{\mathbf{K}} a \Leftrightarrow a - b \geq_{\mathbf{K}} 0 \Leftrightarrow b - a \leq_{\mathbf{K}} 0\} \text{ means that } a - b \in \mathbf{K}$$

Example: coordinate-wise $\geq$. When $\mathbf{K}$ is the nonnegative orthant $\mathbb{R}^n_+$ in $E = \mathbb{R}^n$, $\geq_{\mathbf{K}}$ (denoted just $\geq$) is the standard coordinate-wise vector inequality:

$$[a_1; ...; a_n] \geq [b_1; ...; b_n] \Leftrightarrow a_i \geq b_i, i \leq n.$$

Note: When $n \geq 2$, the ordering $\leq_K$ on $\mathbb{R}^n$, same as coordinate-wise $\leq$, is *partial* – some pairs a, b are comparable, that is, either $a \leq_{\mathbf{K}} b$, or $b \leq_{\mathbf{K}} a$ holds true, and some pairs a, b are *in*comparable – neither $a \leq_{\mathbf{K}} b$, nor $b \leq_{\mathbf{K}} a$ holds true.

$\mathbf{K}$: regular(closed convex pointed with nonempty interior.

$\{a \geq_{\mathbf{K}} b \Leftrightarrow b \leq_{\mathbf{K}} a \Leftrightarrow a - b \geq_{\mathbf{K}} 0 \Leftrightarrow b - a \leq_{\mathbf{K}} 0\}$ means that $a - b \in \mathbf{K}$

$\mathbf{K}$-ordering

♡ has all standard properties of partial ordering:

— is *reflexive*: $a \geq_{\mathbf{K}} a \, \forall a \in E$

— is *antisymmetric*: $a \geq_{\mathbf{K}} b$ and $b \geq_{\mathbf{K}} a$ iff $a = b$

— is *transitive*: if $a \geq_{\mathbf{K}} b$ and $b \geq_{\mathbf{K}} c$, then $a \geq_{\mathbf{K}} c$

♡ is compatible with linear operations on E:

— we can sum up valid $\mathbf{K}$-inequalities of the same sign: if $a \geq_{\mathbf{K}} b$ and $c \geq_{\mathbf{K}} d$, then $a + c \geq_{\mathbf{K}} b + d$

— we can multiply both sides of valid $\mathbf{K}$-inequality by a nonnegative real: if $a \geq_{\mathbf{K}} b$ and $\lambda \geq 0$, then $\lambda a \geq_{\mathbf{K}} \lambda b$

$\mathbf{K}$: closed convex pointed ($\mathbf{K} \cap [-\mathbf{K}] = \{0\}$) cone with nonempty interior in $E = \mathbb{R}^\nu$,
$a \geq_{\mathbf{K}} b \Leftrightarrow b \leq_{\mathbf{K}} a \Leftrightarrow a - b \in \mathbf{K} \Leftrightarrow a - b \geq_{\mathbf{K}} 0 \Leftrightarrow b - a \leq_{\mathbf{K}} 0$

♡ Closedness of $\mathbf{K}$ allows to pass to limits in $\geq_{\mathbf{K}}$-inequalities: if $a_i \geq_{\mathbf{K}} b_i$ for all i and $a_i \to a$, $b_i \to b$ as $i \to \infty$, then $a \geq_{\mathbf{K}} b$

♡ Nonemptiness of $\operatorname{int} \mathbf{K}$ allows to define *strict* $\mathbf{K}$-inequality:

$$\{a >_{\mathbf{K}} b \Leftrightarrow b <_{\mathbf{K}} a \Leftrightarrow a - b >_{\mathbf{K}} 0 \Leftrightarrow b - a <_{\mathbf{K}} 0\} \text{ means that } a - b \in \operatorname{int} \mathbf{K}$$

In contrast to $\geq_{\mathbf{K}}$, the relation $>_{\mathbf{K}}$ is stable: the strict inequality $a >_{\mathbf{K}} b$ is preserved by small enough perturbations in a and b.

♡ Arithmetics of strict and nonstrict inequalities is exactly the same as in the case of arithmetic $\geq$ and $>$; say, the sum of valid $\geq$ and $>$ inequalities is a valid $>$ inequality,

Example: The interior of $\mathbb{R}^n_+$ is the set of vectors with positive coordinates $\Rightarrow$ Strict version of the coordinate-wise vector inequality is

$$[a_1; ...; a_n] > [b_1; ...; b_n] \Leftrightarrow a_i > b_i, i \leq n.$$

Example: The *semidefinite cone* $\mathbf{S}^n_+$ – the set of symmetric positive semidefinite matrices in the space $\mathbf{S}^n$ of $n \times n$ symmetric matrices – defines the semidefinite ordering $\geq_{\mathbf{S}^n_+}$ (denoted just $\succeq$) on $\mathbf{S}^n$:

$$\{A \succeq B \Leftrightarrow B \preceq A \Leftrightarrow A - B \succeq 0 \Leftrightarrow B - A \preceq 0\} \text{ means that } x^T A x - x^T B x \geq 0 \, \forall x \in \mathbb{R}^n$$
$$\{A \succ B \Leftrightarrow B \prec A \Leftrightarrow A - B \succ 0 \Leftrightarrow B - A \prec 0\} \text{ means that } x^T A x - x^T B x > 0 \, \forall x \in \mathbb{R}^n \setminus \{0\}$$

♠ **Dual cone.** With a cone $\mathbf{K} \subset \mathbb{R}^n$ one associates its *dual cone* $\mathbf{K}_*$*:*

$$\mathbf{K}_* = \{\lambda \in \mathbb{R}^n : \lambda^T x \geq 0 \, \forall x \in \mathbf{K}\}$$

Fact: The cone $\mathbf{K}_*$ dual to a regular cone $\mathbf{K}$ is regular, and the cone dual to the dual is the original cone:

$$[\mathbf{K}_*]_* = \mathbf{K}.$$

Fact: *Multiplying both sides of valid* $\mathbf{K}$*-inequality* $a \leq_{\mathbf{K}} b$ *by* $\mathbf{K}_*$*-nonnegative* λ*, we get valid scalar inequality:*

$$a \leq_{\mathbf{K}} b \;\&\; \lambda \geq_{\mathbf{K}_*} 0 \;\Rightarrow \lambda^T a \leq \lambda^T b.$$

Example: The cone dual to the nonnegative orthant $\mathbb{R}^n_+$ is the same nonnegative orthant $\mathbb{R}^n_+$.

♠ **$\mathbf{K}$-convexity:** Let $\mathbf{K}$ be a regular cone in $E = \mathbb{R}^\nu$, X be a convex set in $\mathbb{R}^n$, and $f(x) : X \to E$ be a mapping. f is called $\mathbf{K}$-convex on X if

$$\forall (x, y \in X, \lambda \in [0,1]) : f(\lambda x + (1-\lambda)y) \leq_{\mathbf{K}} \lambda f(x) + (1-\lambda) f(y),$$

or, which is the same, the $\mathbf{K}$-epigraph $\{[x; y] : x \in X, y \geq_{\mathbf{K}} f(x)\}$ of f is a convex set.

Examples: • The usual – scalar – convex function f in X is exactly the $\mathbf{K}$-convex one, with $\mathbf{K} = \mathbb{R}_+ \subset E = \mathbb{R}$

• $\mathbb{R}^k_+$-convex function on X is a vector-valued function on X with k convex scalar components

• The function $f(x) = xx^T : \mathbb{R}^{m \times n} \to \mathbf{S}^m$ is $\mathbf{S}^m_+$-convex:

$$\forall (x, y \in \mathbb{R}^{m \times n}, \lambda \in [0,1]) : [\lambda x + (1-\lambda)y][\lambda x + (1-\lambda)y]^T \preceq \lambda xx^T + (1-\lambda)yy^T$$

due to

$$\lambda xx^T + (1-\lambda)yy^T - [\lambda x + (1-\lambda)y][\lambda x + (1-\lambda)y]^T = \lambda(1-\lambda)[x-y][x-y]^T \succeq 0.$$

♠ **Convex optimization problem in conic form** is

$$\mathrm{Opt}(P) = \min_{x \in X} \{f(x) : \overline{g}(x) = Ax - b \leq 0, \widehat{g}(x) \leq_{\mathbf{K}} 0\}, \qquad (P)$$

where

- $X \subset \mathbb{R}^n$ is a nonempty convex set
- $f(x)$ is a convex real-valued function on X
- $\mathbf{K}$ is a regular cone in $E = \mathbb{R}^\nu$
- $\overline{g}(x) = Ax - b : \mathbb{R}^n \to \mathbb{R}^k$ is affine
- $\widehat{g}(x) : X \to E$ is $\mathbf{K}$-convex mapping

Note: Convex problem in Mathematical Programming form

$$\min_{x \in X} \{f(x) : g_i(x) \leq 0, i \leq m\}$$

with convex nonempty X and convex and real-valued on X functions $f, g_1, ..., g_m$ is a convex optimization problem in conic form, with empty set of polyhedral constraints $\overline{g}(x) = Ax - b \leq 0$, $\widehat{g}(x) = [g_1(x); ...; g_m(x)]$ and $\mathbf{K} = \mathbb{R}^m_+$,

Note: *A system* of J convex conic constraints

$$\widehat{g}_j(x) \leq_{\mathbf{K}_j} 0,\, j \leq J$$

is equivalent to a *single* convex conic constraint

$$\widehat{g}(x) := [\widehat{g}_1(x); , ...; \widehat{g}_J(x)] \leq_{\mathbf{K}} 0$$

on a larger cone

$$\mathbf{K} = \mathbf{K}_1 \times ... \times \mathbf{K}_J := \left\{ [y_1; ...; y_J] : y_j \in \mathbf{K}_j,\, j \leq J \right\}$$

Convex optimization problem in conic form is

$$\mathrm{Opt}(P) = \min_{x \in X} \{f(x) : \overline{g}(x) = Ax - b \leq 0, \widehat{g}(x) \leq_{\mathbf{K}} 0\}, \qquad (P)$$

where

- $X \subset \mathbb{R}^n$ is a nonempty convex set
- $f(x)$ is a convex real-valued function on X
- $\mathbf{K}$ is a regular cone in $E = \mathbb{R}^\nu$
- $\overline{g}(x) = Ax - b : \mathbb{R}^n \to \mathbb{R}^k$ is affine
- $\widehat{g}(x) : X \to E$ is $\mathbf{K}$-convex mapping
- **Conic Lagrange function** of (P) is the function

$$L(x; \overline{\lambda}, \widehat{\lambda}) = f(x) + \overline{\lambda}^T \overline{g}(x) + \widehat{\lambda}^T \widehat{g}(x)$$

We *always* restrict the *Lagrange multipliers* $[\overline{\lambda}; \widehat{\lambda}]$ to reside in the domain $\mathcal{LM} = \mathbb{R}^k_+ \times \mathbf{K}_*$ where

$$\mathbf{K}_* = \{\widehat{\lambda} \in E = \mathbb{R}^\nu : \widehat{\lambda}^T z \geq 0 \, \forall z \in \mathbf{K}\}$$

is the cone dual to $\mathbf{K}$.

Convex optimization problem in conic form is

$$\mathrm{Opt}(P) = \min_{x \in X} \{f(x) : \overline{g}(x) = Ax - b \leq 0, \widehat{g}(x) \leq_{\mathbf{K}} 0\}, \qquad (P)$$

Associated Conic Lagrange function is

$$L(x; \overline{\lambda}, \widehat{\lambda}) = f(x) + \overline{\lambda}^T \overline{g}(x) + \widehat{\lambda}^T \widehat{g}(x) : X \times \mathcal{LM} \to \mathbb{R}, \ \mathcal{LM} = \{[\overline{\lambda}; \widehat{\lambda}] : \overline{\lambda} \geq 0, \widehat{\lambda} \geq_{\mathbf{K}_*} 0\}.$$

♠ **Conic Lagrange dual to** (P) is the problem

$$\mathrm{Opt}(D) = \max_{[\overline{\lambda}; \widehat{\lambda}] \in \mathcal{LM}} \left\{ \underline{L}(\overline{\lambda}, \widehat{\lambda}) := \inf_{x \in X} L(x; \overline{\lambda}, \widehat{\lambda}) \right\} \qquad (D)$$

Note: when $[\overline{\lambda}; \widehat{\lambda}] \in \mathcal{LM}$ and x is feasible for (P), we have

$$L(x; \overline{\lambda}, \widehat{\lambda}) = f(x) + \underbrace{\overline{\lambda}^T \overline{g}(x)}_{\leq 0} + \underbrace{\widehat{\lambda}^T \widehat{g}(x)}_{\leq 0} \leq f(x),$$

implying that $\inf_{x \in X} L(x; \overline{\lambda}, \widehat{\lambda}) \leq \mathrm{Opt}(P)$, that is $\underline{L}(\overline{\lambda}, \widehat{\lambda}) \leq \mathrm{Opt}(P)$ for all $[\overline{\lambda}; \widehat{\lambda}] \in \mathcal{LM}$, that is,

$$\mathrm{Opt}(D) \leq \mathrm{Opt}(P) \qquad \text{[weak duality]}$$

♠ We say that convex optimization problem in conic form

$$\mathrm{Opt}(P) = \min_{x \in X} \{f(x) : \overline{g}(x) = Ax - b \leq 0, \widehat{g}(x) \leq_{\mathbf{K}} 0\}, \qquad (P)$$

satisfies *Relaxed Slater Condition*, if there exists $\overline{x} \in \mathrm{rint}\, X$ such that $\overline{g}(\overline{x}) \leq 0$ and $\widehat{g}(\overline{x}) <_{\mathbf{K}} 0$.

♠ **Convex Duality Theorem, Conic Form:** *Consider convex optimization problem in conic form* (P) (so that X is convex, f is convex and real-valued on X, $\mathbf{K}$ is regular cone, and $\widehat{g}$ is well defined and $\mathbf{K}$-convex on X) *along with its Conic Lagrange Dual problem*

$$\mathrm{Opt}(D) = \max_{[\overline{\lambda}; \widehat{\lambda}] \in \mathcal{LM}} \left\{ \underline{L}(\overline{\lambda}, \widehat{\lambda}) := \inf_{x \in X} \left[f(x) + \overline{\lambda}^T \overline{g}(x) + \widehat{\lambda}^T \widehat{g}(x) \right] \right\} \qquad (D)$$

Assume that (P) *is below bounded and satisfies Relaxed Slater Condition. Then* (D) *is solvable, and*

$$\mathrm{Opt}(P) = \mathrm{Opt}(D).$$

♣ **Conic Programming.** Consider the special case of convex programming problem in conic form – one where the objective is linear, the domain X is the entire $\mathbb{R}^n$ and the function $\widehat{g}(x)$ is affine:

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \overline{g}(x) = Ax - b \leq 0,\ \widehat{g}(x) = Px - p \leq_{\mathbf{K}} 0 \right\} \qquad (P)$$

Optimization problem in this form is called *conic*.

Note: The entire structure, whatever it means, of a conic problem "sits" in the cone $\mathbf{K}$. As a matter of fact, *just 3 types of cones are responsible for nearly all applications of Convex Optimization:*

- *Finite direct products* $\mathbf{K}$ *of nonnegative rays — nonnegative orthants* $\mathbb{R}^n_+$ *giving rise to Linear Programming*
- *Finite direct products* $\mathbf{K}$ *of Lorentz cones* $\mathbf{L}^n = \{x \in \mathbb{R}^n : x_n \geq \sqrt{\sum_{\ell=1}^{n-1} x_\ell^2}\}$ *giving rise to Conic Quadratic (a.k.a. Second Order Conic) Programming,*
- *Finite direct products* $\mathbf{K}$ *of semidefinite cones* $\mathbf{S}^n_+ = \{A \in \mathbf{S}^n : A \succeq 0\}$ *comprised of positive semidefinite* $n \times n$ *matrices, giving rise to Semidefinite Programming*

♣ As far as Convex Programming is concerned, "expressive abilities" of Linear, Conic Quadratic and Semidefinite Programming are extremely strong.

Example: The messy problem

(o)	$\text{minimize } \sum_{\ell=1}^{n} x_\ell$
(a)	$x \geq 0;$
(b)	$a_\ell^T x \leq b_\ell,\ \ell = 1, ..., n;$
(c)	$\|Px - p\|_2 \leq c^T x + d;$
(d)	$x_\ell^{1+1/\ell} \leq e_\ell^T x + f_\ell,\ \ell = 1, ..., n;$
(e)	$x_\ell^{1/(\ell+3)} x_{\ell+1}^{\ell/(\ell+3)} \geq x_\ell^{-\ell/2} x_{\ell+1}^{-\ell} + g_\ell^T x + h_\ell,\ \ell = 1, ..., n-1;$
(f)	$\begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_2 & x_1 & x_2 & \cdots & x_{n-1} \\ x_3 & x_2 & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n & x_{n-1} & x_{n-2} & \cdots & x_1 \end{bmatrix} \succeq 0 \ \&\ \text{Det}\left(\begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_2 & x_1 & x_2 & \cdots & x_{n-1} \\ x_3 & x_2 & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n & x_{n-1} & x_{n-2} & \cdots & x_1 \end{bmatrix}\right) \geq 1;$
(g)	$1 \leq \sum_{\ell=1}^{n} x_\ell \cos(\ell\omega) \leq 1 + \sin^2(5\omega)\ \forall \omega \in \left[-\frac{\pi}{7}, 1.3\right]$

can be converted, *in a systematic way* (by compiler!), into an equivalent conic problem:

- $(o$–$b)$ is just LP
- $(o$–$e)$ is a Conic Quadratic problem
- $(o$–$g)$ is a Semidefinite problem

$\Rightarrow$ *seemingly highly diverse constraints of the original problem allow for unified treatment.*

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \underbrace{Ax - b}_{\overline{g}(x)} \leq 0,\ \underbrace{Px - p}_{\widehat{g}(x)} \leq_{\mathbf{K}} 0 \right\} \qquad (P)$$

♠ Linear, Conic Quadratic, and Semidefinite Programming possess deep intrinsic structural similarity allowing for unified design of theoretically and practically efficient algorithms – *Interior Point Path-Following Methods.* These are *the* algorithms used when high accuracy solutions to convex problems are sought.

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \underbrace{Ax - b}_{\overline{g}(x)} \leq 0,\ \underbrace{Px - p}_{\widehat{g}(x)} \leq_{\mathbf{K}} 0 \right\} \qquad (P)$$

♠ Conic Lagrange Dual to (P) is the problem

$$\begin{aligned}
\mathrm{Opt}(D) &= \max_{\overline{\lambda} \in \mathbb{R}^k_+, \widehat{\lambda} \in K_*} \left\{ \underline{L}(\overline{\lambda}, \widehat{\lambda}) = \underbrace{\min_x \left[c^T x + \overline{\lambda}^T [Ax - b] + \widehat{\lambda}^T [Px - p] \right]}_{\min_x x^T[c + A^T\overline{\lambda} + P^T\widehat{\lambda}] - b^T\overline{\lambda} - p^T\widehat{\lambda}} \right\} \\
&= \max_{\overline{\lambda}, \widehat{\lambda}} \left\{ -b^T\overline{\lambda} - p^T\widehat{\lambda} : c + A^T\overline{\lambda} + P^T\widehat{\lambda} = 0, \overline{\lambda} \geq 0, \widehat{\lambda} \geq_{\mathbf{K}_*} 0 \right\}
\end{aligned} \qquad (D)$$

Fact: *The dual problem is conic, and the duality is symmetric – the problem dual to (D) is (equivalent to) (P)*

Convex Duality Theorem in Conic Form combines with symmetry of Conic Duality to imply the following

Conic Duality Theorem: *Let one of the conic problems in the primal-dual pair (P), (D) be bounded and satisfy the Relaxed Slater Condition. Then the other problem is solvable, and $\mathrm{Opt}(P) = \mathrm{Opt}(D)$. Besides this, we always have $\mathrm{Opt}(D) \leq \mathrm{Opt}(P)$.*

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \underbrace{Ax - b}_{\overline{g}(x)} \leq 0, \underbrace{Px - p}_{\widehat{g}(x)} \leq_{\mathbf{K}} 0 \right\} \qquad (P)$$

$$\mathrm{Opt}(D) = \max_{\overline{\lambda}, \widehat{\lambda}} \left\{ -b^T \overline{\lambda} - p^T \widehat{\lambda} : c + A^T \overline{\lambda} + P^T \widehat{\lambda} = 0, \overline{\lambda} \geq 0, \widehat{\lambda} \geq_{\mathbf{K}_*} 0 \right\} \qquad (D)$$

♠ From Conic Duality Theorem one easily extracts

Optimality Conditions in Conic Programming: *Let (P), (D) satisfy Relaxed Slater Condition and let $x_*, (\overline{\lambda}_*, \widehat{\lambda}_*)$ be feasible solutions to (P) and to (D). Then the solutions x_* and $(\overline{\lambda}_*, \widehat{\lambda}_*)$ are optimal for the respective problems*

["zero duality gap"] *Iff* $c^T x_* = -b^T \overline{\lambda}_* - p^T \widehat{\lambda}_*$

and

["complementary slackness"] *Iff* $\overline{\lambda}_*^T [Ax_* - b] = 0$ *and* $\widehat{\lambda}_*^T [Px_* - p] = 0$

and these equivalent to each other facts take place iff $(x_*, [\overline{\lambda}_*; \widehat{\lambda}_*])$ is a saddle point (min in $x \in \mathbb{R}^n$, max in $[\overline{\lambda}; \widehat{\lambda}] \in \mathcal{LM}$) *of the Conic Lagrange function*

$$L(x; [\overline{\lambda}; \widehat{\lambda}]) = c^T x + \overline{\lambda}^T \overline{g}(x) + \widehat{\lambda}^T \widehat{g}(x)$$

on the domain $(x \in \mathbb{R}^n, [\overline{\lambda}; \widehat{\lambda}] \in \mathcal{LM})$.

Geometry of Primal-dual Pair of Conic Problems

Fact: *A primal-dual pair of conic problems can be equivalently reformulated in the following geometric form:*

`https://www2.isye.gatech.edu/~nemirovs/LMCOLN2021WithSol.pdf`, Section 1.4.4

♠ **Given are:**

- a regular cone $\mathbf{M} \subset \mathbb{R}^n$ along with its dual cone $\mathbf{M}_*$
- a linear subspace $\mathcal{L} \subset \mathbb{R}^n$ along with its orthogonal complement $\mathcal{L}^\perp \subset \mathbb{R}^n$
- two shift vectors $e \in \mathbb{R}^n$, $f \in \mathbb{R}^n$

♠ **Find:** *a pair of vectors*

$$\xi_* \in [\mathcal{L} + e] \cap \mathbf{M}, \ \lambda_* \in [\mathcal{L}^\perp + f] \cap \mathbf{M}_*$$

which are orthogonal to each other:

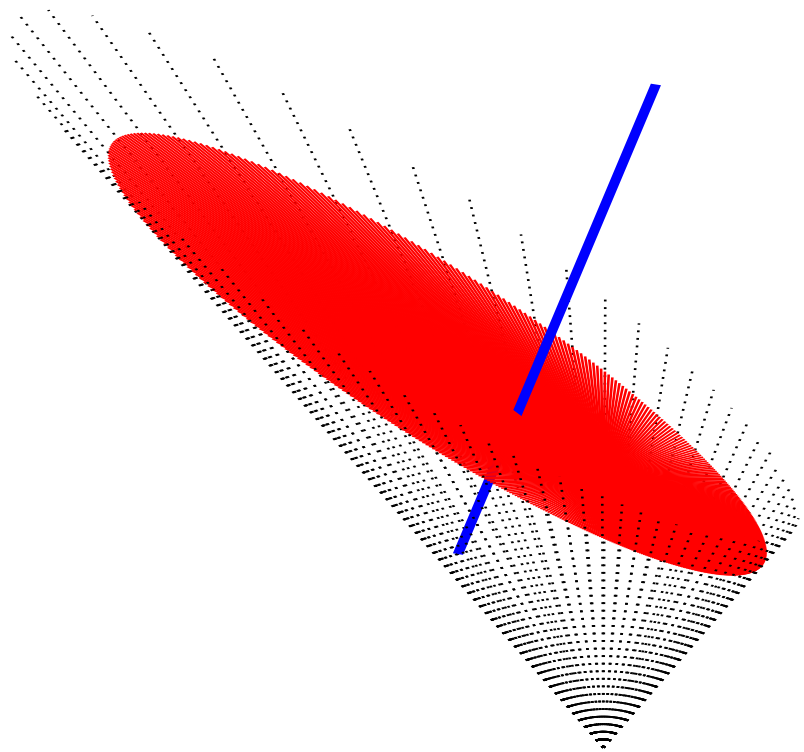
$$\xi_*^T \lambda_* = 0.$$

Fact: *Under primal-dual strict feasibility:*

$$[\mathcal{L} + e] \cap \operatorname{int} \mathbf{M} \neq \emptyset \ \& \ [\mathcal{L}^\perp + f] \cap \operatorname{int} \mathbf{M}_* \neq \emptyset$$

the solution does exist.

Find $\xi_* \in [\mathcal{L} + e] \cap \mathbf{M}$ & $\lambda_* \in [\mathcal{L}^\perp + f] \cap \mathbf{M}_*$: $\xi_*^T \lambda_* = 0$



Geometric form of primal-dual pair $(\mathcal{P})$, $(\mathcal{D})$ of conic problems on 3D Lorentz cone $\mathbf{M} = \mathbf{M}_*$

Red: feasible set $[\mathcal{L} + e] \cap \mathbf{M}$ of $(\mathcal{P})$

Blue: feasible set $[\mathcal{L}^\perp + f] \cap \mathbf{M}_*$ of $(\mathcal{D})$

Lecture 7:
Optimality Conditions

Optimality Conditions in Mathematical Programming

♣ **Situation:** We are given a Mathematical Programming problem

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \\ x \in X \end{array} \right\}. \qquad (P)$$

Question of interest: *Assume that we are given a feasible solution x_* to (P). What are the conditions (necessary, sufficient, necessary and sufficient) for x_* to be optimal?*

Fact: Except for *convex programs*, *there are no verifiable local sufficient conditions for global optimality*. There exist, however,

♢ verifiable local *necessary* conditions for *local* (and thus – for *global*) optimality

♢ verifiable local *sufficient* conditions for *local* optimality

Fact: Existing conditions for local optimality assume that $x_* \in \operatorname{int} X$, which, from the viewpoint of local optimality of x_*, is exactly the same as to say that $X = \mathbb{R}^n$.

♣ **Situation:** We are given a Mathematical Programming problem

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\}. \qquad (P)$$

and a feasible solution x_* to the problem, and are interested in necessary/sufficient conditions for *local optimality* of x_*, meaning that

There exists $r > 0$ such that for every feasible x with $\|x - x_*\| \leq r$ one has

$$f(x) \geq f(x_*).$$

Default assumption: The objective and all the constraints are continuously differentiable in a neighborhood of x_*.

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\}. \qquad (P)$$

♣ **First Order Optimality Conditions** are expressed via values and gradients of the objective and the constraints at x_*. Except for convex case, only *necessary* First Order conditions are known.

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\}. \qquad (P)$$

The idea:

$\diamondsuit$ Assume that x_* is locally optimal for (P). Let us approximate (P) around x_* by a Linear Programming program

$$\begin{array}{l} \min_x f(x_*) + (x - x_*)^T f'(x_*) \\ \text{s.t.} \\ \overbrace{g_j(x_*)}^{0} + (x - x_*)^T g_j'(x_*) \leq 0, \; j \in J(x_*) \\ \underbrace{h_i(x_*)}_{0} + (x - x_*)^T h_i'(x_*) = 0, \; 1 \leq i \leq k \end{array} \qquad (LP)$$

$$\left[J(x_*) = \{j : g_j(x_*) = 0\} \right]$$

Note: Since all $g_j(\cdot)$ are continuous at x_*, the *non-active at* x_* inequality constraints (those with $g_j(x_*) < 0$) do not affect local optimality of x_* and do not participate in (LP).

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \qquad (P)$$

$$\Rightarrow \min_x \left\{ (x - x_*)^T f'(x_*) : \begin{array}{l} (x - x_*)^T g_j'(x_*) \leq 0, \\ j \in J(x_*) \\ (x - x_*)^T h_i'(x_*) = 0, \\ i = 1, ..., k \end{array} \right\} \qquad (LP)$$

$$J(x_*) = \{j : g_j(x_*) = 0\}$$

$\diamondsuit$ It is natural *to guess* that if x_* is locally optimal for (P), then x_* is locally optimal for (LP) as well.

• LP is a *convex* program with *affine* constraints, whence the KKT conditions are necessary and sufficient for optimality:

$$\begin{array}{rl} & x_* \text{ is optimal for } (LP) \\ \Leftrightarrow & \exists(\lambda_j^* \geq 0, j \in J(x_*), \mu_i^*) : f'(x_*) + \sum\limits_{j \in J(x_*)} \lambda_j^* g_j'(x_*) + \sum\limits_{i=1}^k \mu_i^* h_i'(x_*) = 0 \\ \Leftrightarrow & \exists(\lambda_j^* \geq 0, \mu_i^*) : \left\{ \begin{array}{l} f'(x_*) + \sum\limits_j \lambda_j^* g_j'(x_*) + \sum\limits_i \mu_i^* h_i'(x_*) = 0 \\ \lambda_j^* g_j(x_*) = 0, \ j = 1, ..., m \end{array} \right. \end{array}$$

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \qquad (P)$$

$$\Rightarrow L(x; \lambda, \mu) = f(x) + \sum_j \lambda_j g_j(x) + \sum_i \mu_i h_i(x)$$

Proposition. *Let x_* be a locally optimal solution of (P).*
Assume that x_* remains locally optimal when passing from (P) to the linearized problem

$$\min_x \left\{ (x - x_*)^T f'(x_*) : \begin{array}{l} (x - x_*)^T g_j'(x_*) \leq 0,\ j \in J(x_*) = \{j : g_j(x_*) = 0\} \\ (x - x_*)^T h_i'(x_*) = 0,\ i = 1, ..., k \end{array} \right\} \qquad (LP)$$

Then at x_* the KKT condition holds:

$$\exists(\lambda_j^* \geq 0, \mu_i^*) : \left\{ \begin{array}{l} f'(x_*) + \sum_j \lambda_j^* g_j'(x_*) + \sum_i \mu_i^* h_i'(x_*) = 0 \text{ [KKT equality]} \\ \lambda_j^* g_j(x_*) = 0,\ j = 1, ..., m \text{ [complementary slackness]} \end{array} \right.$$

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \qquad (P)$$

$$\min_x \left\{ (x - x_*)^T f'(x_*) : \begin{array}{l} (x - x_*)^T g_j'(x_*) \leq 0,\ j \in J(x_*) = \{j : g_j(x_*) = 0\} \\ (x - x_*)^T h_i'(x_*) = 0,\ i = 1, ..., k \end{array} \right\} \qquad (LP)$$

To make Proposition useful, we need a verifiable sufficient condition for *"x_* remains locally optimal when passing from (P) to (LP)"*.

The most natural form of such a condition is *regularity*:

Gradients, taken at x_*, of all constraints active at x_* are linearly independent.

Of course, all equality constraints by definition are active at every feasible solution.

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \qquad (P)$$

$$\min_x \left\{ (x - x_*)^T f'(x_*) : \begin{array}{l} (x - x_*)^T g_j'(x_*) \leq 0,\ j \in J(x_*) = \{j : g_j(x_*) = 0\} \\ (x - x_*)^T h_i'(x_*) = 0, i = 1, ..., k \end{array} \right\} \qquad (LP)$$

Proposition: *Let x_* be a locally optimal regular solution of (P). Then x_* is optimal for (LP) and, consequently, the KKT conditions take place at x_*.*

Note: *Regularity of x_* is important for the KKT condition to be necessary for local optimality.* For example, $x_* = 0$ is the only feasible solution to the problem

$$\min_x \{ f(x) := x : h(x) := x^2 = 0 \}$$

and therefore is even globally optimal. The KKT condition would say that there exists μ^* such that

$$0 = \nabla f(x_*) + \mu^* \nabla h(x_*) = 1 + \mu^* \cdot 0,$$

which is impossible. The source of the difficulty is that $\nabla h(x_*) = 0$, that is, $x_* = 0$ is not a *regular* locally optimal solution.

Proof of Theorem is based on an important fact of Analysis – a version of Implicit Function Theorem.

Theorem: *Let $x_* \in \mathbb{R}^n$ and let $p_\ell(x)$, $\ell = 1, ..., L$, be real-valued functions such that*

$\diamondsuit$ *p_ℓ are $\kappa \geq 1$ times continuously differentiable in a neighborhood of x_**

$\diamondsuit$ $p_\ell(x_*) = 0$

$\diamondsuit$ *vectors $\nabla p_\ell(x_*)$, $\ell = 1, ..., L$, are linearly independent.*

Then there exists substitution of variables

$$y \mapsto x = \Phi(y)$$

defined in a neighborhood V of the origin and mapping V, in a one-to-one manner, onto a neighborhood B of x_*, such that

$\diamondsuit$ $x_* = \Phi(0)$

$\diamondsuit$ *both $\Phi : V \to B$ and its inverse mapping $\Phi^{-1} : B \to V$ are κ times continuously differentiable*

$\diamondsuit$ *in coordinates y, the functions p_ℓ become just the coordinates:*

$$y \in V \Rightarrow p_\ell(\Phi(y)) \equiv y_\ell, \ \ell = 1, ..., L.$$

♠ This is how Implicit Function Theorem leads to KKT Optimality conditions:

Situation: We are given *regular locally optimal* solution x_* to the optimization problem

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \qquad (P)$$

and want to prove that at x_* the KKT conditions take place. W.l.o.g. we can assume that the inequality constraints active at x are those with $j = 1, ..., \ell$.

Step 1. Using regularity of x_* and Implicit Function Theorem, we find local continuously differentiable, along with its inverse, substitution of variables $x = \Phi(y)$ which maps $y_* = 0$ onto x_* and is such that the functions $h_i(\Phi(y))$ and $g_j(\Phi(y))$, $j \leq \ell$, are just the coordinates:

$$h_i(\Phi(y)) \equiv y_i,\ i \leq k;\ g_j(\Phi(y)) \equiv y_{k+j},\ j \leq \ell.$$

Since x_* is locally optimal solution to (P), $y_* = 0$ is locally optimal solution to the problem

$$\min_y \left\{ \phi(y) := f(\Phi(y)) : y_i = 0, i \leq k, y_{k+j} \leq 0, j \leq \ell \right\} \qquad (P')$$

Since x_* is locally optimal solution to (P), $y_* = 0$ is locally optimal solution to the problem

$$\min_y \{\phi(y) := f(\Phi(y)) : y_i = 0, i \leq k, y_{k+j} \leq 0, j \leq \ell\} \qquad (P')$$

Step 2. The constraints in Problem (P') are that simple, that one can immediately verify that the locally optimal solution $y_* = 0$ to this problem is optimal for the linearization of (P'). Due to local smoothness of substitution Φ and its inverse, *linearizations of* (P) *and* (P') *are obtained from each other by one-to-one linear substitution of variables* $y \mapsto x = x_* + \Phi'(0)y$ given by the Jacobian of Φ at $y_* = 0$, implying that x_* is an optimal solution to the linearization of (P), and thus - a KKT point of (P).

$$\min_x \left\{ (x-x_*)^T f'(x_*) : \begin{array}{l} (x-x_*)^T g_j'(x_*) \le 0,\ j \in J(x_*) \\ (x-x_*)^T h_i'(x_*) = 0,\ i = 1,...,k \end{array} \right\} \qquad (LP)$$

Formal proof: Let x_* be a regular locally optimal solution to (P); assume, on the contrary to what should be proven, that x_* is not an optimal solution to (LP), and let us lead this to contradiction.

1^0. Since $x = x_*$ is not an optimal solution to (LP), there exists a feasible solution

$$x' = x_* + d$$

to the problem with

$$(x'-x_*)^T f'(x_*) = d^T f'(x_*) < 0,$$

so that

$$d^T f'(x_*) < 0,\ \underbrace{d^T h_i'(x_*) = 0}_{\forall i},\ \underbrace{d^T g_j'(x_*) \le 0}_{\forall j \in J(x_*)}$$

2^0. W.l.o.g., assume that $J(x_*) = \{1,...,\ell\}$. By Theorem, there exists continuously differentiable local substitution of argument

$$x = \Phi(y) \qquad [\Phi(0) = x_*]$$

with a continuously differentiable in a neighborhood of x_* inverse $y = \Psi(x)$ such that in a neighborhood of origin one has

$$h_i(\Phi(y)) \equiv y_i,\ g_j(\Phi(y)) = y_{k+j},\ j = 1,...,\ell.$$

Since $\Psi(\Phi(y)) \equiv y$, we have $\Psi'(x_*)\Phi'(0) = I$, whence

$$\exists e : \Phi'(0)e = d.$$

Situation: We have found a smooth local substitution of argument $x = \Phi(y)$ ($y = 0$ corresponds to $x = x_*$) and a direction e such that in a neighborhood of $y = 0$ one has

$$\begin{array}{lrcl}
(a) & h_i(\Phi(y)) & \equiv & y_i,\ i \le k \\
(b) & g_j(\Phi(y)) & \equiv & y_{k+j},\ j \le \ell\ [J(x_*) = \{1, ..., \ell\}] \\
(c) & [\Phi'(0)e]^T h_i'(x_*) & = & 0,\ i \le k \\
(d) & [\Phi'(0)e]^T g_j'(x_*) & \le & 0,\ j \le \ell \\
(e) & [\Phi'(0)e]^T f'(x_*) & < & 0
\end{array}$$

Consider the differentiable curve

$$x(t) = \Phi(te).$$

We have

$$\begin{array}{l}
te_i \equiv h_i(\Phi(te)) \Rightarrow e_i = [\Phi'(0)e]^T h_i'(x_*) = 0 \\
te_{k+j} \equiv g_j(\Phi(te)) \Rightarrow e_{k+j} = [\Phi'(0)e]^T g_j'(x_*) \le 0 \\
\Rightarrow \underbrace{h_i(x(t)) = te_i = 0}_{\forall i},\ \underbrace{g_j(x(t)) = te_{k+j} \le 0}_{\forall j \in J(x_*)}
\end{array}$$

Thus, $x(t)$ is feasible for all small $t \ge 0$. But:

$$\frac{d}{dt}\Big|_{t=0} f(x(t)) = [\Phi'(0)e]^T f'(x_*) < 0,$$

whence $f(x(t)) < f(x(0)) = f(x_*)$ for all small enough $t > 0$, which is a contradiction with local optimality of x_*.

Second Order Optimality Conditions

♣ In the case of unconstrained minimization problem

$$\min_x f(x) \qquad (P)$$

with continuously differentiable objective, the KKT conditions reduce to Fermat Rule: *If x_* is locally optimal for (P), then $\nabla f(x_*) = 0$.*
Fermat Rule is the "first order" part of
Second Order Necessary Optimality Condition in unconstrained minimization:
If x_* is locally optimal for (P) and f is twice differentiable in a neighborhood of x_*, then

$$\nabla f(x_*) = 0 \ \& \ \nabla^2 f(x_*) \succeq 0 \Leftrightarrow d^T \nabla^2 f(x_*) d \geq 0 \forall d$$

If x_* is locally optimal for (P) and f is twice differentiable in a neighborhood of x_*, then

$$\nabla f(x_*) = 0 \ \& \ \nabla^2 f(x_*) \succeq 0 \Leftrightarrow d^T \nabla^2 f(x_*) d \geq 0 \forall d$$

Indeed, let x_* be locally optimal for (P); then for appropriate $r_d > 0$

$$\begin{aligned}
&0 \leq t \leq r_d \\
&\Rightarrow 0 \leq f(x_* + td) - f(x_*) = t \underbrace{d^T \nabla f(x_*)}_{=0} + \tfrac{1}{2} t^2 d^T \nabla^2 f(x_*) d + t^2 \underbrace{R_d(t)}_{\substack{\to 0, \\ t \to 0}} \\
&\Rightarrow \tfrac{1}{2} d^T \nabla^2 f(x_*) d + R_d(t) \geq 0 \Rightarrow d^T \nabla^2 f(x_*) d \geq 0
\end{aligned}$$

$$\min_x f(x) \qquad (P)$$

The *necessary* Second Order Optimality Condition in unconstrained minimization can be strengthened to

Second Order Sufficient Optimality Condition in unconstrained minimization: *Let f be twice differentiable in a neighborhood of x_*. If*

$$\nabla f(x_*) = 0 \ \& \ \nabla^2 f(x_*) \succ 0 \Leftrightarrow d^T \nabla^2 f(x_*) d > 0 \forall d \neq 0$$

then x_* is locally optimal for (P).

Proof: Since $d^T \nabla^2 f(x_*) d > 0$ for all $d > 0$, then there exists $\alpha > 0$ such that $d^T \nabla^2 f(x_*) d \geq \alpha d^T d$ for all d.

By differentiability, for every $\epsilon > 0$ there exists $r_\epsilon > 0$ such that

$$\begin{aligned}
&\|d\|_2 \leq r_\epsilon \\
&\Rightarrow f(x_* + d) - f(x_*) \geq \underbrace{d^T \nabla f(x_*)}_{=0} + \tfrac{1}{2} \underbrace{d^T \nabla^2 f(x_*) d}_{\geq \alpha d^T d} - \tfrac{\epsilon}{2} d^T d \\
&\Rightarrow f(x_* + d) - f(x_*) \geq \tfrac{1}{2}(\alpha - \epsilon) d^T d
\end{aligned}$$

Setting $\epsilon = \frac{\alpha}{2}$, we see that x_* is a local minimizer of f.

We are given a Mathematical Programming problem

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \quad (P)$$

$$\Downarrow$$

$$L(x; \lambda, \mu) = f(x) + \sum_j \lambda_j g_j(x) + \sum_i \mu_i h_i(x)$$

♣ In Optimality Conditions for a constrained problem (P), the role of $\nabla^2 f(x_*)$ is played by the Hessian of the *Lagrange function:*

Second Order Necessary Optimality Condition: *Let x_* be a regular feasible solution of (P) such that the functions f, g_j, h_i are twice continuously differentiable in a neighborhood of x_*. If x_* is locally optimal, then*

♢ [first order part] *there exist uniquely defined Lagrange multipliers $\lambda_j^* \geq 0$, μ_i^* such that the KKT conditions hold:*

$$\nabla_x L(x_*; \lambda^*, \mu^*) = 0 \ \& \ \lambda_j^* g_j(x_*) = 0, \ j = 1, ..., m$$

♢ [second order part] *for every d orthogonal to the gradients, taken at x_*, of all equality constraints and all active at x_* inequality constraints, one has*

$$d^T \nabla_x^2 L(x_*; \lambda^*, \mu^*) d \geq 0.$$

Sketch of the proof:

• As with the KKT condition, assuming w.l.o.g. that the active at x_* inequality constraints are $g_j(x) \leq 0$, $1 \leq j \leq \ell$, we use Implicit Function Theorem to pass from the problem

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \qquad (P)$$

in x-variables to much simpler (locally) equivalent problem

$$\min_y \left\{ \phi(y) := f(\Phi(y)) : y_i = 0, i \leq k, y_{k+j} \leq 0, j \leq \ell \right\} \qquad (P')$$

in y-variables using locally one-to-one twice continuously differentiable, along with its inverse, substitution of variables $x = \Phi(y)$.

• For (P'), the Necessary Second Order Optimality Condition is nearly evident (it is readily given by similar condition for the unconstrained case).

• When translating the latter condition back to x-variables, we get exactly what we are looking for.

Proof. 1^0. Constraints which are non-active at x_* clearly affect neither local optimality of x_*, nor the conclusion to be proven. Removing these constraints, we reduce the situation to one where *all constraints in the problem*

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \qquad (P)$$

are active at x_*.

2^0. Applying Implicit Function Theorem, we can find a local change of variables

$$x = \Phi(y) \Leftrightarrow y = \Psi(x) \qquad [\Phi(0) = x_*, \Psi(x_*) = 0]$$

with locally twice continuously differentiable Φ, Ψ such that

$$g_j(\Phi(y)) \equiv y_j,\ j \leq m, h_i(\Phi(y)) \equiv y_{m+i},\ i \leq k.$$

In variables y, problem (P) becomes

$$\min_y \left\{ \underbrace{f(\Phi(y))}_{\phi(y)} : y_j \leq 0,\ j \leq m, y_{m+i} = 0,\ i \leq k \right\}. \qquad (P')$$

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \qquad (P)$$

$$\Updownarrow \qquad [x = \Phi(y)]$$

$$\min_y \left\{ \underbrace{f(\Phi(y))}_{\phi(y)} : y_j \leq 0,\ j \leq m, y_{m+i} = 0,\ i \leq k \right\} \qquad (P')$$

$$\Downarrow$$

$$M(y; \lambda, \mu) = \phi(y) + \sum_{j=1}^{m} \lambda_j y_j + \sum_{i=1}^{k} \mu_i y_{m+i}$$

Our plan is as follows:

$\diamondsuit$ Since Φ is a smooth one-to-one mapping of a neighborhood of x_* onto a neighborhood of $y_* = 0$, x_* is locally optimal for (P) iff $y_* = 0$ is locally optimal for (P').

$\diamondsuit$ We intend to build necessary/sufficient conditions for $y_* = 0$ to be locally optimal for (P'); "translated" to x-variables, these conditions will imply necessary/sufficient conditions for local optimality of x_* for (P).

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \qquad (P)$$

$$\Updownarrow \qquad [x = \Phi(y)]$$

$$\min_y \left\{ \underbrace{f(\Phi(y))}_{\phi(y)} : y_j \leq 0,\ j \leq m, y_{m+i} = 0,\ i \leq k \right\} \qquad (P')$$

$$\Rightarrow M(y; \lambda, \mu) = \phi(y) + \sum_{j=1}^{m} \lambda_j y_j + \sum_{i=1}^{k} \mu_i y_{m+i}$$

3^0. Since $x_* = \Phi(0)$ is locally optimal for (P), $y_* = 0$ is locally optimal for (P'). In particular, if e_i is i-th basic orth, then for appropriate $\epsilon > 0$:

$$\begin{array}{rcl} j \leq m & \Rightarrow & y(t) = -te_j \text{ is feasible for } (P') \text{ when} \\ \epsilon \geq t \geq 0 & \Rightarrow & -\frac{\partial \phi(0)}{\partial y_t} = \frac{d}{dt}\Big|_{t=0} \phi(y(t)) \geq 0 \\ & \Rightarrow & \lambda_j^* \equiv -\frac{\partial \phi(0)}{\partial y_i} \geq 0 \end{array}$$

and

$$\begin{array}{rcl} s > m + k & \Rightarrow & y(t) = te_s \text{ is feasible for } (P') \text{ when} \\ \epsilon \geq t \geq -\epsilon & \Rightarrow & \frac{\partial \phi(0)}{\partial y_s} = \frac{d}{dt}\Big|_{t=0} \phi(y(t)) = 0 \end{array}$$

Setting $\mu_i^* = -\frac{\partial \phi(0)}{\partial y_{m+i}}$, $i = 1, ..., k$, we get

$$\lambda^* \geq 0 \ \& \ \nabla_y M(0; \lambda^*, \mu^*) = 0. \qquad \text{(KKT)}$$

Situation: $y_* = 0$ is locally optimal for

$$\min_y \left\{ \phi(y) \equiv f(\Phi(y)) : \begin{array}{l} y_j \le 0,\ j \le m \\ y_{m+i} = 0,\ i \le k \end{array} \right\} \quad (P')$$

$$\Downarrow$$

$$M(y;\lambda,\mu) = \phi(y) + \sum_{j=1}^{m} \lambda_j y_j + \sum_{i=1}^{k} \mu_i y_{m+i}$$

and $\exists \lambda^* \ge 0, \mu^*$:

$$0 = \frac{\partial M(0;\lambda^*,\mu^*)}{\partial y_\ell} \equiv \begin{cases} \frac{\partial \phi(0)}{\partial y_\ell} + \lambda^*_\ell, & \ell \le m \\ \frac{\partial \phi(0)}{\partial y_\ell} + \mu^*_{\ell-m}, & m < \ell \le m+k \\ \frac{\partial \phi(0)}{\partial y_\ell}, & \ell > m+k \end{cases} \qquad \text{(KKT)}$$

Note that the condition $\nabla_y M(0;\lambda^*,\mu^*) = 0$ defines λ^*, μ^* in a unique fashion.

4^0. We have seen that for (P'), the first order part of the Necessary Second Order Optimality Condition holds true. Let us prove the second order part of the condition, which reads

$$\forall (d : d^T \nabla_y y_\ell = 0,\ \ell \le m+k) : \\ d^T \nabla_y^2 M(0;\lambda^*,\mu^*) d \ge 0.$$

Situation: $y_* = 0$ is locally optimal solution to the problem

$$\min_y \left\{ \phi(y) \equiv f(\Phi(y)) : \begin{array}{l} y_j \le 0,\ j \le m \\ y_{m+i} = 0,\ i \le k \end{array} \right\} \qquad (P')$$

Claim:

$$\forall (d : d^T \nabla_y y_\ell = 0,\ \ell \le m+k) : \\ d^T \nabla_y^2 M(0; \lambda^*, \mu^*) d \ge 0.$$

This is evident: since $M(y; \lambda^*, \mu^*) = \phi(y) + \sum_{j=1}^{m} \lambda_j^* y_j + \sum_{i=1}^{k} \mu_i^* y_{m+i}$, we have

$$\nabla_y^2 M(0; \lambda^*, \mu^*) = \nabla^2 \phi(0).$$

Claim therefore states that $d^T \nabla^2 \phi(0) d \ge 0$ for every vector d from the linear subspace $L = \{d : d_1 = ... = d_{m+k} = 0\}$. But this subspace is feasible for (P'), so that ϕ, restricted onto L, should attain unconstrained local minimum at the origin. By Necessary Second Order Optimality Condition for unconstrained minimization,

$$d^T \nabla^2 \phi(0) d \ge 0 \ \forall d \in L.$$

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \qquad (P)$$

$$\Updownarrow \qquad [x = \Phi(y)]$$

$$\min_y \left\{ \underbrace{f(\Phi(y))}_{\phi(y)} : y_j \leq 0,\ j \leq m, y_{k+i} = 0,\ i \leq k \right\} \qquad (P')$$

$$\Rightarrow M(y; \lambda, \mu) = \phi(y) + \sum_j \lambda_j y_j + \sum_i \mu_i y_{m+i}$$

5^0. We have seen that if x_* is locally optimal for (P), then there exist uniquely defined $\lambda^* \geq 0$, μ^* such that

$$\nabla_y M(0; \lambda^*, \mu^*) = 0,$$

$$d^T \nabla_y y_\ell = 0,\ \ell \leq m + k \Rightarrow d^T \nabla_y^2 M(0; \lambda^*, \mu^*) d \geq 0.$$

Let us prove that then

$$\nabla_x L(x_*; \lambda^*, \mu^*) = 0 \qquad (*)$$

and

$$\left. \begin{array}{l} e^T g_j'(x_*) = 0,\ j \leq m \\ e^T h_i'(x_*) = 0,\ i \leq k \end{array} \right\} \Rightarrow e^T \nabla_x^2 L(x_*; \lambda^*, \mu^*) e \geq 0. \qquad (**)$$

Given:

$$\nabla_y M(0;\lambda^*,\mu^*) = 0$$
$$d^T\nabla_y y_\ell = 0,\ \ell \le m+k \Rightarrow d^T\nabla_y^2 M(0;\lambda^*,\mu^*)d \ge 0.$$

Should prove:

$$\nabla_x L(x_*;\lambda^*,\mu^*) = 0 \qquad (*)$$
$$\left.\begin{array}{l} e^T g_j'(x_*) = 0,\ j \le m \\ e^T h_i'(x_*) = 0,\ i \le k \end{array}\right\} \Rightarrow e^T\nabla_x^2 L(x_*;\lambda^*,\mu^*)e \ge 0 \qquad (**)$$

$\diamondsuit$ Setting $\mathcal{L}(x) = L(x;\lambda^*,\mu^*)$, $\mathcal{M}(y) = M(y;\lambda^*,\mu^*)$, we have

$$\mathcal{L}(x) = \mathcal{M}(\Psi(x))$$
$$\Rightarrow \quad \nabla_x\mathcal{L}(x_*) = [\Psi'(x_*)]^T\nabla_y\mathcal{M}(y_*) = 0,$$

as required in $(*)$.

Given:

$$\begin{cases} \nabla_y M(0;\lambda^*,\mu^*) = 0 \\ d^T\nabla_y y_\ell = 0,\ \ell \le m+k \Rightarrow d^T\nabla_y^2 M(0;\lambda^*,\mu^*)d \ge 0. \end{cases}$$

Should prove:

$$\left.\begin{array}{c} e^T g_j'(x_*) = 0,\ j \le m \\ e^T h_i'(x_*) = 0,\ i \le k \end{array}\right\} \Rightarrow e^T\nabla_x^2 L(x_*;\lambda^*,\mu^*)e \ge 0 \quad (**)$$

◇ Let e satisfy the premise in $(**)$, and let $d = [\Phi'(0)]^{-1}e$. Then

$\overbrace{\left.\frac{d}{dt}\right\|_{t=0} g_j(\Phi(td))}^{\frac{d}{dt}\|_{t=0} td_j} = [g_j'(x_*)]^T \overbrace{[\Phi'(0)]d}^{e}$ $\Rightarrow d_j = e^T g_j'(x_*) = 0,\ j \le m$	$\underbrace{\left.\frac{d}{dt}\right\|_{t=0} h_i(\Phi(td))}_{\frac{d}{dt}\|_{t=0} td_{m+i}} = [h_i'(x_*)]^T \underbrace{[\Phi'(0)]d}_{e}$ $\Rightarrow d_{m+i} = e^T h_i'(x_*) = 0,\ i \le k$

We have

$$\begin{aligned} e^T\nabla^2\mathcal{L}(x_*)e &= \left.\frac{d^2}{dt^2}\right|_{t=0} \mathcal{L}(x_*+te) = \left.\frac{d^2}{dt^2}\right|_{t=0} \mathcal{M}(\Psi(x_*+te)) \\ &= \left.\frac{d}{dt}\right|_{t=0} \left[e^T[\Psi'(x_*+te)]^T\nabla\mathcal{M}(\Psi(x_*+te))\right] \\ &= e^T[\Psi'(x_*)]^T\nabla^2\mathcal{M}(0)\ \overbrace{[\Psi'(x_*)e]}^{=[\Phi'(0)]^{-1}e=d} + e^T[\left.\tfrac{d}{dt}\right|_{t=0}\Psi'(x_*+te)]^T \underbrace{\nabla\mathcal{M}(0)}_{=0} \\ &= d^T\nabla^2\mathcal{M}d \ge 0 \text{ due to } d_j = 0,\ 1 \le j \le m+k \end{aligned}$$

Thus, whenever e is orthogonal to the gradients of all constraints active at x_*, we have $e^T\nabla^2\mathcal{L}e \ge 0$.

Second Order Sufficient Condition for Local Optimality

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \quad (P)$$

$$\Downarrow$$

$$L(x; \lambda, \mu) = f(x) + \sum_j \lambda_j g_j(x) + \sum_i \mu_i h_i(x)$$

Second Order Sufficient Optimality Condition: *Let x_* be a regular feasible solution of (P) such that the functions f, g_j, h_i are twice continuously differentiable in a neighborhood of x_*. If there exist Lagrange multipliers $\lambda_j^* \geq 0$, μ_i^* such that*

$\diamondsuit$ [first order part] *the KKT conditions hold:*

$$\nabla_x L(x_*; \lambda^*, \mu^*) = 0 \ \& \ \lambda_j^* g_j(x_*) = 0, \ j = 1, ..., m$$

$\diamondsuit$ [second order part] *for every $d \neq 0$ orthogonal to the gradients, taken at x_*, of all equality constraints and those active at x_* inequality constraints for which $\lambda_j^* > 0$, one has*

$$d^T \nabla_x^2 L(x_*; \lambda^*, \mu^*) d > 0$$

then x_* is locally optimal for (P).

Note: Difference between Sufficient and Necessary optimality conditions is in their "second order" parts and is twofold:

$\diamondsuit$ [minor difference] Necessary condition states positive *semi*definiteness of $\nabla_x^2 L(x_*;\lambda^*,\mu^*)$ along certain linear subspace:

$$\forall d \in T = \{d : \overbrace{d^T h_i'(x_*) = 0}^{\forall i \le k}, \overbrace{d^T g_j'(x_*) = 0}^{\forall j \in J(x_*)}\} : d^T \nabla_x^2 L(x_*;\lambda^*,\mu^*) d \ge 0$$

while Sufficient condition requires positive *definiteness* of $\nabla_x^2 L(x_*;\lambda^*,\mu^*)$ along certain linear subspace:

$$\forall 0 \ne d \in T^+ = \{d : \overbrace{d^T h_i'(x_*) = 0}^{\forall i \le k}, \overbrace{d^T g_j'(x_*) = 0}^{\forall j : \lambda_j^* > 0}\} : d^T \nabla_x^2 L(x_*;\lambda^*,\mu^*) d > 0$$

$\diamondsuit$ [major difference] *The linear subspaces in question are different, and* $T \subset T^+$ (i.e., to be positive definite on T^+ is, in general, much more restrictive that to be positive definite on T). The subspaces T, T^+ are equal to each other iff *no* active at x_* inequality constraints get *zero* Lagrange multipliers λ_j^*, i.e., iff $\lambda_j^* > 0$ for all $j \in J(x_*)$.

Note: "Major difference" does not arise at all if there are no inequality constraints.

Note: The "major difference" is essential, as is shown by example

$$\min_{x_1,x_2}\left\{f(x)=x_2^2-x_1^2 : g_1(x)\equiv x_1\le 0\right\} \qquad [x_*=(0,0)^T]$$

Here the Necessary Second Order Optimality Condition is satisfied "strictly":

$$L(x;\lambda)=x_2^2-x_1^2+\lambda x_1,$$

whence

$$\begin{gathered}\lambda^*=0\Rightarrow\nabla_x L(x_*;\lambda^*)=0,\\ T=\{d:d^Tg_1'(0)=0\}=\{d:d_1=0\},\\ 0\neq d\in T\Rightarrow d^T\nabla_x^2L(x_*;\lambda^*)d=2d_2^2>0\}\end{gathered}$$

while $x_*=0$ is *not* a local solution:

$$0=f(0,0)>-t^2=f(-t,0)\ \&\ (-t,0)^T \text{ is feasible for } t\ge 0.$$

Note: In this example, $T=\{d\in\mathbb{R}^2 : d_1=0\}$, while

$$T^+=\{d\in\mathbb{R}^2 : d \text{ is orthogonal to empty set of vectors}\}=\mathbb{R}^2,$$

and $d^T\nabla_x^2L(x_*,\lambda^*)d=2(d_2^2-d_1^2)$ is *not* positive definite on T^+.

$\Rightarrow$ *Sufficient Second Order Condition for local optimality fails to be true.*

Under the circumstances it is a must – x_* is not locally optimal!. However, it makes no harm to "see by eyes" that what must fail, indeed fails...

Intrinsic source of the difficulty: the "true" second order sufficient condition for local optimality of a regular solution x_* states that the Hessian $\nabla^2 L(x_*; \lambda^*, \mu^*)$ of the Lagrange function (with Lagrange multipliers set to what they should be according to the First Order part of necessary Optimality condition) should produce a quadratic form $d^T \nabla^2 L(x_*; \lambda^*, \mu^*) d$ which *should be positive at every nonzero point d from a properly defined polyhedral cone.*

However: There is no easily verifiable necessary and sufficient condition for a quadratic form to be positive at the interior of a polyhedral cone, even as simple as the nonnegative orthant. As a result, in the Sufficient Second Order Optimality Condition the positivity at nonzero points outside of the origin of *a cone* is strengthened to positivity outside of the origin *on the linear span* of this cone.

Optimality Conditions: Simple and Instructive Illustration

♠ Let us look at optimality conditions in the situation where

• **The problem** is

$$\min_{x\in\mathbb{R}^n}\{f(x) : g_j(x) := -x_j \le 0, i \le n\} \Leftrightarrow \min_x\{f(x) : x \ge 0\}$$

• **The feasible solution** we want to check for optimality is $x_* = 0$

Note:

• $x_* = 0$ is regular solution: the gradients $-e_j$ (e_j are the standard basic orths) of all constraints which are active at x_* (these are all our constraints!) are linearly independent.

• The Lagrange function is $L(x,\lambda) = f(x) + \sum_j \lambda_j g_j(x) = f(x) - \sum_j \lambda_j x_j$

$$\Rightarrow \left\{ \begin{array}{lcll} \nabla_x L(x_*,\lambda) & = & \nabla f(0) - [\lambda_1; ...; \lambda_n] & (a) \\ \nabla_x^2 L(x_*,\lambda) & = & \nabla^2 f(0) & (b) \end{array} \right.$$

♠ An evident necessary condition for $x_* = 0$ to be locally optimal is $\frac{\partial f(0)}{\partial x_j} \ge 0\, \forall j$.

Indeed, if, say, $\frac{\partial f(0)}{\partial x_1} < 0$, then passing from x_* to $x_* + te_1$ with small positive t we stay feasible and strictly reduce f, which is impossible when $x_* = 0$ is locally optimal.

$\Rightarrow$ *Setting* $\lambda_j^* = \frac{\partial f(0)}{\partial x_j} \ge 0$ *and invoking* (a), *we get* $\nabla_x L(x_*,\lambda^*) = 0$ *and* $\lambda_j^* g_j(x_*) = 0$, *as required by First Order Necessary Optimality Conditions.*

$$\min_{x\in\mathbb{R}^n}\{f(x) : g_j(x) := -x_j \le 0, i \le n\} \Leftrightarrow \min_x\{f(x) : x \ge 0\}$$
$$\Rightarrow \left\{ \begin{array}{lcll} \nabla_x L(x_*,\lambda) & = & \nabla f(0) - [\lambda_1;...;\lambda_n] & (a) \\ \nabla_x^2 L(x_*,\lambda) & = & \nabla^2 f(0) & (b) \end{array} \right.$$

♠ Assume that $\lambda_j^* := \frac{\partial f(0)}{\partial x_j} \ge 0$, so that $x_* = 0$ satisfies the First Order Necessary Optimality Condition. *What about Second Order Optimality Conditions?*

♡ The "second order" part of the *Necessary Second Order Optimality Condition* says that $d^T\nabla_x^2 L(x_*,\lambda^*)d \equiv d^T\nabla^2 f(0)d$ *should be nonnegative for all d orthogonal to the gradients of all active at $x_* = 0$ constraints.*
In our example this condition is trivially valid: the only d orthogonal to all e_j is $d = 0$.

$$\min_{x\in\mathbb{R}^n}\{f(x) : g_j(x) := -x_j \le 0, i \le n\} \Leftrightarrow \min_x\{f(x) : x \ge 0\}$$

$$\Rightarrow \left\{ \begin{array}{lcll} \nabla_x L(x_*, \lambda) & = & \nabla f(0) - [\lambda_1; ...; \lambda_n] & (a) \\ \nabla_x^2 L(x_*, \lambda) & = & \nabla^2 f(0) & (b) \end{array} \right.$$

♡ The "second order" part of the *Sufficient Second Order Optimality Condition* guarantees local optimality of $x_* = 0$ provided that $d^T \nabla_x^2 L(x_*, \lambda^*) d \equiv d^T \nabla^2 f(0) d > 0$ *for all* $d \neq 0$ *orthogonal to the gradients of those active at* $x_* = 0$ *constraints* g_j *for which* $\lambda_j^* > 0$.

What it actually says, it depends on Lagrange multipliers $\lambda_j^* = \frac{\partial f(0)}{\partial x_j}$. Consider two "extreme" cases:

— [nondegenerate] $\lambda_j^* > 0$ for all j, i.e. $\nabla f(0) > 0$

— [fully degenerate] $\lambda_j^* = 0$ for all j, i.e., $\nabla f(0) = 0$

$$\min_{x\in\mathbb{R}^n}\{f(x) : g_j(x) := -x_j \le 0, i \le n\} \Leftrightarrow \min_x\{f(x) : x \ge 0\}$$

$$\Rightarrow \left\{ \begin{array}{lcll} \nabla_x L(x_*,\lambda) & = & \nabla f(0) - [\lambda_1;...;\lambda_n] & (a) \\ \nabla_x^2 L(x_*,\lambda) & = & \nabla^2 f(0) & (b) \end{array} \right.$$

The "second order" part of the *Sufficient Second Order Optimality Condition* guarantees local optimality of $x_* = 0$ provided that

(!) $d^T\nabla_x^2 L(x_*,\lambda^*)d \equiv d^T\nabla^2 f(0)d > 0$ *for all* $d \neq 0$ *orthogonal to the gradients of those active at* $x_* = 0$ *constraints* g_j *for which* $\lambda_j^* > 0$.

• In the nondegenerate case $\lambda_j^* \equiv \frac{\partial f(0)}{\partial x_j} > 0\,\forall j$, and (!) is trivially valid: *all* g_j get positive Lagrange multipliers, and the only d "orthogonal to the gradients of those..." is $d = 0$.

$\Rightarrow$ *In the nondegenerate case,* $x_* = 0$ *is locally optimal whatever* $\nabla_x^2 L(x_*,\lambda) = \nabla^2 f(0)$ *is.* This indeed is true: with $\nabla f(0) > 0$ and the feasible set $\mathbb{R}^n_+$, *already the first order information on* f *ensures that* $x_* = 0$ *is locally optimal.*

• In the fully degenerate case, the validity of (!) means that $d^T\nabla^2 f(0)d > 0$ *for all* $d \neq 0$. Since in this case $\nabla f(0) = 0$, our Sufficient Second Order Optimality Condition becomes Sufficient Second Order Optimality Condition for $x_* = 0$ to be *unconstrained* local minimizer of f – much more than we actually need!

$$\min_{x\in\mathbb{R}^n}\{f(x) : g_j(x) := -x_j \le 0, i \le n\} \Leftrightarrow \min_x\{f(x) : x \ge 0\}$$

♡ In the fully degenerate case $\nabla f(0) = 0$, the "ideal" Second Order Optimality conditions for $x_* = 0$ to be locally optimal are as follows:

— Necessary: The quadratic function $d^T\nabla^2 f(0)d$ is nonnegative for $d \in \mathbb{R}^n_+$

— Sufficient: The quadratic function $d^T\nabla^2 f(0)d$ is positive for all nonzero $d \in \mathbb{R}^n_+$

However: There is no easy-to-verify way to express the fact that a quadratic form d^TQd is nonnegative on the nonnegative orthant.

$\Rightarrow$ To make Second Order Optimality conditions verifiable, we sacrifice their tightness. Specifically, in the fully degenerate case of our problem,

— the ideal Necessary optimality condition is weakened to the trivially true requirement $d^T\nabla^2 f(0)d \ge 0$ for $d = 0$

— the ideal Sufficient optimality condition is strengthened to $d^T\nabla^2 f(0)d > 0$ for all $d \ne 0$

Proof of Sufficient Second Order Optimality Condition.

1^0. As in the case of Second Order Necessary Optimality Condition, we can reduce the situation to one where

◇ All inequality constraints are active at x_*

◇ The problem is of the special form

$$\min_y \left\{ \phi(y) : \begin{array}{l} y_j \leq 0,\ j \leq m \\ y_{m+i} = 0,\ i \leq k \end{array} \right\} \qquad (P')$$

$$\min_y \left\{ \phi(y) : \begin{array}{l} y_j \le 0,\ j \le m \\ y_{m+i} = 0,\ i \le k \end{array} \right\} \qquad (P')$$

2⁰. In the case of (P'), Sufficient condition reads: $\exists \lambda^* \ge 0, \mu^*$:

$$\begin{array}{c} \nabla_y \big|_{y=0} \left\{ \phi(y) + \sum_{j=1}^{m} \lambda_j^* y_j + \sum_{i=1}^{k} \mu_i^* y_{m+i} \right\} = 0 \\ d_j = 0, j \in J, d \ne 0 \Rightarrow d^T \nabla^2 \phi(0) d > 0 \\ \left[J = \{ j \le m : \lambda_j^* > 0 \} \cup \{ m+1, ..., m+k \} \right] \end{array} \qquad (*)$$

Assuming w.l.o.g. $\{j : \lambda_j^* > 0\} = \{1, ..., q\}$, $(*)$ reads:

$$\begin{array}{rcl} \frac{\partial \phi(0)}{\partial y_\ell} & < & 0,\ \ell = 1, ..., q; \quad \frac{\partial \phi(0)}{\partial y_\ell} = 0,\ \ell = q+1, ..., m \\ \frac{\partial \phi(0)}{\partial y_\ell} & = & 0,\ \ell = m+k+1, ..., n \\ 0 \ne d & \in & T^+ = \{ d : d_\ell = 0,\ \ell \in \{1, ..., q, m+1, ..., m+k\} \} : \\ & \Rightarrow & d^T \nabla^2 \phi(0) d > 0 \end{array}$$

Our goal is to derive from this assumption local optimality of $y_* = 0$ for (P').

3⁰. The feasible set of (P') is the closed cone

$$K = \{ d : d_\ell \le 0,\ \ell = 1, ..., m, d_\ell = 0,\ \ell = m+1, ..., m+k \}$$

Lemma: *For $0 \ne d \in K$ one has $d^T \nabla \phi(0) \ge 0$ and*

$$d^T \nabla \phi(0) = 0 \Rightarrow d^T \nabla^2 \phi(0) d > 0.$$

Situation:

$$\begin{array}{rcl} \frac{\partial\phi(0)}{\partial y_\ell} & < & 0,\ \ell = 1, ..., q \\ \frac{\partial\phi(0)}{\partial y_\ell} & = & 0,\ \ell = q+1, ..., m \\ \frac{\partial\phi(0)}{\partial y_\ell} & = & 0,\ \ell = m+k+1, ..., n \\ 0 \neq d & \in & T^+ = \{d : d_\ell = 0,\ \ell \in \{1, ..., q, m+1, ..., m+k\}\} : \\ & \Rightarrow & d^T \nabla^2 \phi(0) d > 0 \end{array} \qquad (*)$$

$$K = \{d : d_\ell \leq 0,\ \ell = 1, ..., m, d_\ell = 0,\ \ell = m+1, ..., m+k\} \qquad (**)$$

Claim: *For $0 \neq d \in K$ one has $d^T \nabla\phi(0) \geq 0$ and*

$$d^T \nabla\phi(0) = 0 \Rightarrow d^T \nabla^2 \phi(0) d > 0.$$

Proof: For $d \in K$, we have

$$d^T \nabla\phi(0) = \sum_{\ell=1}^{n} \frac{\partial\phi(0)}{\partial y_\ell} d_\ell$$

By $(*)$ – $(**)$, the first q terms in this sum are nonnegative, and the remaining are 0. Thus, the sum always is ≥ 0. For $d \neq 0$, the only possibility for the sum to vanish is to have $d \in T^+$, and in this case $d^T \phi''(0) d > 0$.

"Graphical" explanation:

$$
\begin{array}{rcl}
\frac{\partial \phi(0)}{\partial y_\ell} & < & 0,\ 1 \le \ell \le q \\
\frac{\partial \phi(0)}{\partial y_\ell} & = & 0,\ q+1 \le \ell \le m \\
\frac{\partial \phi(0)}{\partial y_\ell} & = & ?,\ m+1 \le \ell \le m+k \\
\frac{\partial \phi(0)}{\partial y_\ell} & = & 0,\ m+k+1 \le \ell \le n \\
0 \ne d & \in & T^+ = \left\{ d : \begin{array}{l} d_\ell = 0,\ 1 \le \ell \le q \\ d_\ell = 0,\ m+1 \le \ell \le m+k \end{array} \right\} \\
 & \Rightarrow & d^T \nabla^2 \phi(0) d > 0 \\
\hline
K & = & \left\{ d : \begin{array}{l} d_\ell \le 0,\ 1 \le \ell \le m, \\ d_\ell = 0,\ m+1 \le \ell \le m+k \end{array} \right\}
\end{array}
$$

Thus,

$$
\begin{array}{rcl}
K \ni d & = & [\ \overbrace{\le 0}^{q}\ \ \overbrace{\le 0}^{m-q}\ \ \overbrace{= 0}^{k}\ \ \overbrace{???}^{n-m-k}\ ] \\
\nabla \phi(0) & = & [\ \overbrace{< 0}^{q}\ \ \overbrace{= 0}^{m-q}\ \ \overbrace{???}^{k}\ \ \overbrace{= 0}^{n-m-k}\ ] \\
\hline
\{K \cap [\nabla \phi(0)]^\perp\} \ni d & = & [\ \overbrace{= 0}^{q}\ \ \overbrace{\le 0}^{m-q}\ \ \overbrace{= 0}^{k}\ \ \overbrace{???}^{n-m-k}\ ] \\
\hline
T_+ \ni d & = & [\ \underbrace{= 0}_{q}\ \ \underbrace{???}_{m-q}\ \ \underbrace{= 0}_{k}\ \ \underbrace{???}_{n-m-k}\ ]
\end{array}
$$

We see that

- $d \in K \Rightarrow d^T \nabla \phi(0) \ge 0$
- $0 \ne d \in K$ & $d^T \nabla \phi(0) = 0 \Rightarrow d \in T_+$
  $\Rightarrow d^T \nabla^2 \phi(0) d > 0$

Situation: (P') is the problem

$$\min_{y \in K} \phi(y), \tag{!}$$

K is a closed cone, ϕ is twice continuously differentiable in a neighborhood of the origin and is such that

$$d \in K \Rightarrow d^T \nabla \phi(0) \geq 0$$
$$d \in K \backslash \{0\}, d^T \nabla \phi(0) = 0 \Rightarrow d^T \nabla^2 \phi(0) d > 0$$

Claim: *In the situation in question, 0 is a locally optimal solution to (!).*

Proof: Let $M = \{d \in K : \|d\|_2 = 1\}$, and let $M_0 = \{d \in M : d^T \nabla \phi(0) = 0\}$. Since K is closed, both M and M_0 are compact sets.

We know that $d^T \nabla^2 \phi(0) d > 0$ for $d \in M_0$. Since M_0 is a compact set, there exist a neighborhood V of M_0 and $\alpha > 0$ such that

$$d \in V \Rightarrow d^T \nabla^2 \phi(0) d \geq \alpha.$$

The set $V_1 = M \backslash V$ is compact and $d^T \nabla \phi(0) > 0$ when $d \in V_1$; thus, there exists $\beta > 0$ such that

$$d \in V_1 \Rightarrow d^T \nabla \phi(0) \geq \beta.$$

Situation: K is a cone, and the set $M = \{d \in K : \|d\|_2 = 1\}$ is partitioned into two subsets $V_0 = V \cap M$ and V_1 in such a way that

$$\begin{array}{lcl} d \in V_0 & \Rightarrow & d^T\nabla\phi(0) \geq 0, d^T\nabla^2\phi(0)d \geq \alpha > 0 \\ d \in V_1 & \rightarrow & d^T\nabla\phi(0) \geq \beta > 0 \end{array}$$

Goal: To prove that 0 is local minimizer of ϕ on K, or, which is the same, that

$$\exists r > 0 : \phi(0) \leq \phi(td)\ \forall (d \in M, 0 \leq t \leq r).$$

Proof: Let $d \in M$, $t \geq 0$. When $d \in V_0$, we have

$$\phi(td) - \phi(0) \ \geq\ td^T\nabla\phi(0) + \tfrac{1}{2}t^2 d^T\nabla^2\phi(0)d - t^2 \underbrace{R(t)}_{\substack{\to 0, \\ t\to +0}} \geq \tfrac{1}{2}t^2(\alpha - 2R(t))$$

$$\Rightarrow \exists r_0 > 0 : \qquad \phi(td) - \phi(0) \geq \tfrac{1}{4}t^2\alpha \geq 0\ \forall t \leq r_0$$

When $d \in V_1$, we have

$$\phi(td) - \phi(0) \ \geq\ td^T\nabla\phi(0) + \tfrac{1}{2}t^2 d^T\nabla^2\phi(0)d - t^2 \underbrace{R(t)}_{\substack{\to 0, \\ t\to +0}} \geq \beta t - Ct^2 - t^2 R(t)$$

$$\Rightarrow \exists r_1 > 0 : \qquad \phi(td) - \phi(0) \geq \tfrac{\beta}{2}t \geq 0\ \forall t \leq r_1$$

Thus, $\phi(td) - \phi(0) \geq 0$ for all $t \leq \min[r_0, r_1]$, $d \in M$.

Sensitivity Analysis

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \quad (P)$$

$$\Downarrow$$

$$L(x; \lambda, \mu) = f(x) + \sum_j \lambda_j g_j(x) + \sum_i \mu_i h_i(x)$$

Definition: Let x_* be a feasible solution to (P) such that the functions f, g_j, h_i are $\ell \geq 2$ times continuously differentiable in a neighborhood of x_*.
x_* is called a *nondegenerate locally optimal* solution to (P), if
$\diamondsuit$ x_* is a regular solution (i.e., gradients of active at x_* constraints are linearly independent)
$\diamondsuit$ at x_*, Sufficient Second Order Optimality condition holds: $\exists(\lambda^* \geq 0, \mu^*)$:

$$\nabla_x L(x_*; \lambda^*, \mu^*) = 0 \ \& \ \lambda_j^* g_j(x_*) = 0, \ j = 1, ..., m$$

$$\left. \begin{array}{l} d^T \nabla g_j(x_*) = 0 \, \forall (j : \lambda_j^* > 0) \\ d^T \nabla h_i(x_*) = 0 \, \forall i \\ d \neq 0 \end{array} \right\} \Rightarrow d^T \nabla_x^2 L(x_*; \lambda^*, \mu^*) d > 0$$

$\diamondsuit$ for all active at x_* inequality constraints, Lagrange multipliers are positive:

$$g_j(x_*) = 0 \Rightarrow \lambda_j^* > 0.$$

$$\min_x \left\{ f(x) : \begin{array}{c} (g_1(x), g_2(x), ..., g_m(x)) \leq 0 \\ (h_1(x), ..., h_k(x)) = 0 \end{array} \right\} \quad (P)$$

Theorem: *Let x_* be a nondegenerate locally optimal solution to (P). Let us embed (P) into the parametric family of problems*

$$\min_x \left\{ f(x) : \begin{array}{c} g_1(x) \leq a_1, ..., g_m(x) \leq a_m \\ h_1(x) = b_1, ..., h_k(x) = b_k \end{array} \right\} \qquad (P[a,b])$$

so that (P) is $(P[0,0])$.
There exists a neighborhood V_x of x_* and a neighborhood $V_{a,b}$ of the point $a = 0, b = 0$ in the space of parameters a, b such that
$\diamondsuit$ *$\forall (a,b) \in V_{a,b}$, in V_x there exists a unique KKT point $x_*(a,b)$ of $(P[a,b])$, and this point is a nondegenerate locally optimal solution to $(P[a,b])$; moreover, $x_*(a,b)$ is optimal solution for the optimization problem*

$$\mathrm{Opt}_{\mathrm{loc}}(a,b) = \min_x \left\{ f(x) : \begin{array}{c} g_1(x) \leq a_1, ..., g_m(x) \leq a_m \\ h_1(x) = b_1, ..., h_k(x) = b_k \\ x \in V_x \end{array} \right\} \qquad (P_{\mathrm{loc}}[a,b])$$

$\diamondsuit$ *both $x_*(a,b)$ and the corresponding Lagrange multipliers $\lambda^*(a,b)$, $\mu^*(a,b)$ are $\ell - 1$ times continuously differentiable functions of $(a,b) \in V_{a,b}$, and*

$$\frac{\partial \mathrm{Opt}_{\mathrm{loc}}(a,b)}{\partial a_j} = \frac{\partial f(x_*(a,b))}{\partial a_j} = -\lambda_j^*(a,b)$$
$$\frac{\partial \mathrm{Opt}_{\mathrm{loc}}(a,b)}{\partial b_i} = \frac{\partial f(x_*(a,b))}{\partial b_i} = -\mu_i^*(a,b)$$

Simple example: Existence of Eigenvalue

♣ Consider optimization problem

$$\mathrm{Opt} = \min_{x\in\mathbb{R}^n} \left\{ f(x) = x^T A x : h(x) := 1 - x^T x = 0 \right\} \qquad (P)$$

where $A = A^T$ is an $n \times n$ matrix. The problem clearly is solvable. Let x_* be its optimal solution. What can we say about x_*?

Claim: x_* is a regular solution to (P).

Indeed, we should prove that the gradients of active at x_* constraints are linearly independent. There is only one constraint, and its gradient at the feasible set is nonzero.

• Since x_* is a regular globally (and therefore locally) optimal solution, at x_* the Necessary Second Order Optimality condition should hold: $\exists \mu^*$:

$$\nabla_x \overbrace{\left[x^T A x + \mu^*(1 - x^T x)\right]}^{L(x;\mu^*)} = 0 \Leftrightarrow 2(A - \mu^* I)x_* = 0$$

$$\underbrace{d^T \nabla_x h(x_*) = 0}_{\Leftrightarrow d^T x_* = 0} \Rightarrow \underbrace{d^T \nabla_x^2 L(x_*; \mu^*) d \geq 0}_{\Leftrightarrow d^T (A - \mu^* I) d \geq 0}$$

$$\text{Opt} = \min_{x\in\mathbb{R}^n} \left\{ f(x) = x^T A x : h(x) := 1 - x^T x = 0 \right\} \qquad (P)$$

Situation: If x_* is optimal, then $\exists \mu^*$:

$$\begin{array}{rl} Ax_* = \mu^* x_* & (A) \\ d^T x_* = 0 \Rightarrow d^T(A - \mu^* I)d \geq 0 & (B) \end{array}$$

♣ (A) says that $x_* \neq 0$ is an eigenvector of A with eigenvalue μ^*; in particular, *we see that a symmetric matrix always has a real eigenvector*

♣ (B) along with (A) says that $y^T(A - \mu^* I)y \geq 0$ *for all* y.

Indeed, every $y \in \mathbb{R}^n$ can be represented as $y = tx_* + d$ with $d^T x_* = 0$. We now have

$$\begin{array}{rl} & y^T[A - \mu^* I]y = (tx_* + d)^T[A - \mu^* I](tx_* + d) \\ = & t^2 x_*^T \underbrace{[A - \mu^* I]x_*}_{=0} + 2td^T \underbrace{[A - \mu^* I]x_*}_{=0} \\ & + \underbrace{d^T[A - \mu^* I]d}_{\geq 0} \geq 0 \end{array}$$

$$\text{Opt} = \min_{x \in \mathbb{R}^n} \left\{ f(x) = x^T A x : h(x) := 1 - x^T x = 0 \right\} \qquad (P)$$

Note: In the case in question, Necessary Second Order Optimality conditions can be rewritten equivalently as $\exists \mu^*$:

$$\begin{array}{c} [A - \mu^* I] x_* = 0 \\ y^T [A - \mu^* I] y \geq 0 \, \forall y \end{array} \qquad (*)$$

and are not only necessary, but also sufficient for feasible solution x_* to be globally optimal.

To prove sufficiency, let x_* be feasible, and μ^* be such that $(*)$ holds true. For every feasible solution x, one has

$$0 \leq x^T [A - \mu^* I] x = x^T A x - \mu^* x^T x = x^T A x - \mu^*,$$

whence $x^T A x \geq \mu^*$. For $x = x_*$, we have

$$0 = x_*^T [A - \mu^* I] x_* = x_*^T A x_* - \mu^* x_*^T x_* = x_*^T A x_* - \mu^*,$$

whence $x_*^T A x_* = \mu^*$. Thus, x_* is globally optimal for (P), and μ^* is the optimal value in (P).

Extension: $\mathcal{S}$-Lemma. *Let A, B be symmetric matrices, and let B be such that*

$$\exists \bar{x} : \bar{x}^T B \bar{x} > 0. \qquad (*)$$

Then the inequality

$$x^T A x \geq 0 \qquad (A)$$

is a consequence of the inequality

$$x^T B x \geq 0 \qquad (B)$$

iff (A) is a "linear consequence" of (B): there exists $\lambda \geq 0$ such that

$$x^T [A - \lambda B] x \geq 0 \forall x \quad [\Leftrightarrow A \succeq \lambda B] \qquad (C)$$

that is, (A) is a weighted sum of (B) (weight $\lambda \geq 0$) and identically true inequality (C).

Sketch of the proof: The only nontrivial statement is that "*If* (A) is a consequence of (B), *then* there exists $\lambda \geq 0$ such that ...". To prove this statement, assume that (A) is a consequence of (B).

Situation:

$$\exists \bar{x} : \bar{x}^T B \bar{x} > 0; \ \underbrace{x^T B x \geq 0}_{(B)} \Rightarrow \underbrace{x^T A x \geq 0}_{(A)}$$

Consider optimization problem

$$\text{Opt} = \min_x \left\{ x^T A x : h(x) \equiv 1 - x^T B x = 0 \right\}.$$

Problem is feasible by $(*)$, and Opt ≥ 0. Assume that an optimal solution x_* exists. Then, same as above, x_* is regular, and at x_* the Second Order Necessary condition holds true: $\exists \mu^*$:

$$\nabla_x \big|_{x=x_*} \left[x^T A x + \mu^* [1 - x^T B x] \right] = 0 \Leftrightarrow [A - \mu^* B] x_* = 0$$
$$\underbrace{d^T \nabla_x \big|_{x=x_*} h(x) = 0}_{\Leftrightarrow d^T B x_* = 0} \Rightarrow d^T [A - \mu^* B] d \geq 0$$

We have $0 = x_*^T [A - \mu^* B] x_*$, that is, $\mu_* = \text{Opt} \geq 0$. Representing $y \in \mathbb{R}^n$ as $t x_* + d$ with $d^T B x_* = 0$ (that is, $t = x_*^T B y$), we get

$$y^T [A - \mu^* B] y = t^2 x_*^T \underbrace{[A - \mu^* B] x_*}_{=0}$$
$$+ 2 t d^T \underbrace{[A - \mu^* B] x_*}_{=0} + \underbrace{d^T [A - \mu^* B] d}_{\geq 0} \geq 0,$$

Thus, $\mu^* \geq 0$ and $y^T [A - \mu^* B] y \geq 0$ for all y, Q.E.D.

Part II

Continuous Optimization: Basic Algorithms

Lecture 8:

Introduction to Optimization Algorithms

Introduction to Optimization Algorithms

♣ **Goal:** Approximate numerically solutions to Mathematical Programming problems

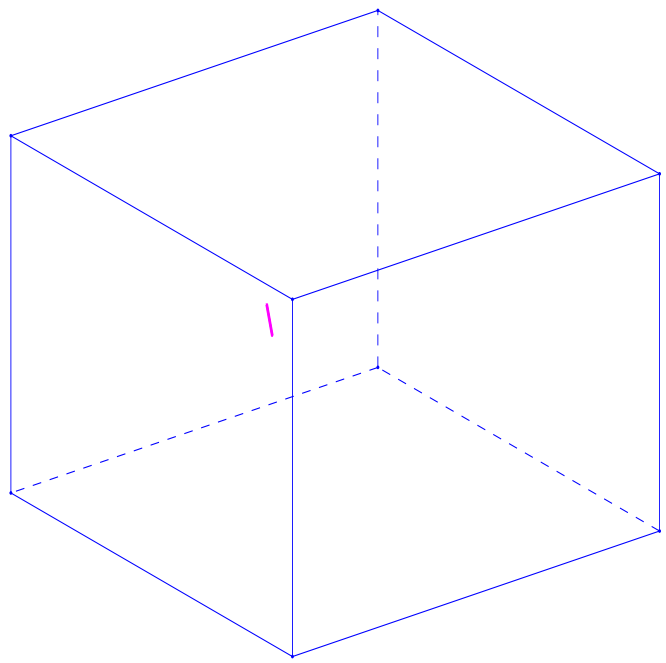
$$\min_x \left\{ f(x) : \begin{array}{l} g_j(x) \leq 0,\ j = 1, ..., m \\ h_i(x) = 0,\ i = 1, ..., k \end{array} \right\} \qquad (P)$$

♣ **Traditional MP algorithms** to be considered in the Course do *not* assume the analytic structure of (P) to be known in advance (and do not know how to use the structure when it is known). These algorithms are *black-box-oriented*: when solving (P), method generates a sequence of *iterates* x_1, x_2,... in such a way that x_{t+1} *depends solely on* **local** *information of* (P) *gathered along the preceding iterates* $x_1, ..., x_t$.
Information on (P) obtained at x_t usually is comprised of the values and the first and the second derivatives of the objective and the constraints at x_t.

How difficult it is to find a needle in haystack?

♣ In some cases, local information, available to black-box-oriented algorithms, is really poor, so that approximating *global* solution to the problem becomes seeking needle in *multidimensional* haystack.

♣ Let us look at a 3D haystack with 2 m edges, and let a needle be a cylinder of height 20 mm and radius of cross-section 1 mm;



Haystack and the needle

How to find the needle in the haystack?

♣ **Optimization setting:** *We want to minimize a smooth function f which is zero "outside of the needle" and negative inside it.*

Note: *When only local information on the function is available*, we get trivial information until the sequence of iterates we are generating hits the needle.

$\Rightarrow$ As a result, it is easy to show that *the number of iterations needed to hit the needle with a reasonable confidence cannot be much smaller than when generating the iterates at random.* In this case, *the probability for an iterate to hit a needle is as small as* $7.8 \cdot 10^{-9}$, that is, to find the needle with a reasonable confidence, we need to generate *hundreds of millions* of iterates.

♠ *As the dimension of the problem grows, the indicated difficulties are dramatically amplified.* For example, *preserving the linear sizes of the haystack and the needle and increasing the dimension of the haystack from 3 to 20, the probability for an iterate to hit the needle becomes as small as* $8.9 \cdot 10^{-67}$ *!*

♣ In the "needle in the haystack" problem it is easy to find a *locally optimal* solution. However, slightly modifying the problem, we can make the latter task disastrously difficult as well.

• In *unconstrained minimization, it is not too difficult to find a point where the gradient of the objective becomes small*, i.e., where the First Order Necessary Optimality condition is "nearly" satisfied.

• In *constrained minimization, it could be disastrously difficult to find just a feasible solution....*

♠ **However:** *The classical algorithms of Continuous Optimization, while providing no meaningful guarantees in the worst case, are capable to process quite efficiently typical optimization problems arising in applications.*

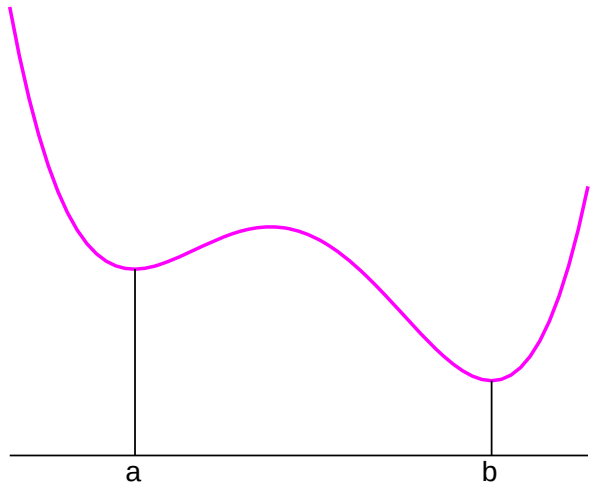
♠ **<u>Note:</u>** *In optimization, there exist algorithms which do exploit problem's structure and allow to approximate the global solution in a reasonable time.* Traditional methods of this type – Simplex method and its variations – do not go beyond *Linear Programming* and *Linearly Constrained Convex Quadratic Programming*.

In 1990's, *new efficient ways to exploit problem's structure were discovered (Interior Point methods). The resulting algorithms, however, do not go beyond Convex Programming.*

♣ *Except for very specific and relatively simple problem classes*, like Linear Programming or Linearly Constrained Quadratic Programming, *optimization algorithms cannot guarantee finding exact solution – local or global – in finite time*. The best we can expect from these algorithms is *convergence* of approximate solutions generated by algorithms to the exact solutions.

♠ Even in the case when "finite" solution methods do exist (Simplex method in Linear Programming), no reasonable complexity bounds for these methods are known, therefore in reality the ability of a method to generate the exact solution in finitely many steps is neither necessary, nor sufficient to justify the method.

♣ *Aside of Convex Programming*, traditional optimization methods are unable to *guarantee* convergence to a *globally optimal* solution. Indeed, in the non-convex case there is no way to conclude from *local* information whether a given point is/is not globally optimal:



"looking" at problem around a, we get absolutely no hint that the true global optimal solution is b.

♠ In order to *guarantee* approximating *global* solution, it seems unavoidable to "scan" a dense set of the values of x in order to be sure that the globally optimal solution is not missed. Theoretically, such a possibility exists; however, the complexity of "exhaustive search" methods blows up exponentially with the dimension of the decision vector, which makes these methods completely impractical.

♣ Traditional optimization methods do *not* incorporate exhaustive search and, as a result, can*not* guarantee convergence to a global solution.

♠ A typical theoretical result on a traditional optimization method as applied to a general (not necessary convex) problem sounds like:

> *Assume that problem (P) possesses the following properties:*
>
> *...*
>
> *...*
>
> *Then the sequence of approximate solutions generated by method X is bounded, and all its limiting points are KKT points of the problem.*

or

> *Assume that x_* is a nondegenerate local solution to (P). Then method X, started close enough to x_*, converges to x_*.*

Classification of MP Algorithms

♣ There are two major traditional classifications of MP algorithms:

♢ *Classification by application fields,* primarily into

- algorithms for unconstrained optimization
- algorithms for constrained optimization

♢ *Classification by information* used by the algorithms, primarily into

- zero order methods which use only the values of the objective and the constraints
- first order methods (use both values and first order derivatives)
- second order methods (use values, first- and second order derivatives).

Rate of Convergence of MP Algorithm

$$\min_x \left\{ f(x) : \begin{array}{l} g_j(x) \leq 0,\ j = 1, ..., m \\ h_i(x) = 0,\ i = 1, ..., k \end{array} \right\} \qquad (P)$$

♣ There is a necessity to quantify the convergence properties of MP algorithms. Traditionally, this is done via *asymptotical rate of convergence* defined as follows:

Step 1. We introduce an appropriate *error measure of a candidate solution* x – a nonnegative function $\mathrm{Error}_P(x)$ which is zero exactly at the set X_* of solutions to (P) we intend to approximate.

Examples: (i) Distance to the set X_*:

$$\mathrm{Error}_P(x) = \inf_{x_* \in X_*} \|x - x_*\|_2$$

(ii) Residual in terms of the objective and the constraints

$$\mathrm{Error}_P(x) = \max \left[f(x) - \mathrm{Opt}(P), [g_1(x)]_+, ..., [g_m(x)]_+, |h_1(x)|, ..., |h_k(x)| \right]$$

$$[a_* = \max[a, 0]]$$

<u>Step 2.</u> Assume that we have established *convergence* of our method, that is, we know that if x_t^* are approximate solutions generated in t steps by the method as applied to a problem (P) from a given family, then

$$\mathrm{Error}_P(t) \equiv \mathrm{Error}_P(x_t^*) \to 0,\ t \to \infty$$

We then roughly quantify the *rate* at which the sequence $\mathrm{Error}_P(t)$ of nonnegative reals converges to 0. Specifically, we say that

$\diamondsuit$ the method converges *sublinearly*, if the error goes to zero less rapidly than a geometric progression, e.g., as $1/t$ or $1/t^2$;

$\diamondsuit$ the method converges *linearly*, if there exist $C < \infty$ and $q \in (0,1)$ such that

$$\mathrm{Error}_{(P)}(t) \le Cq^t$$

q is called the *convergence ratio.* E.g.,

$$\mathrm{Error}_P(t) \asymp e^{-at}$$

exhibits linear convergence with ratio e^{-a}.

<u>Sufficient condition</u> for linear convergence with ratio $q \in (0,1)$ is that

$$\overline{\lim_{t\to\infty}} \frac{\mathrm{Error}_P(t+1)}{\mathrm{Error}_P(t)} < q$$

◇ the method converges *superlinearly*, if the sequence of errors converges to 0 faster than every geometric progression:

$$\forall q \in (0,1) \exists C : \text{Error}_P(t) \leq C q^t$$

For example,

$$\text{Error}_P(t) \asymp e^{-at^2}$$

corresponds to superlinear convergence.

Sufficient condition for superlinear convergence is

$$\lim_{t \to \infty} \frac{\text{Error}_P(t+1)}{\text{Error}_P(t)} = 0$$

◇ the method exhibits convergence of order $p > 1$, if

$$\exists C : \text{Error}_P(t+1) \leq C \left(\text{Error}_P(t)\right)^p$$

Convergence of order 2 is called *quadratic*. For example,

$$\text{Error}_P(t) \asymp e^{-ap^t}$$

converges to 0 with order p.

Informal explanation: When the method converges, $\text{Error}_P(t)$ goes to 0 as $t \to \infty$, that is, eventually the decimal representation of $\text{Error}_P(t)$ has zero before the decimal dot and more and more zeros after the dot; the number of zeros following the decimal dot is called *the number of accuracy digits in the corresponding approximate solution*. Traditional classification of rates of convergence is based on *how many steps, asymptotically, is required to add a new accuracy digit to the existing ones.*

$\diamondsuit$ With *sublinear* convergence, the "price" of accuracy digit grows with the position of the digit. For example, with rate of convergence $O(1/t)$ every new accuracy digit is 10 times more expensive, in terms of # of steps, than its predecessor.

◇ With *linear* convergence, every accuracy digit has the same price, proportional to $\dfrac{1}{\ln\left(\dfrac{1}{\text{convergence ratio}}\right)}$. Equivalently: every step of the method adds a fixed number r of accuracy digits (for q not too close to 0, $r \approx 1 - q$);

◇ With *superlinear* convergence, every subsequent accuracy digit eventually becomes cheaper than its predecessor – the price of accuracy digit goes to 0 as the position of the digit grows. Equivalently, every additional step adds more and more accuracy digits.

◇ With convergence of order $p > 1$, the price of accuracy digit not only goes to 0 as the position k of the digit grows, but does it rapidly enough – in a geometric progression. Equivalently, eventually every additional step of the method *multiplies by* p the number of accuracy digits.

♣ With the traditional approach, the convergence properties of a method are the better the higher is the "rank" of the method in the above classification. Given a family of problems, traditionally it is thought that linearly converging on every problem of the family method is faster than a sublinearly converging, superlinearly converging method is faster than a linearly converging one, etc.

♣ **<u>Note:</u>** Usually we are able to *prove existence* of parameters C and q quantifying linear convergence:

$$\mathrm{Error}_P(t) \leq Cq^t$$

or convergence of order $p > 1$:

$$\mathrm{Error}_P(t+1) \leq C(\mathrm{Error}_P(t))^p,$$

but are unable to find numerical values of these parameters – they may depend on "unobservable" characteristics of a particular problem we are solving. As a result, traditional "quantification" of convergence properties is *qualitative* and *asymptotical*.

Solvable Case of MP – Convex Programming

♣ We have seen that *as applied to general MP programs*, optimization methods have a number of severe *theoretical* limitations, including the following major ones:

♢ Unless exhaustive search (completely unrealistic in high-dimensional optimization) is used, there are no guarantees of approaching *global* solution

♢ Quantification of convergence properties is of asymptotical and qualitative character. As a result, the most natural questions like:

> *We should solve problems of such and such structure with such and such sizes and the data varying in such and such ranges. How many steps of method X are sufficient to solve problems within such and such accuracy?*

usually do not admit theoretically valid answers.

♣ In spite of their *theoretical* limitations, *in reality* traditional MP algorithms allow to solve many, if not all, MP problems of real-world origin, including those with many thousands of variables and constraints.

♣ Moreover, there exists a "solvable case" when practical efficiency admits solid theoretical guarantees – the case of Convex Programming.

- Here is a typical "Convex Programming" result:

Assume we are solving a Convex Programming program

$$\mathrm{Opt} = \min_x \left\{ f(x) : g_j(x) \leq 0,\, j \leq m, |x_i| \leq 1, i \leq n \right\}.$$

where the objective and the constraints are normalized by the requirement

$$|x_i| \leq 1,\, i \leq n \Rightarrow\ |f(x)| \leq 1,\, |g_j(x)| \leq 1,\, j \leq m$$

Given $\epsilon \in (0,1)$*, one can find an* ϵ*-solution* x^ϵ *to the problem:*

$$\underbrace{|x_i^\epsilon| \leq 1}_{\forall i \leq n}\ \&\ \underbrace{g_j(x^\epsilon) \leq \epsilon}_{\forall j \leq m}\ \&\ f(x^\epsilon) - \mathrm{Opt} < \epsilon$$

in no more than

$$2n^2 \ln\left(\frac{2n}{\epsilon}\right)$$

steps, with a single computation of the values and the first order derivatives of $f, g_1, ..., g_m$ *at a point and* $100(m+n)n$ *additional arithmetic operations per step.*

Line Search

♣ Line Search is a common name for techniques for *one-dimensional* "simply constrained" optimization, specifically, for problems

$$\min_x \{f(x) : a \leq x \leq b\},$$

where $[a, b]$ is a given segment on the axis (sometimes, we shall allow for $b = +\infty$), and f is a function which is at least once continuously differentiable on (a, b) and is continuous at the segment $[a, b]$ (on the ray $[a, \infty)$, if $b = \infty$).

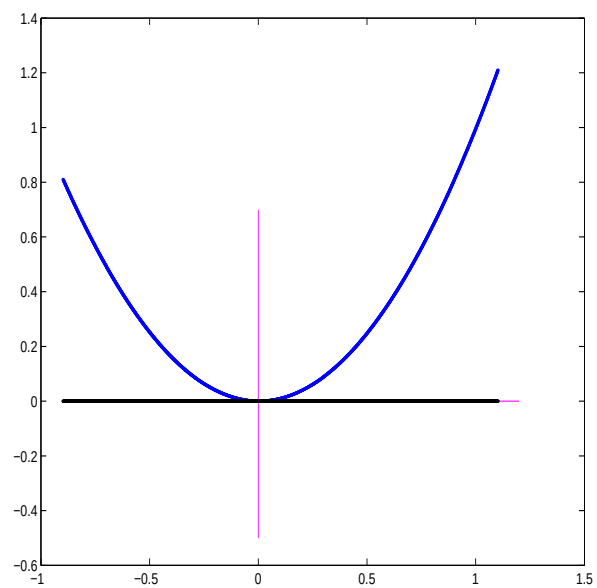
♣ Line search is used, as a subroutine, in many algorithms for multi-dimensional optimization.

$$\min_{a \le x \le b} f(x) \qquad (P)$$

♣ **Zero-order line search.** In zero-order line search one uses the values of the objective f in (P) and does not use its derivatives.

♠ To ensure well-posedness of the problem, assume that the objective is *unimodal*, that is, possesses a unique local minimizer x_* on $[a, b]$.

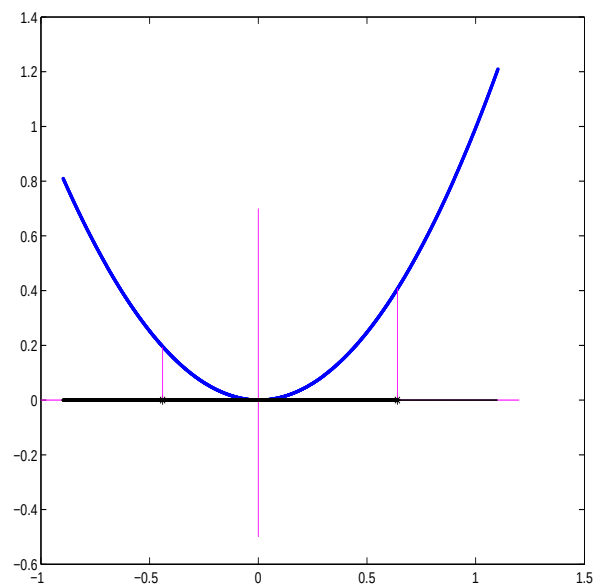
Equivalently: There exists a unique point $x_* \in [a, b]$ such that $f(x)$ strictly decreases on $[a, x_*]$ and strictly increases on $[x_*, b]$:



Main observation: Let f be unimodal on $[a, b]$, and assume we know $f(x')$, $f(x'')$ for certain x', x'' with

$$a < x' < x'' < b.$$

$\diamondsuit$ If $f(x'') \geq f(x')$, then $f(x) > f(x'')$ for $x > x''$, so that the minimizer belongs to $[a, x'']$:



$\diamondsuit$ Similarly, if $f(x'') < f(x')$, then $f(x) > f(x')$ when $x < x'$, so that the minimizer belongs to $[x', b]$.

♠ In both cases, two computations of f at x', x'' allow to reduce the initial "search domain" to a smaller one ($[a, x'']$ or $[x', b]$).

♣ Choosing x', x'' so that they split $[a_0, b_0] = [a, b]$ into three equal segments, computing $f(x')$, $f(x'')$ and comparing them to each other, we can build a new segment $[a_1, b_1] \subset [a_0, b_0]$ such that

♢ the new segment is a *localizer* – it contains the solution x_*

♢ the length of the new localizer is 2/3 of the length of the initial localizer $[a_0, b_0] = [a, b]$.

♠ On the new localizer, same as on the original one, the objective is unimodal, and we can iterate our construction.

♠ In $N \geq 1$ steps ($2N$ computations of f), we shall reduce the size of localizer by factor $(2/3)^N$, that is, we get *linearly converging*, in terms of the argument, algorithm with the convergence ratio

$$q = \sqrt{2/3} = 0.8165...$$

Can we do better ? - YES!

$$\left.\begin{array}{c}[a_{t-1},b_{t-1}]\\ x'_t<x''_t\end{array}\right\}\Rightarrow f(x'_t),f(x''_t)\Rightarrow\left\{\begin{array}{l}[a_t,b_t]=[a_{t-1},x''_t]\\ [a_t,b_t]=[x'_t,b_{t-1}]\end{array}\right.$$

♣ Observe that one of two points at which we compute f at a step becomes the endpoint of the new localizer, while the other one is an interior point of this localizer, *and therefore we can use it as the one of two points where f should be computed at the next step!*
With this approach, only the very first step costs 2 function evaluations, while the subsequent steps cost just 1 evaluation each!
♠ Let us implement the idea in such a way that all search points will divide respective localizers in a fixed proportion:

$$x'-a=b-x''=\theta(b-a)$$

The proportion is given by the equation

$$\theta\equiv\frac{x'-a}{b-a}=\frac{x''-x'}{b-x'}\equiv\frac{1-2\theta}{1-\theta}\Rightarrow\theta=\frac{3-\sqrt{5}}{2}.$$

$$\frac{\text{red}}{\text{red}+\text{red}+\text{blue}} = \frac{\text{blue}}{\text{red}+\text{blue}}$$

Golden Search

♣ We have arrived at *golden search*, where the search points x_{t-1}, x_t of step t are placed in the current localizer $[a_{t-1}, b_{t-1}]$ according to

$$\frac{x' - a}{b - a} = \frac{b - x''}{b - a} = \frac{3 - \sqrt{5}}{2}$$

In this method, a step reduces the error (the length of localizer) by factor $1 - \frac{3-\sqrt{5}}{2} = \frac{\sqrt{5}-1}{2}$. The convergence ratio is about

$$\frac{\sqrt{5} - 1}{2} \approx 0.6180...$$

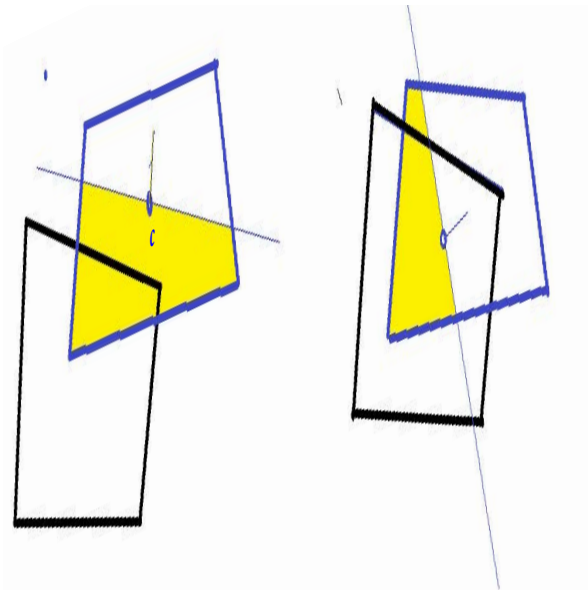
$$\min_x \{f(x) : a \le x \le b\},$$

♣ **<u>First order line search: Bisection.</u>** Assume that f is differentiable on (a, b) and *strictly unimodal*, that is, it is unimodal, $x_* \in (a, b)$ and $f'(x) < 0$ for $a < x < x_*$, $f'(x) > 0$ for $x_* < x < b$.

Let both f and f' be available. In this case the method of choice in *Bisection*.

♠ **<u>Main observation:</u>** Given $x_1 \in [a, b] \equiv [a_0, b_0]$, let us compute $f'(x_1)$.

♢ If $f'(x_1) > 0$, then, from strict unimodality, $f(x) > f(x_1)$ to the right of x_1, thus, x_* belongs to $[a, x_1]$:



♢ Similarly, if $f'(x_1) \leq 0$, then $f(x) > f(x_1)$ for $x < x_1$, and x_* belongs to $[a, x_1]$.

♠ In both cases, we can replace the original localizer $[a, b] = [a_0, b_0]$ with a smaller localizer $[a_1, b_1]$ and then iterate the process.

In Bisection, the point x_t where at step t $f'(x_t)$ is computed, is the midpoint of $[a_{t-1}, b_{t-1}]$, so that every step reduces localizer's length by factor 2.

♣ Clearly, Bisection converges linearly in terms of argument with convergence ratio 0.5:

$$a_t - x_* \leq 2^{-t}(b_0 - a_0).$$

Inexact Line Search

♣ Many algorithms for multi-dimensional minimization use Line Search as a subroutine, in the following way:

♢ given current iterate $x_t \in \mathbb{R}^n$, the algorithm defines a *search direction* $d_t \in \mathbb{R}^n$ which is a direction of decrease of f:

$$d_t^T \nabla f(x_t) < 0.$$

Then Line Search is invoked to minimize the one-dimensional function

$$\phi(\gamma) = f(x_t + \gamma d_t)$$

over $\gamma \geq 0$; the resulting $\gamma = \gamma^t$ defines the stepsize along the direction d_t, so that the new iterate of the outer algorithm is

$$x_{t+1} = x_t + \gamma^t d_t.$$

♠ In many situations of this type, there is no necessity in exact minimization in γ; an "essential" reduction in ϕ is sufficient.

♣ Standard way to define (and to achieve) "essential reduction" is given by *Armijo's rule*:

Let $\phi(\gamma)$ be continuously differentiable function of $\gamma \geq 0$ such that $\phi'(0) < 0$, and let $\epsilon \in (0,1)$, $\eta > 1$ be parameters (popular choice is $\epsilon = 0.2$ and $\eta = 2$ or $\eta = 10$). We say that a stepsize $\gamma > 0$ is *appropriate*, if

$$\phi(\gamma) \leq \phi(0) + \epsilon\gamma\phi'(0), \qquad (*)$$

and is *nearly maximal*, if η times larger step is *not* appropriate:

$$\phi(\eta\gamma) > \phi(0) + \epsilon\eta\gamma\phi'(0). \qquad (**)$$

A stepsize $\gamma > 0$ passes Armijo test (reduces ϕ "essentially"), if it is both appropriate and nearly maximal.

♠ **<u>Fact:</u>** *Assume that ϕ is bounded below on the ray $\gamma > 0$. Then a stepsize passing Armijo rule does exist and can be found efficiently.*

♣ Armijo-acceptable step $\gamma > 0$:

$$\phi(\gamma) \leq \phi(0) + \epsilon\gamma\phi'(0) \qquad (*)$$
$$\phi(\eta\gamma) > \phi(0) + \epsilon\eta\gamma\phi'(0) \quad (**)$$

♣ **Algorithm for finding Armijo-acceptable step:**

Start: Choose $\gamma_0 > 0$ and check whether it passes (*). If YES, go to Branch A, otherwise go to Branch B.

Branch A: γ_0 satisfies $(*)$. Testing subsequently the values $\eta\gamma_0$, $\eta^2\gamma_0$, $\eta^3\gamma_0$,... of γ, stop when the current value for the first time violates $(*)$; the preceding value of γ passes the Armijo test.

Branch B: γ_0 does not satisfy $(*)$. Testing subsequently the values $\eta^{-1}\gamma_0$, $\eta^{-2}\gamma_0$, $\eta^{-3}\gamma_0$,... of γ, stop when the current value for the first time satisfies $(*)$; this value of γ passes the Armijo test.

♣ **Validation of the algorithm:** It is clear that *if the algorithm terminates*, then the result indeed passes the Armijo test. Thus, all we need to verify is that the algorithm eventually terminates.

♢ Branch A clearly is finite: here we test the inequality

$$\phi(\gamma) > \phi(0) + \epsilon\gamma\phi'(0)$$

along the sequence $\gamma_i = \eta^i\gamma_0 \to \infty$, and terminate when this inequality is satisfied for the first time. Since $\phi'(0) < 0$ and ϕ is below bounded, this indeed will eventually happen.

♢ Branch B clearly is finite: here we test the inequality

$$\phi(\gamma) \leq \phi(0) + \epsilon\gamma\phi'(0) \qquad (*)$$

along a sequence $\gamma_i = \eta^{-i}\gamma_0 \to +0$ of values of γ and terminate when this inequality is satisfied for the first time. Since $\epsilon \in (0,1)$ and $\phi'(0) < 0$, this inequality is satisfied for all small enough positive values of γ, since

$$\phi(\gamma) = \phi(0) + \gamma\Big[\phi'(0) + \underbrace{R(\gamma)}_{\to 0, \gamma \to +0}\Big].$$

For large i, γ_i definitely will be "small enough", thus, Branch B is finite.

Lecture 9:

Methods for Unconstrained Minimization

Methods for Unconstrained Minimization

♣ **Unconstrained minimization problem** is

$$f_* = \min_x f(x),$$

where f well-defined and continuously differentiable on the entire $\mathbb{R}^n$.
Note: Most of the constructions to be presented can be straightforwardly extended onto "essentially unconstrained case" where f is continuously differentiable on an **open** domain D in $\mathbb{R}^n$ and is such that the level sets $\{x \in D : f(x) \leq a\}$ are closed.

$$f_* = \min_x f(x) \qquad (P)$$

Gradient Descent

♣ **Gradient Descent** is the simplest first order method for unconstrained minimization.

The idea: Let x be a current iterate which is *not* a critical point of f: $f'(x) \neq 0$. We have

$$f(x+th) = f(x) + th^T f'(x) + t\|h\|_2 R_x(th)$$
$$[R_x(s) \to 0 \text{ as } s \to 0]$$

Since $f'(x) \neq 0$, the unit antigradient direction $g = -f'(x)/\|f'(x)\|_2$ is a direction of decrease of f:

$$\frac{d}{dt}\Big|_{t=0} f(x+tg) = g^T f'(x) = -\|f'(x)\|_2$$

so that shift $x \mapsto x + tg$ along the direction g *locally* decreases f "at the rate" $\|f'(x)\|_2$.

♠ **Note:** As far as local rate of decrease is concerned, g is the best possible direction of decrease: for any other unit direction h, we have

$$\frac{d}{dt}\Big|_{t=0} f(x+th) = h^T f'(x) > -\|f'(x)\|_2.$$

♣ In generic Gradient Descent, we update the current iterate x by a step from x in the antigradient direction which reduces the objective:

$$x_t = x_{t-1} - \gamma_t f'(x_{t-1}),$$

where γ_t are positive stepsizes such that

$$f'(x_{t-1}) \neq 0 \Rightarrow f(x_t) < f(x_{t-1}).$$

♠ **Standard implementations:**

♢ **Steepest GD**:

$$\gamma_t = \operatorname*{argmin}_{\gamma \geq 0} f(x_{t-1} - \gamma f'(x_{t-1}))$$

(slight idealization, except for the case of quadratic f)

♢ **Armijo GD**: $\gamma_t > 0$ is such that

$$f(x_{t-1} - \gamma_t f'(x_{t-1})) \leq \underbrace{f(x_{t-1}) - \epsilon\gamma_t \|f'(x_{t-1})\|_2^2}_{f(x_{t-1}) + \epsilon\gamma_t \frac{d}{d\gamma}\big|_{\gamma=0} f(x_{t-1} - \gamma f'(x_{t-1}))}$$

$$f(x_{t-1} - \eta\gamma_t f'(x_{t-1})) > f(x_{t-1}) - \epsilon\eta\gamma_t \|f'(x_{t-1})\|_2^2$$

$[\epsilon \in (0,1), \eta > 1 :$ fixed parameters$]$

(implementable, provided that $f'(x_{t-1}) \neq 0$ and $f(x_{t-1} - \gamma f'(x_{t-1}))$ is below bounded when $\gamma \geq 0$)

Note: By construction, GD is unable to leave a critical point:

$$f'(x_{t-1}) = 0 \Rightarrow x_t = x_{t-1}.$$

♣ **Global Convergence Theorem:** *Assume that the level set of f corresponding to the starting point x_0:*

$$G = \{x : f(x) \leq f(x_0)\}$$

is compact, and f is continuously differentiable in a neighbourhood of G. Then for both SGD and AGD:

♢ *the trajectory $x_0, x_1, ...$ of the method, started at x_0, is well-defined and never leaves G (and thus is bounded);*

♢ *the method is monotone:*

$$f(x_0) \geq f(x_1) \geq ...$$

and inequalities are strict, unless method reaches a critical point x_t, so that $x_t = x_{t+1} = x_{t+2} = ...$

♢ *Every limiting point of the trajectory is a critical point of f.*

Sketch of the proof: 1^0. If $f'(x_0) = 0$, the method never leaves x_0, and the statements are evident. Now assume that $f'(x_0) \neq 0$. Then the function $\phi_0(\gamma) = f(x_0 - \gamma f'(x_0))$ is below bounded, and the set $\{\gamma \geq 0 : \phi_0(\gamma) \leq \phi_0(0)\}$ is compact along with G, so that $\phi_0(\gamma)$ achieves its minimum on the ray $\gamma \geq 0$, and $\phi_0'(0) < 0$. It follows that the first step of GD is well-defined and $f(x_1) < f(x_0)$. The set $\{x : f(x) \leq f(x_1)\}$ is a closed subset of G and thus is compact, and we can repeat our reasoning with x_1 in the role of x_0, etc. We conclude that the trajectory is well-defined, never leaves G and the objective is strictly decreased, unless a critical point is reached.

2^0. "all limiting points of the trajectory are critical points of f":

<u>Fact:</u> *Let $x \in G$ and $f'(x) \neq 0$. Then there exists $\epsilon > 0$ and a neighbourhood U of x such that for every $x' \in U$ the step $x' \to x'_+$ of the method from x' reduces f by at least ϵ.*

Given Fact, let x be a limiting point of $\{x_i\}$; assume that $f'(x) \neq 0$, and let us lead this assumption to contradiction. By Fact, there exists a neighbourhood U of x such that

$$x_i \in U \Rightarrow f(x_{i+1}) \leq f(x_i) - \epsilon.$$

Since the trajectory visits U infinitely many times and the method is monotone, we conclude that $f(x_i) \to -\infty$, $i \to \infty$, which is impossible, since G is compact, so that f is below bounded on G.

Limiting points of Gradient Descent

♣ Under assumptions of Global Convergence Theorem, limiting points of GD exist, and all of them are critical points of f. What kind of limiting points could they be?

♢ A **nondegenerate maximizer** of f cannot be a limiting point of GD, unless the method is started at this maximizer.

♢ A **saddle point** of f is "highly unlikely" candidate to the role of a limiting point. **Practical** experience says that limiting points are local minimizers of f.

♢ A **nondegenerate global minimizer** x_* of f, if any, is an "attraction point" of GD: when starting close enough to this minimizer, the method converges to x_*.

Rates of convergence

♣ In general, we cannot guarantee more than convergence to the set of critical points of f. A natural error measure associated with this set is

$$\delta^2(x) = \|f'(x)\|_2^2.$$

♠ **<u>Definition:</u>** Let U be an open subset of $\mathbb{R}^n$, $L \geq 0$ and f be a function defined on U. We say that f is $\mathsf{C}^{1,1}(L)$ on U, if f is continuously differentiable in U with locally Lipschitz continuous, with constant L, gradient:

$$[x, y] \in U \Rightarrow \|f'(x) - f'(y)\|_2 \leq L\|x - y\|_2.$$

We say that f is $\mathsf{C}^{1,1}(L)$ on a set $Q \subset \mathbb{R}^n$, if there exists an open set $U \supset Q$ such that f is $\mathsf{C}^{1,1}(L)$ on U.

<u>Note:</u> *Assume that f is twice continuously differentiable on U. Then f is $C^{1,1}(L)$ on U* **<u>iff</u>** *the norm of the Hessian of f does not exceed L:*

$$\forall (x \in U, d \in \mathbb{R}^n) : |d^T f''(x) d| \leq L\|d\|_2^2.$$

<u>Theorem:</u> *In addition to assumptions of Global Convergence Theorem, assume that f is $C^{1,1}(L)$ on $G = \{x : f(x) \le f(x_0)\}$. Then*

◇ *For SGD, one has*

$$\min_{0 \le \tau \le t} \delta^2(x_\tau) \le \frac{2[f(x_0) - f_*]L}{t+1}, \; t = 0, 1, 2, ...$$

◇ *For AGD, one has*

$$\min_{0 \le \tau \le t} \delta^2(x_\tau) \le \frac{\eta}{2\epsilon(1-\epsilon)} \cdot \frac{[f(x_0) - f_*]L}{t+1}, \; t = 0, 1, 2, ...$$

Lemma: *For $x \in G$, $0 \le s \le 2/L$ one has*

$$x - sf'(x) \in G \qquad (1)$$
$$f(x - sf'(x)) \le f(x) - \delta^2(x)s + \tfrac{L\delta^2(x)}{2}s^2, \qquad (2)$$

There is nothing to prove when $g \equiv -f'(x) = 0$. Let $g \neq 0$, $s_* = \max\{s \ge 0 : x + sg \in G\}$, $\delta^2 = \delta^2(x) = g^T g$. The function

$$\phi(s) = f(x - sf'(x)) : [0, s_*] \to \mathbb{R}$$

is continuously differentiable and satisfies

$$(a) \quad \phi'(0) = -g^T g \equiv -\delta^2; \quad (b) \quad \phi(s_*) = f(x_0)$$
$$(c) \quad |\phi'(s) - \phi'(0)| = |g^T[f'(x + sg) - f'(x)]| \le Ls\delta^2$$
$$\text{Therefore } \phi(s) \le \phi(0) - \delta^2 s + \tfrac{L\delta^2}{2}s^2 \qquad (*)$$

which is (2). Indeed, setting

$$\theta(s) = \phi(s) - [\phi(0) - \delta^2 s + \frac{L\delta^2}{2}s^2],$$

we have

$$\theta(0) = 0, \theta'(s) = \phi'(s) - \phi'(0) - Ls\delta^2 \underbrace{\le}_{\text{by } (c)} 0.$$

By $(*)$ and (b), we have

$$f(x_0) \le \phi(0) - \delta^2 s_* + \tfrac{L\delta^2}{2}s_*^2 \le f(x_0) - \delta^2 s_* + \tfrac{L\delta^2}{2}s_*^2$$
$$\Rightarrow s_* \ge 2/L$$

Lemma $\Rightarrow$ **Theorem:** **<u>SGD:</u>** By Lemma, we have

$$f(x_t) - f(x_{t+1}) = f(x_t) - \min_{\gamma \geq 0} f(x_t - \gamma f'(x_t))$$

$$\geq f(x_t) - \min_{0 \leq s \leq 2/L} \left[f(x_t) - \delta^2(x_t) s + \frac{L\delta^2(x_t)}{2} s^2 \right]$$

$$= \frac{\delta^2(x_t)}{2L}$$

$$\Rightarrow f(x_0) - f_* \geq \sum_{\tau=0}^{t} [f(x_\tau) - f(x_{\tau+1})] \geq \sum_{\tau=0}^{t} \frac{\delta^2(x_\tau)}{2L}$$

$$\geq \frac{t+1}{2L} \min_{0 \leq \tau \leq t} \delta^2(x_\tau)$$

$$\Rightarrow \min_{0 \leq \tau \leq t} \delta^2(x_\tau) \leq \frac{2L(f(x_0) - f_*)}{t+1}$$

Lemma $\Rightarrow$ Theorem: **<u>AGD:</u>** **<u>Claim:</u>** $\gamma_{t+1} > \frac{2(1-\epsilon)}{L\eta}$. Indeed, otherwise by Lemma

$$\begin{aligned} f(x_t - \gamma_t \eta f'(x_t)) &\leq f(x_t) - \gamma_{t+1}\eta\delta^2(x_t) + \frac{L\delta^2(x_t)}{2}\eta^2\gamma_{t+1}^2 \\ &= f(x_t) - \underbrace{\left[1 - \frac{L}{2}\eta\gamma_{t+1}\right]}_{\geq \epsilon} \eta\gamma_{t+1}\delta^2(x_t) \\ &\leq f(x_t) - \epsilon\eta\gamma_{t+1}\delta^2(x_t) \end{aligned}$$

which is impossible.

- We have seen that $\gamma_{t+1} > \frac{2(1-\epsilon)}{L\eta}$. By Armijo rule,

$$f(x_t) - f(x_{t+1}) \geq \epsilon\gamma_{t+1}\delta^2(x_t) \geq \frac{2\epsilon(1-\epsilon)}{L\eta}\delta^2(x_t);$$

the rest of the proof is as for SGD.

♣ **Convex case.** In addition to assumptions of Global Convergence Theorem, assume that f is convex.

♠ All critical points of a convex function are its global minimizers

$\Rightarrow$ In Convex case, SGD and AGD converge to the set of global minimizers of f: $f(x_t) \to f_*$ as $t \to \infty$, and all limiting points of the trajectory are global minimizers of f.

♠ In Convex $C^{1,1}(L)$ case, one can quantify the global rate of convergence in terms of the residual $f(x_t) - f_*$:

Theorem. *Assume that the set $G = \{x : f(x) \leq f(x_0)\}$ is convex compact, f is convex on G and $C^{1,1}(L)$ on this set:*

$$\|f'(x) - f'(y)\|_2 \leq L\|x - y\|_2 \; \forall x, y \in G.$$

Consider AGD, and let $\epsilon \geq 0.5$. Then the trajectory of the method converges to a global minimizer x_* of f, and

$$f(x_t) - f_* \leq \frac{\eta L \|x_0 - x_*\|_2^2}{4(1-\epsilon)t}, \; t = 1, 2, ...$$

♣ **Definition:** Let M be a convex set in $\mathbb{R}^n$ and $0 < \ell \leq L < \infty$. A function f is called **strongly convex**, with parameters ℓ, L, on M, if

◇ f is $\mathsf{C}^{1,1}(L)$ on M

◇ for $x, y \in M$, one has

$$[x-y]^T[f'(x) - f'(y)] \geq \ell\|x-y\|_2^2. \qquad (*)$$

The ratio $Q_f = L/\ell$ is called **condition number** of f.

♠ **Comment:** If f is $\mathsf{C}^{1,1}(L)$ on a convex set M, then

$$x, y \in M \Rightarrow |f(y) - [f(x) + (y-x)^T f'(x)]| \leq \tfrac{L}{2}\|x-y\|_2^2.$$

If f satisfies $(*)$ on a convex set M, then

$$\forall x, y \in M : f(y) \geq f(x) + (y-x)^T f'(x) + \frac{\ell}{2}\|y-x\|_2^2.$$

In particular, f is convex on M.

$\Rightarrow$ A strongly convex, with parameters ℓ, L, function f on a convex set M satisfies the relation

$$\begin{aligned}&\forall x, y \in M : f(x) + (y-x)^T f'(x) + \tfrac{\ell}{2}\|y-x\|_2^2 \\ &\leq f(y) \leq f(x) + (y-x)^T f'(x) + \tfrac{L}{2}\|y-x\|_2^2\end{aligned}$$

Note: *Assume that f is twice continuously differentiable in a neighbourhood of a convex set M. Then f is (ℓ, L)-strongly convex on M* **iff** *for all $x \in M$ and all $d \in \mathbb{R}^n$ one has*

$$\ell\|d\|_2^2 \leq d^T f''(x) d \leq L\|d\|_2^2$$
$$\Updownarrow$$
$$\lambda_{\min}(f''(x)) \geq \ell, \ \lambda_{\max}(f''(x)) \leq L.$$

In particular,

♠ A quadratic function

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

with positive definite symmetric matrix A is strongly convex with the parameters $\ell = \lambda_{\min}(A)$, $L = \lambda_{\max}(A)$ on the entire space.

♣ GD in strongly convex case.

Theorem. *In the strongly convex case, AGD exhibits linear global rate of convergence. Specifically, let the set* $G = \{x : f(x) \le f(x_0)\}$ *be closed and convex and* f *be strongly convex, with parameters* ℓ, L*, on* G*. Then*

♢ G *is compact, and the global minimizer* x_* *of* f *exists and is unique;*

♢ *AGD with* $\epsilon \ge 1/2$ *converges linearly to* x_**:*

$$\|x_t - x_*\|_2 \le \theta^t \|x_0 - x_*\|_2$$

$$\theta = \sqrt{\frac{Q_f - (2-\epsilon^{-1})(1-\epsilon)\eta^{-1}}{Q_f + (\epsilon^{-1}-1)\eta^{-1}}} = 1 - O(Q_f^{-1}).$$

Besides this,

$$f(x_t) - f_* \le \theta^{2t} Q_f [f(x_0) - f_*].$$

♣ SGD in Strongly convex quadratic case.

Assume that $f(x) = \frac{1}{2}x^T Ax - b^T x + c$ is a strongly convex quadratic function: $A = A^T \succ 0$. In this case, SGD becomes implementable and is given by the recurrence

$$\begin{array}{rcl} g_t & = & f'(x_t) = Ax_t - b \\ \gamma_{t+1} & = & \frac{g_t^T g_t}{g_t^T A g_t} \\ x_{t+1} & = & x_t - \gamma_{t+1} g_t \end{array}$$

and guarantees that

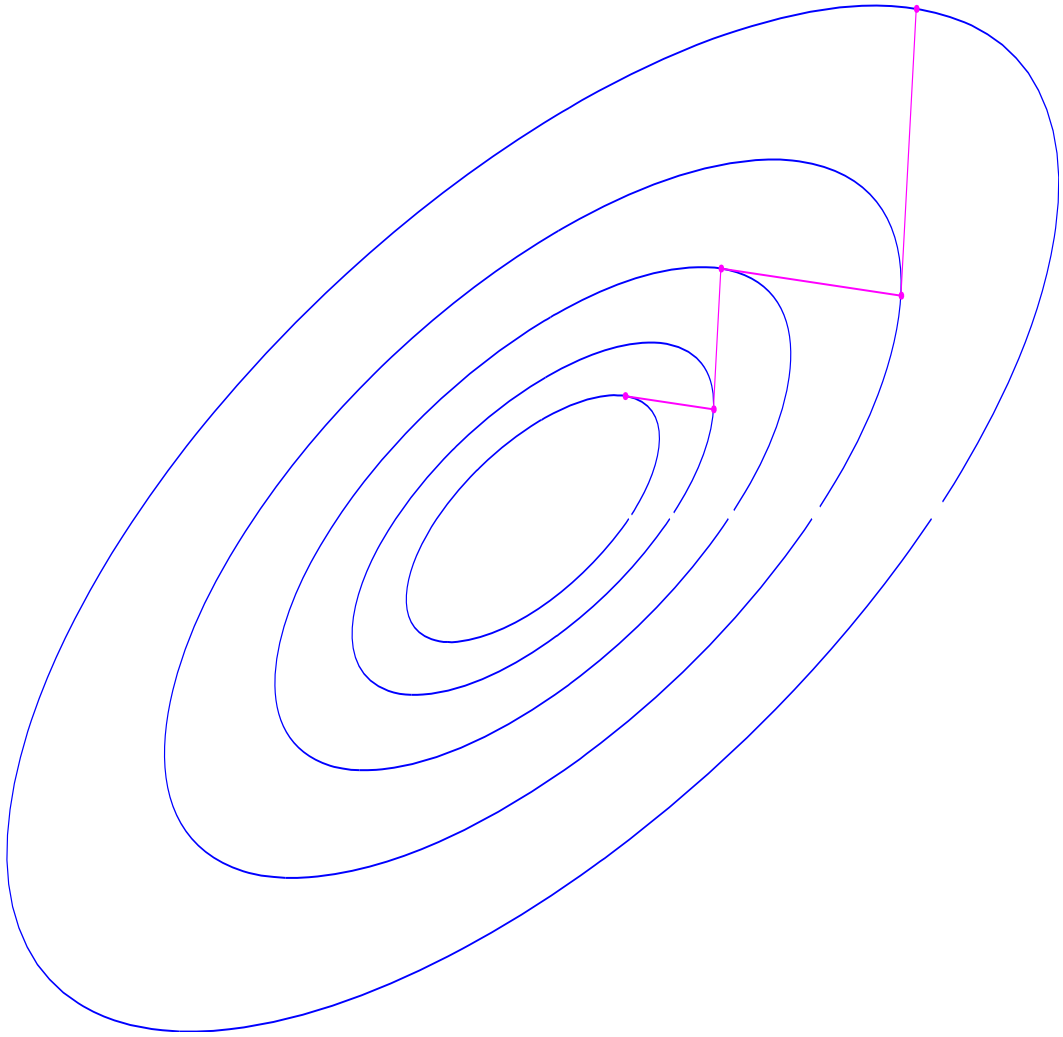
$$\underbrace{f(x_{t+1}) - f_*}_{E_{t+1}} \leq \left[1 - \frac{(g_t^T g_t)^2}{[g_t^T A g_t][g_t^T A^{-1} g_t]}\right] E_t \leq \left(\frac{Q_f - 1}{Q_f + 1}\right)^2 E_t$$

whence

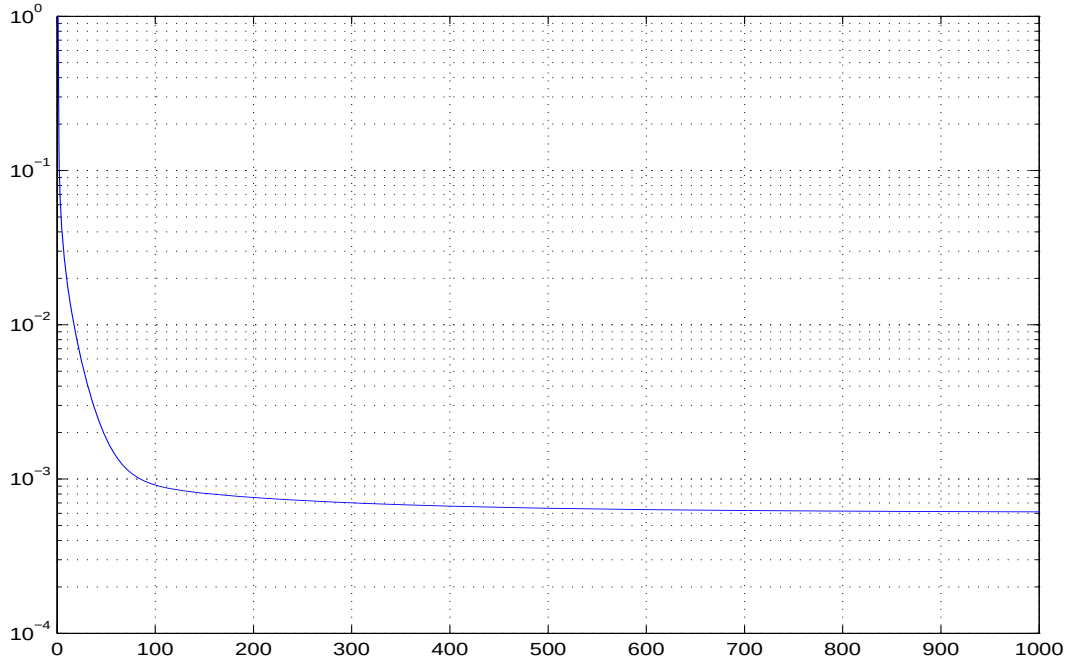
$$f(x_t) - f_* \leq \left(\frac{Q_f - 1}{Q_f + 1}\right)^{2t} [f(x_0) - f_*], \; t = 1, 2, ...$$

<u>Note:</u> **<u>If</u>** we know that SGD converges to a *nondegenerate* local minimizer x_* of f, **<u>then</u>**, under mild regularity assumptions, the *asymptotical* behaviour of the method will be as if f were the strongly convex quadratic form

$$f(x) = \text{const} + \frac{1}{2}(x - x_*)^T f''(x_*)(x - x_*).$$



9.19



Plot of $\dfrac{f(x_t)-f_*}{(f(x_0)-f_*)\left(\dfrac{Q_f-1}{Q_f+1}\right)^{2t}}$

SGD as applied to quadratic form with $Q_f = 1000$

$\diamondsuit$ $f(x_0) = 2069.4$, $f(x_{999}) = 0.0232$

$$\boxed{x_{t+1} = x_t - \gamma_t f'(x_t)}$$

♣ **Summary on Gradient Descent:**

♢ Under mild regularity and boundedness assumptions, both SGD and AGD converge the set of critical points of the objective.

In the case of $\mathrm{C}^{1,1}(L)$-smooth objective, the methods exhibit non-asymptotical $O(1/t)$-rate of convergence in terms of the error measure $\delta^2(x) = \|f'(x)\|_2^2$.

♢ Under the same regularity assumptions, in **Convex** case the methods converge to the set of global minimizers of the objective.

In convex $\mathrm{C}^{1,1}(L)$-case, AGD exhibits non-asymptotical $O(1/t)$ rate of convergence in terms of the residual in the objective $f(x) - f_*$

♢ In *Strongly convex case*, AGD exhibits non-asymptotical linear convergence in both the residual in terms of the objective $f(x) - f_*$ and the distance in the argument $\|x - x_*\|_2$.

The convergence ratio is $1 - O(1/Q_f)$, where Q_f is the condition number of the objective. In other words, to get extra accuracy digit, it takes $O(Q_f)$ steps.

♣ **<u>Good news on GD:</u>**

♠ Simplicity

♠ Reasonable global convergence properties under mild assumptions on the function to be minimized.

♣ **Drawbacks of GD:**

♠ **"Frame-dependence"**: The method is *not* affine invariant!

♢ You are solving the problem $\min_x f(x)$ by GD, starting with $x_0 = 0$, Your first search point will be

$$x_1 = -\gamma_1 f'(0).$$

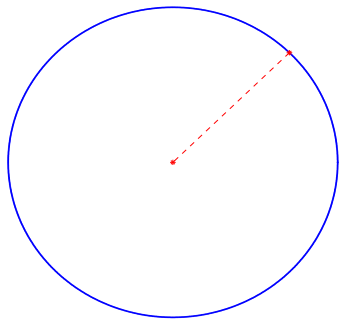
♢ I solve the same problem, but in new variables y: $x = Ay$. My problem is $\min_y g(y)$, $g(y) = f(Ay)$, and I start with $y_0 = 0$. My first search point will be

$$y_1 = -\widehat{\gamma}_1 g'(0) = -\widehat{\gamma}_1 A^T f'(0).$$

In x-variables, my search point will be

$$\widehat{x}_1 = Ay_1 = -\widehat{\gamma}_1 AA^T f'(0)$$

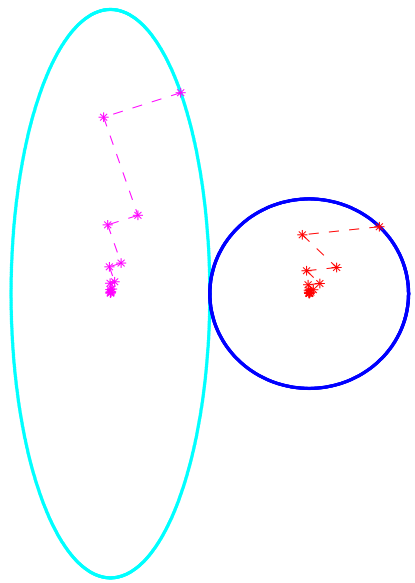
If AA^T is not proportional to the unit matrix, my search point will, in general, be different from yours!



SGD as applied to $f(x) = \frac{1}{2}x^T x$ – exact solution in 1 step!

Substituting $x_1 = y_1, x_2 = y_2/3$, the problem becomes

$$\min_y g(y) = \frac{1}{2}\left[y_1^2 + \frac{1}{9}y_2^2\right]$$



Left: SGD as applied to g

Right: The same trajectory in x-coordinates

t	1	3	5	7	9
$g(y_t)$	0.5000	0.0761	0.0116	0.0018	0.0003

♠ "Frame-dependence" is common drawback of nearly all *first order* optimization methods, and this is what makes their rate of convergence, even under the most favourable case of strongly convex objective, sensitive to the condition number of the problem. GD is "hyper-sensitive" to the condition number: When minimizing strongly convex function f, the convergence ratio of GD is $1 - O(1/Q_f)$, while for better methods it is $1 - O(1/Q_f^{1/2})$.

The Newton Method

♣ Consider unconstrained problem

$$\min_x f(x)$$

with *twice* continuously differentiable objective. Assuming second order information available, we approximate f around a current iterate x by the second order Taylor expansion:

$$f(y) \approx f(x) + (y-x)^T f'(x) + \frac{(y-x)^T f''(x)(y-x)}{2}$$

In the Newton method, the new iterate is the minimizer of this quadratic approximation. *If exists*, the minimizer is given by

$$\nabla_y[f(x) + (y-x)^T f'(x) + \tfrac{(y-x)^T f''(x)(y-x)}{2}] = 0 \Leftrightarrow f''(x)(y-x) = -f'(x)$$
$$\Leftrightarrow y = x - [f''(x)]^{-1} f'(x)$$

We have arrived at the Basic Newton method

$$x_{t+1} = x_t - [f''(x_t)]^{-1} f'(x_t)$$

(step t is undefined when the matrix $f''(x_t)$ is singular).

$$x_{t+1} = x_t - [f''(x_t)]^{-1} f'(x_t)$$

♠ **<u>Alternative motivation:</u>** We seek for a solution to the Fermat equation

$$f'(x) = 0;$$

given current approximate x_t to the solution, we linearize the left hand side around x_t, thus arriving at the linearized Fermat equation

$$f'(x_t) + f''(x_t)[x - x_t] = 0$$

and take the solution to this equation, that is, $x_t - [f''(x_t)]^{-1} f'(x_t)$, as our new iterate.

$$x_{t+1} = x_t - [f''(x_t)]^{-1} f'(x_t) \qquad \text{(Nwt)}$$

Theorem on Local Quadratic Convergence: *Let x_* be a nondegenerate local minimizer of f, so that $f''(x_*) \succ 0$, and let f be three times continuously differentiable in a neighbourhood of x_*. Then the recurrence (Nwt), started close enough to x_*, is well-defined and converges to x_* quadratically:*

$$\|x_t - x_*\|_2 \to 0, t \to \infty \ \& \ \|x_{t+1} - x_*\|_2 \leq C\|x_t - x_*\|_2^2.$$

Proof: 1^0. Let U be a ball centered at x_* where the third derivatives of f are bounded. For $y \in U$ and appropriate constant β_1 one has

$$\begin{aligned} \|\nabla f(y) + \nabla^2 f(y)(x_* - y)\|_2 &\equiv \|\nabla f(y) - [\nabla^2 f(y)(y - x_*) + \underbrace{\nabla f(x_*)}_{=0}]\|_2 \\ &\leq \beta_1 \|y - x_*\|_2^2 \end{aligned} \qquad (1)$$

2^0. Since $f''(x)$ is continuous at $x = x_*$ and $f''(x_*)$ is nonsingular, there exists a ball $U' \subset U$ centered at x_* and a constant β_2 such that

$$y \in U' \Rightarrow \|[f''(y)]^{-1}\| \leq \beta_2. \qquad (2)$$

Situation: There exists a $r > 0$ and positive constants β_1, β_2 such that

$$\|y - x_*\| < r \Rightarrow \left\{ \begin{array}{lrcl} (a) & \|\nabla f(y) + \nabla^2 f(y)(x_* - y)\|_2 & \leq & \beta_1 \|y - x_*\|_2^2 \\ (b) & \|[f''(y)]^{-1}\| & \leq & \beta_2 \end{array} \right.$$

3^0. Let an iterate x_t of the method be close to x_*:

$$x_t \in V = \{x : \|x - x_*\|_2 \leq \rho \equiv \min[\frac{1}{2\beta_1\beta_2}, r]\}.$$

We have

$$\begin{array}{rcl} \|x_{t+1} - x_*\| & = & \|x_t - x_* - [f''(x_t)]^{-1} f'(x_t)\|_2 \\ & = & \| \left[[f''(x_t)]^{-1} [-f''(x_t)(x_* - x_t) - f'(x_t)] \right] \|_2 \\ & \leq & \beta_1\beta_2 \|x_t - x_*\|_2^2 \leq 0.5 \|x_t - x_*\|_2 \end{array}$$

We conclude that the method remains well-defined after step t, and converges to x_* quadratically.

♣ **Illustration:** computing $\sqrt{a}$.

When $a > 0$, $\sqrt{a} = \text{argmin}_{x>0}\,[f_a(x) = a/x + x]$

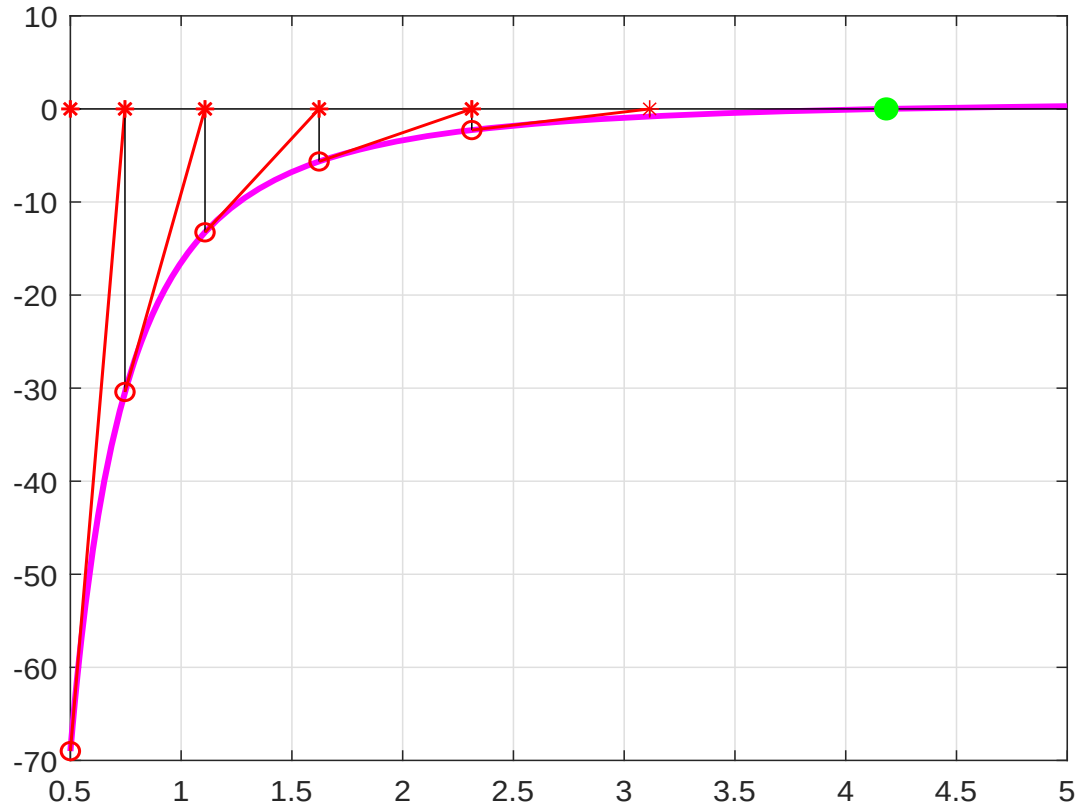
$\Rightarrow \sqrt{a}$ can be computed via Newton recurrence:

$$x_0 > 0, x_{t+1} = x_t - f_a'(x_t)/f_a''(x_t) = x_t + \tfrac{1}{2}[x_t - x_t^3/a].$$

provided x_0 is "reasonable," e.g., $< \sqrt{a}$. This is how it works, a=17.5:

t= 0	x =0.001	
t= 1	x =0.0014999999714285	$x^2 - 17.5$=-1.750e+01
	.....	
t= 21	x =3.6072719622189853	$x^2 - 17.5$=-4.488e+00
t= 22	x =4.0697849320393722	$x^2 - 17.5$=-9.369e-01
t= 23	x =4.1787215172884533	$x^2 - 17.5$=-3.829e-02
t= 24	x =4.1832926184818549	$x^2 - 17.5$=-6.287e-05
t= 25	x =4.1833001326501318	$x^2 - 17.5$=-1.694e-10
t= 26	x =4.1833001326703769	$x^2 - 17.5$=-7.105e-15
t= 27	x =4.1833001326703778	$x^2 - 17.5$=0.000e+00
t= 0	x =5	
t= 1	x =3.9285714285714284	$x^2 - 17.5$=-2.066e+00
t= 2	x =4.1605060391503539	$x^2 - 17.5$=-1.902e-01
t= 3	x =4.1831141693165472	$x^2 - 17.5$=-1.556e-03
t= 4	x =4.1833001202704096	$x^2 - 17.5$=-1.037e-07
t= 5	x =4.1833001326703778	$x^2 - 17.5$=0.000e+00
t= 0	x =10	
t= 1	x =-1.3571428571428573e+01	$x^2 - 17.5$=1.667e+02
t= 2	x =5.1061016243232004e+01	$x^2 - 17.5$=2.590e+03
	.....	
t= 6	x =2.2593416145409303e+76	$x^2 - 17.5$=5.105e+152
t= 7	x =-3.2951687514110151e+227	$x^2 - 17.5$=Inf
t= 8	x = Inf	$x^2 - 17.5$=Inf

♠ Finding $\sqrt{a}$ by Newton minimization of $f(x) = x + a/x$ is the same as finding the root of $f'(x) = 0$ by Newton root finding:



Magenta: $f'(x)$ **Green**: root $\sqrt{a}$ of Fermat equation $f'(x) = 0$ **Red**: Newton iterates

♣ A remarkable property of Newton method is affine invariance ("frame independence"): Let $x = Ay + b$ be invertible affine change of variables. Then

$$\begin{array}{rcl} f(x) & \Leftrightarrow & g(y) = f(Ay+b) \\ \bar{x} = A\bar{y} + b & \Leftrightarrow & \bar{y} \end{array}$$

$$\begin{array}{rcl} \bar{y}_+ & = & \bar{y} - [g''(\bar{y})]^{-1} g'(\bar{y}) = \bar{y} - [A^T f''(\bar{x}) A]^{-1} [A^T f'(\bar{x})] \\ & = & \bar{y} - A^{-1} [f''(\bar{x})]^{-1} f'(\bar{x}) \\ \Rightarrow A\bar{y}_+ + b & = & [A\bar{y} + b] - [f''(\bar{x})]^{-1} f'(\bar{x}) \\ & = & \bar{x} - [f''(\bar{x})]^{-1} f'(\bar{x}) \end{array}$$

♣ Difficulties with Basic Newton method.

The Basic Newton method

$$x_{t+1} = x_t - [f''(x_t)]^{-1} f'(x_t),$$

started close enough to nondegenerate local minimizer x_* *of* f, converges to x_* quadratically. However,

♢ Even for a nice strongly convex f, the method, started *not* too close to the (unique) local≡global minimizer of f, may diverge:

$$f(x) = \sqrt{1 + x^2} \Rightarrow x_{t+1} = -x_t^3.$$

$\Rightarrow$ when $|x_0| < 1$, the method converges quadratically (even at order 3) to $x_* = 0$; when $|x_0| > 1$, the method rapidly diverges...

♢ When f is not strongly convex, the Newton direction

$$-[f''(x)]^{-1} f'(x)$$

can be undefined or fail to be a direction of decrease of f...

♣ As a result of these drawbacks, one needs to modify the Basic Newton method in order to ensure global convergence. Modifications include:

♢ Incorporating line search

♢ Correcting Newton direction when it is undefined or is not a direction of decrease of f.

♣ **Incorporating linesearch:** Assume that the level set $G = \{x : f(x) \leq f(x_0)\}$ is closed and convex, and f is strongly convex on G. Then for $x \in G$ the Newton direction

$$e(x) = -[f''(x)]^{-1} f'(x)$$

is a direction of decrease of f, except for the case when x is a critical point (or, which is the same in the strongly convex case, global minimizer) of f:

$$f'(x) \neq 0 \Rightarrow e^T(x) f'(x) = -[f'(x)]^T \underbrace{[f''(x)]^{-1}}_{\succ 0} f'(x) < 0.$$

In Line Search version of Newton method, one uses $e(x)$ as a search direction rather than the displacement:

$$x_{t+1} = x_t + \gamma_{t+1} e(x_t) = x_t - \gamma_{t+1} [f''(x_t)]^{-1} f'(x_t),$$

where $\gamma_{t+1} > 0$ is the stepsize given by exact minimization of f in the Newton direction or by Armijo linesearch.

Theorem: *Let the level set $G = \{x : f(x) \leq f(x_0)\}$ be convex and compact, and f be strongly convex on G. Then Newton method with the Steepest Descent or with the Armijo linesearch converges to the unique global minimizer of f.*
With proper implementation of the linesearch, convergence is quadratic.

♣ Newton method: Summary

◇ **Good news:** Quadratic asymptotical convergence, provided we manage to bring the trajectory close to a nondegenerate local minimizer

◇ **Bad news:**

— relatively high computational cost, coming from the necessity to compute and to invert the Hessian matrix

— necessity to "cure" the method in the non-strongly-convex case, where the Newton direction can be undefined or fail to be a direction of decrease...

Modifications of the Newton method

♣ Modifications of the Newton method are aimed at overcoming its shortcomings (difficulties with nonconvex objectives, relatively high computational cost) while preserving its major advantage – rapid asymptotical convergence. There are four major groups of modifications:

♢ Newton method with Cubic Regularization

♢ Modified Newton methods based on second-order information

♢ Modifications based on first order information:

— conjugate gradient methods

— quasi-Newton methods

Newton Method with Cubic Regularization

♣ **Problem of interest:**

$$\min_{x \in X} f(x),$$

where

— $X \subset \mathbb{R}^n$ is a closed convex set with a nonempty interior

— f is three times continuously differentiable on X

♠ **Assumption:** We are given starting point $x_0 \in \operatorname{int} X$ such that the set

$$X_0 = \{x \in X : f(x) \leq f(x_0)\}$$

is *bounded* and is contained in the *interior* of X.

♠ **<u>The idea:</u>** To get the idea of the method, consider the case when $X = \mathbb{R}^n$ and the third derivative of f is bounded on X, so that the third order directional derivative of f taken at any point along any unit direction does not exceed some $L \in (0, \infty)$. In this case one has

$$\begin{array}{rcl} \forall x, h: & & \\ f(x+h) & \leq & \overline{f}_x(h), \\ \overline{f}_x(h) & = & f(x) + h^T \nabla f(x) + \frac{1}{2} h^T \nabla^2 f(x) h + \frac{L}{6} \|h\|^3. \end{array}$$

<u>Note:</u> For small h, $\overline{f}_x(h)$ approximates $f(x+h)$ basically as well as the second order Taylor expansion of f taken at x, with the advantage that $\overline{f}_x(h)$ upper-bounds $f(x+h)$ for all h.

$\Rightarrow$ *When passing from x to $x^+ = x + h_*$, with $h_* \in \mathrm{Argmin}_h \overline{f}_x(h)$, we ensure that $f(x^+) \leq \overline{f}_x(h_*) \leq \overline{f}_x(0) = f(x)$, the inequality being strict unless $h_* = 0$ is a global minimizer of $\overline{f}_x(\cdot)$.*

The latter takes place if and only if x satisfies the second order necessary optimality conditions for unconstrained smooth optimization:

$$\nabla f(x) = 0, \nabla^2 f(x) \succeq 0.$$

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

Assumption: *We are given starting point x_0 such that the set $X_0 = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is compact. Besides this, there exists a convex compact set X such that $X_0 \subset \operatorname{int} X$ and f is three times continuously differentiable on X.*

♣ Generic Newton method with Cubic Regularization works as follows. At step t, given previous iterate x_t, we select $L_t > 0$ which is *good* – is such that the displacement

$$h_t \in \mathrm{Argmin}_h \overline{f}(h),$$
$$\overline{f}(h) = f(x_t) + h^T \nabla f(x_t) + \tfrac{1}{2} h^T \nabla^2 f(x_t) h + \tfrac{L_t}{6} \|h\|^3$$

results in $f(x_t + h_t) \leq \overline{f}(h_t)$ and set

$$x_{t+1} = x_t + h_t.$$

<u>Facts:</u> ♢ *Whenever $x_t \in X_0$, all large enough values of L_t, specifically, those with*

$$L_t \geq M_X(f) = \max_{x \in X, h \in \mathbb{R}^n : \|h\| \leq 1} \left. \frac{d^3}{dt^3} \right|_{t=0} f(x + th)$$

are good.

♢ *The algorithm is well defined and ensures that $f(x_0) \geq f(x_1) \geq ...$, all inequalities being strict, unless the algorithm arrives at a point x where second order necessary optimality conditions $\nabla f(x) = 0$, $\nabla^2 f(x) \succeq 0$ take place – at such a point, the algorithm gets stuck.*

◇ *Boundedness and goodness of L_t's is easy to maintain via line search:*

- Given $t \geq 0$, x_t and L_{t-1} (with, say, $L_{-1} = 1$), check one by one whether the candidate values $L^{(k)} = 2^k L_{t-1}$ of L_t are good ($k = 0, \pm 1, \pm 2, ...$).
- Start with $k = 0$.

— If $L^{(0)}$ is good, try $L^{(-1)}$, $L^{(-2)}$,..., until either goodness is lost, or a small threshold (say, 10^{-6}) is achieved, and use the last good candidate value $L^{(k)}$ of L_t as the actual value of L_t.

— If $L^{(0)}$ is bad, try $L^{(1)}$, $L^{(2)}$,..., until goodness is recovered, and use the first good candidate value $L^{(k)}$ of L_t as the actual value of L_t.

This policy ensures that $L_t \leq 2 \max[M_X(f), L_{-1}]$.

$\diamondsuit$ With a policy maintaining boundedness of L_t, the algorithm ensures that

- *All limiting points of the trajectory* (they do exist – the trajectory belongs to a bounded set X_0) *satisfy necessary second order optimality conditions in unconstrained minimization;*
- *Whenever a nondegenerate local minimizer of f is a limiting point of the trajectory, the trajectory converges to this minimizer quadratically.*

♣ **Implementing step of algorithm** requires solving unconstrained minimization problem

$$\min_h \left[p^T h + h^T P h + c\|h\|^3\right] \qquad [P = P^T, c > 0] \qquad (*)$$

• Computing eigenvalue decomposition $P = U\mathrm{Diag}\{\beta\}U^T$ and passing from variables h to variables $g = U^T h$, the problem becomes

$$\min_g \left\{ q^T g + \sum_i \beta_i g_i^2 + c(\sum_i g_i^2)^{\frac{3}{2}} \right\} \qquad [q = U^T p]$$

• At optimum, $\mathrm{sign}(g_i) = -\mathrm{sign}(q_i) \Rightarrow$ *the problem reduces to*

$$\min_g \left\{ -\sum\nolimits_i |q_i||g_i| + \sum\nolimits_i \beta_i g_i^2 + c(\sum\nolimits_i g_i^2)^{\frac{3}{2}} \right\}$$

• Passing to variables $s_i = g_i^2$, *the problem becomes convex*:

$$\min_{s \geq 0} \left\{ -\sum\nolimits_i |q_i|\sqrt{s_i} + \sum\nolimits_i \beta_i s_i + c(\sum\nolimits_i s_i)^{\frac{3}{2}} \right\}. \qquad (!)$$

Optimal solution s^* to (!) gives rise to optimal solution h^* to $(*)$:

$$h^* = U g^*, \; g_i^* = -\mathrm{sign}(q_i)\sqrt{s_i^*}.$$

$$\min_{s\geq 0}\left\{-\sum_i |q_i|\sqrt{s_i}+\sum_i \beta_i s_i + c(\sum_i s_i)^{\frac{3}{2}}\right\}. \qquad (!)$$

- The simplest way to solve (!) is to rewrite (!) as

$$\min_{s,r}\left\{\sum_i [\beta_i s_i - |q_i|\sqrt{s_i}] + cr^{\frac{3}{2}} : s\geq 0, \sum_i s_i \leq r\right\}$$

and to pass to the Lagrange dual

$$\max_{\lambda\geq 0}\left\{\underline{L}(\lambda) := \min_{s\geq 0, r\geq 0}\left[cr^{\frac{3}{2}} - \lambda r + \sum_i [(\beta_i+\lambda)s_i - |q_i|\sqrt{s_i}]\right]\right\} \qquad (D)$$

$\underline{L}(\cdot)$ is easy to compute $\Rightarrow$ (D) can be solved by Bisection. Assuming $|q_i| > 0$ (achievable by small perturbation of q_i's), optimal solution λ_* to the dual problem gives rise to the optimal solution

$$(s_*, r_*) \in \operatorname*{Argmin}_{s\geq 0, r\geq 0}\left[cr^{\frac{3}{2}} - \lambda_* r + \sum_i [[\beta_i+\lambda_*]s_i - |q_i|\sqrt{s_i}]\right]$$

to (!).

Traditional modifications: Variable Metric Scheme

♣ All traditional modifications of Newton method exploit a natural Variable Metric idea.

♠ When speaking about GD, it was mentioned that the method

$$x_{t+1} = x_t - \gamma_{t+1} \underbrace{BB^T}_{A^{-1} \succ 0} f'(x_t) \qquad (*)$$

with nonsingular matrix B has the same "right to exist" as the Gradient Descent

$$x_{t+1} = x_t - \gamma_{t+1} f'(x_t);$$

the former method is nothing but the GD as applied to

$$g(y) = f(By).$$

and then "translated" to $x = By$:

$$x_t = By_t \mapsto x_{t+1} = By_{t+1},\ y_{t+1} = y_t - \gamma_{t+1} g'(y_t) = y_t - \gamma_{t+1} B^T f'(x_t)$$

$$x_{t+1} = x_t - \gamma_{t+1} A^{-1} f'(x_t) \qquad (*)$$

<u>Equivalently:</u> Let A be a positive definite symmetric matrix. We have exactly the same reason to measure the "local directional rate of decrease" of f by the quantity

$$\frac{d^T f'(x)}{\sqrt{d^T d}} \qquad (a)$$

as by the quantity

$$\frac{d^T f'(x)}{\sqrt{d^T A d}} \qquad (b)$$

$\diamondsuit$ When choosing, as the current search direction, the direction of steepest decrease in terms of (a), we get the anti-gradient direction $-f'(x)$ (and all its positive multiples) and arrive at GD.

$\diamondsuit$ When choosing, as the current search direction, the direction of steepest decrease in terms of (b), we get the "scaled anti-gradient direction" $-A^{-1} f'(x)$ (and all its positive multiples) and arrive at "scaled" GD $(*)$.

♣ We have motivated the scaled GD

$$x_{t+1} = x_t - \gamma_{t+1} A^{-1} f'(x_t) \qquad (*)$$

Why not to take one step ahead by considering a generic Variable Metric algorithm

$$x_{t+1} = x_t - \gamma_{t+1} A_{t+1}^{-1} f'(x_t) \qquad \text{(VM)}$$

with "scaling matrix" $A_{t+1} \succ 0$ varying from step to step?

♠ **<u>Note:</u>** When $A_{t+1} \equiv I$, (VM) becomes the generic Gradient Descent; When f is strongly convex and $A_{t+1} = f''(x_t)$, (VM) becomes the generic Newton method...

♠ **<u>Note:</u>** When x_t is *not* a critical point of f, the search direction $d_{t+1} = -A_{t+1}^{-1} f'(x_t)$ is a direction of decrease of f:

$$d_{t+1}^T f'(x_t) = -[f'(x_t)]^T A_{t+1}^{-1} f'(x_t) < 0.$$

Thus, we have no conceptual difficulties with *monotone* linesearch versions of (VM)...

$$x_{t+1} = x_t - \gamma_{t+1} A_{t+1}^{-1} f'(x_t) \qquad \text{(VM)}$$

♣ It turns out that Variable Metric methods possess good global convergence properties:

<u>Theorem:</u> *Let the level set $G = \{x : f(x) \leq f(x_0)\}$ be closed and bounded, and let f be twice continuously differentiable in a neighbourhood of G.*

Assume, further, that the policy of updating the matrices A_t ensures their uniform positive definiteness and boundedness:

$$\exists 0 < \ell \leq L < \infty : \ell I \preceq A_t \preceq LI \ \forall t.$$

Then for both the Steepest Descent and the Armijo versions of (VM) started at x_0, the trajectory is well-defined, belongs to G (and thus is bounded), and f strictly decreases along the trajectory unless a critical point of f is reached. Moreover, all limiting points of the trajectory are critical points of f.

♣ **Implementation via Spectral Decomposition:**

◇ Given x_t, compute $H_t = f''(x_t)$ and then find spectral decomposition of H_t:

$$H_t = V_t \mathrm{Diag}\{\lambda_1, ..., \lambda_n\} V_t^T$$

◇ Given once for ever chosen tolerance $\delta > 0$, set

$$\widehat{\lambda}_i = \max[\lambda_i, \delta]$$

and

$$A_{t+1} = V_t \mathrm{Diag}\{\widehat{\lambda}_1, ..., \widehat{\lambda}_n\} V_t^T$$

Note: The construction ensures uniform positive definiteness and boundedness of $\{A_t\}_t$, provided the level set $G = \{x : f(x) \leq f(x_0)\}$ is compact and f is twice continuously differentiable in a neighbourhood of G.

♣ **Levenberg-Marquard implementation:**

$$A_{t+1} = \epsilon_t I + H_t,$$

where $\epsilon_t \geq 0$ is chosen to ensure that $A_{t+1} \succeq \delta I$ with once for ever chosen $\delta > 0$.

♢ ϵ_t is found by Bisection as applied to the problem

$$\min \{\epsilon : \epsilon \geq 0, H_t + \epsilon I \succeq \delta I\}$$

♢ Bisection requires to check whether the condition

$$H_t + \epsilon I \succ \delta I \Leftrightarrow H_t + (\epsilon - \delta) I \succ 0$$

holds true for a given value of ϵ, and the underlying test comes from Choleski decomposition.

♣ **Choleski Decomposition.** By Linear Algebra, a symmetric matrix P is $\succ 0$ iff

$$P = DD^T \tag{$*$}$$

for some lower triangular matrix D with positive diagonal entries. When Choleski Decomposition $(*)$ exists, it can be found by a simple algorithm.

Choleski Decomposition Algorithm

♠ In Choleski Decomposition $P = DD^T$ lower triangular D is filled *column by column*.

$$\boxed{\begin{array}{c} P_{ik} = \mathrm{Row}_i \mathrm{Row}_k^T \\ [\mathrm{Row}_i\text{: } i\text{-th row of } D] \end{array}}$$

$$\begin{bmatrix} ? & & & & \\ ? & ? & & & \\ ? & ? & ? & & \\ \vdots & \vdots & \vdots & \ddots & \\ ? & ? & ? & \dots & ? \end{bmatrix} \Rightarrow \begin{bmatrix} D_{11} & & & & \\ ? & ? & & & \\ ? & ? & ? & & \\ \vdots & \vdots & \vdots & \ddots & \\ ? & ? & ? & \dots & ? \end{bmatrix} \Rightarrow \begin{bmatrix} D_{11} & & & & \\ D_{21} & ? & & & \\ D_{31} & ? & ? & & \\ \vdots & \vdots & \vdots & \ddots & \\ D_{n1} & ? & ? & \dots & ? \end{bmatrix}$$

$D_{11}^2 = \mathrm{Row}_1 \mathrm{Row}_1^T = P_{11}$ $\qquad$ $D_{i1} D_{11} = \mathrm{Row}_i \mathrm{Row}_1^T = P_{i1}$

$$\begin{bmatrix} D_{11} & & & & \\ D_{21} & ? & & & \\ D_{31} & ? & ? & & \\ \vdots & \vdots & \vdots & \ddots & \\ D_{n1} & ? & ? & \dots & ? \end{bmatrix} \Rightarrow \begin{bmatrix} D_{11} & & & & \\ D_{21} & D_{22} & & & \\ D_{31} & ? & ? & & \\ \vdots & \vdots & \vdots & \ddots & \\ D_{n1} & ? & ? & \dots & ? \end{bmatrix} \Rightarrow \begin{bmatrix} D_{11} & & & & \\ D_{21} & D_{22} & & & \\ D_{31} & D_{32} & ? & & \\ \vdots & \vdots & \vdots & \ddots & \\ D_{n1} & D_{n2} & ? & \dots & ? \end{bmatrix}$$

$D_{21}^2 + D_{22}^2 = \mathrm{Row}_2 \mathrm{Row}_2^T = P_{22}$ $\qquad$ $D_{i1} D_{21} + D_{i2} D_{22} = \mathrm{Row}_i \mathrm{Row}_2^T = P_{i2}$

$$P_{ik} = \text{Row}_i \text{Row}_k^T$$

[Row_i: i-th row of D]

$$\begin{bmatrix} D_{11} & & & & \\ D_{21} & D_{22} & & & \\ D_{31} & D_{32} & ? & & \\ \vdots & \vdots & \vdots & \ddots & \\ D_{n1} & D_{n2} & ? & \dots & ? \end{bmatrix} \qquad \begin{bmatrix} D_{11} & & & & \\ D_{21} & D_{22} & & & \\ D_{31} & D_{32} & D_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ D_{n1} & D_{n2} & ? & \dots & ? \end{bmatrix} \qquad \begin{bmatrix} D_{11} & & & & \\ D_{21} & D_{22} & & & \\ D_{31} & D_{32} & D_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ D_{n1} & D_{n2} & D_{n3} & \dots & ? \end{bmatrix}$$

$$D_{31}^2 + D_{32}^2 + D_{33}^2 = \text{Row}_3 \text{Row}_3^T = P_{33}$$

$$D_{i1}D_{31} + D_{i2}D_{32} + D_{i3}D_{33} = \text{Row}_i \text{Row}_3^T = P_{i3}$$

Illustration

- $P = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 2 & -2 \\ 1 & -2 & 3 \end{bmatrix} = \begin{bmatrix} D_{1,1} & & \\ D_{2,1} & D_{2,2} & \\ D_{3,1} & D_{3,2} & D_{3,3} \end{bmatrix} \begin{bmatrix} D_{1,1} & D_{2,1} & D_{3,1} \\ & D_{2,2} & D_{3,2} \\ & & D_{3,3} \end{bmatrix}$

Step 1: $1 = P_{1,1} = D_{1,1}^2 \Rightarrow D_{1,1} = 1$

$-1 = P_{2,1} = D_{2,1} \cdot D_{1,1} \Rightarrow D_{2,1} = -1$

$1 = P_{3,1} = D_{3,1} \cdot D_{1,1} \Rightarrow D_{3,1} = 1$

- $P = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 2 & -2 \\ 1 & -2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & & \\ -1 & D_{2,2} & \\ 1 & D_{3,2} & D_{3,3} \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 \\ & D_{2,2} & D_{3,2} \\ & & D_{3,3} \end{bmatrix}$

Step 2: $2 = P_{2,2} = (-1)^2 + D_{2,2}^2 \Rightarrow D_{2,2} = 1$

$-2 = P_{3,2} = 1 \cdot (-1) + D_{3,2} \cdot D_{2,2} \Rightarrow D_{3,2} = -1$

- $P = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 2 & -2 \\ 1 & -2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & & \\ -1 & 1 & \\ 1 & -1 & D_{3,3} \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 \\ & 1 & -1 \\ & & D_{3,3} \end{bmatrix}$

Step 3: $3 = P_{3,3} = 1^2 + (-1)^2 + D_{3,3}^2 \Rightarrow D_{3,3} = 1$

$\Rightarrow D = \begin{bmatrix} 1 & & \\ -1 & 1 & \\ 1 & -1 & 1 \end{bmatrix}$

$$P = DD^T \qquad (*)$$

The general algorithm is as follows:

◇ Representation $(*)$ means that

$$i \le j \Rightarrow p_{ij} = d_i d_j^T,$$

where

$$\begin{aligned} d_i &= (d_{i1}, d_{i2}, ..., d_{ii}, 0, 0, 0, 0, ..., 0) \\ d_j &= (d_{j1}, d_{j2}, ..., d_{ji}, ..., d_{jj}, 0, ..., 0) \end{aligned}$$

are the rows of D.

◇ In particular, $p_{i1} = d_{11} d_{i1}$, and we can set $d_{11} = \sqrt{p_{11}}$, $d_{i1} = p_{i1}/d_{11}$, thus specifying the first column of D.

◇ Further, $p_{22} = d_{21}^2 + d_{22}^2$, whence $d_{22} = \sqrt{p_{22} - d_{21}^2}$. After we know d_{22}, we can find all remaining entries in the second column of D from the relations

$$p_{i2} = d_{i1} d_{21} + d_{i2} d_{22} \Rightarrow d_{i2} = \frac{p_{i2} - d_{i1} d_{21}}{d_{22}}, \; i > 2.$$

♢ We proceed in this way: after the first $(k-1)$ columns in D are found, we fill the k-th column according to

$$\begin{aligned} d_{kk} &= \sqrt{p_{kk} - d_{k1}^2 - d_{k2}^2 - \ldots - d_{k,k-1}^2} \\ d_{ik} &= \frac{p_{ik} - d_{i1}d_{k1} - \ldots - d_{i,k-1}d_{k,k-1}}{d_{kk}}, \; i > k. \end{aligned}$$

♠ The outlined process either results in the required D, or terminates when you cannot carry out current pivot, that is, when

$$p_{kk} - d_{k1}^2 - d_{k2}^2 - \ldots - d_{k,k-1}^2 \leq 0$$

This "bad termination" indicates that P is not positive definite.

The outlined *Choleski Algorithm* allows to find the Choleski decomposition, if any, in $\approx \frac{n^3}{6}$ a.o. It is used routinely to solve linear systems

$$Px = b \tag{S}$$

with $P \succ 0$. To solve the system, one first computes the Choleski decomposition

$$P = DD^T$$

and then solves (S) by two *back-substitutions*

$$b \mapsto y : Dy = b, \; y \mapsto x : D^T x = y,$$

that is, by solving two triangular systems of equations (which takes just $O(n^2)$ a.o.).

Another application of the algorithm (e.g., in Levenberg-Marquardt method) is to check positive definiteness of a symmetric matrix.

<u>Note:</u> The Levenberg-Marquardt method produces uniformly positive definite bounded sequence $\{A_t\}$, provided that the set $G = \{x : f(x) \leq f(x_0)\}$ is compact and f is twice continuously differentiable in a neighbourhood of G.

♣ The "most practical" implementation of Modified Newton Method is based on running the Choleski decomposition as applied to $H_t = f''(x_t)$. When in course of this process the current pivot (that is, specifying d_{kk}) becomes impossible or results in $d_{kk} < \delta$, one increases the corresponding diagonal entry in H_t until the condition $d_{kk} = \delta$ is met.
With this approach, one finds a diagonal correction of H_t which makes the matrix "well positive definite" and ensures uniform positive definiteness and boundedness of the resulting sequence $\{A_t\}$, provided that the set $G = \{x : f(x) \leq f(x_0)\}$ is compact and f is twice continuously differentiable in a neighbourhood of G.

Conjugate Gradient methods

♣ Consider a problem of minimizing a positive definite quadratic form

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

Here is a "conceptual algorithm" for minimizing f, or, which is the same, for solving the system

$$Hx = b:$$

Given starting point x_0, *let* $g_0 = f'(x_0) = Hx_0 - b$, *and let us define Krylov's subspaces*

$$E_t = \mathrm{Lin}\{g_0, Hg_0, H^2 g_0, ..., H^{t-1} g_0\},\ t = 0, 1, ...$$

$$[E_0 = \{0\}, E_1 = \mathrm{Lin}\{g_0\} = \mathbb{R} \cdot g_0, E_2 = \mathrm{Lin}\{g_0, Hg_0\} = \mathbb{R} \cdot g_0 + \mathbb{R} \cdot Hg_0, ...]$$

and set

$$x_t = \underset{x \in x_0 + E_t}{\mathrm{argmin}}\ f(x).$$

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

Given starting point x_0*, let* $g_0 = f'(x_0) = Hx_0 - b$*, and let*

$$E_t = \mathrm{Lin}\{g_0, Hg_0, H^2 g_0, ..., H^{t-1} g_0\},$$

and

$$x_t = \underset{x \in x_0 + E_t}{\mathrm{argmin}}\ f(x).$$

<u>Fact I:</u> $\{0\} = E_0 \subseteq E_1 \subseteq E_2 \subseteq E_3 ...$ Let t_* be the smallest integer t such that $E_{t+1} = E_t$. Then $t_* \leq n$, and x_{t_*} is the unique minimizer of f on $\mathbb{R}^n$

<u>Fact II:</u> One has

$$f(x_t) - \min_x f(x) \leq 4 \left[\frac{\sqrt{Q_f} - 1}{\sqrt{Q_f} + 1} \right]^{2t} [f(x_0) - \min_x f(x)] \qquad (*)$$

where Q_f is the *condition number of* f – the ratio of the largest and the smallest eigenvalues of H.

Note: *Every* $\sqrt{Q_f}$ *new iterations decrease the right hand side in* $(*)$ *by absolute constant factor.* For Steepest decent similar improvement requires Q_f new iterations...

Fact III: The trajectory $\{x_t\}$ is given by explicit recurrence generating *iterates* x_t, *search directions* d_t, and *gradients* $g_t = f'(x_t)$ according to:

◇ **Initialization:** Set

$$d_0 = -g_0 \equiv -f'(x_0) = b - Hx_0;$$

◇ **Step t:** if $g_{t-1} \equiv f'(x_{t-1}) = 0$, terminate, x_{t-1} being the result. Otherwise set

$$\boxed{\begin{array}{lcl} \gamma_t & = & -\dfrac{g_{t-1}^T d_{t-1}}{d_{t-1}^T H d_{t-1}} \\ x_t & = & x_{t-1} + \gamma_t d_{t-1} \\ \hline g_t & = & f'(x_t) = Hx_t - b \ = g_{t-1} + \gamma_t H d_{t-1} \\ \hline \beta_t & = & \dfrac{g_t^T H d_{t-1}}{d_{t-1}^T H d_{t-1}} \\ d_t & = & -g_t + \beta_t d_{t-1} \end{array}}$$

and loop to step $t+1$.

• **Note:** A step costs *a single* matrix-vector multiplication to compute Hd_{t-1} plus linear in n number of arithmetic operations.

Note: In the above process,

◇ The gradients $g_0, ..., g_{t_*-1}, g_{t_*} = 0$ are mutually orthogonal

◇ The search directions $d_0, d_1, ..., d_{t_*-1}$ are H-orthogonal:

$$i \neq j \Rightarrow d_i^T H d_j = 0$$

◇ One has

$$\begin{aligned} \gamma_t &= \operatorname*{argmin}_{\gamma} f(x_{t-1} + \gamma d_{t-1}) \\ \beta_t &= \frac{g_t^T g_t}{g_{t-1}^T g_{t-1}} \end{aligned}$$

Note: *With this representation of γ_t and β_t, the algorithm does not involve explicit multiplications of vectors by H, only computing gradients of f at iterates and linesearch!*

♣ Conjugate Gradient method as applied to a strongly convex quadratic form f can be viewed as an iterative algorithm for solving the linear system

$$Hx = b.$$

As compared to "direct solvers", like Choleski Decomposition or Gauss elimination, the advantages of CG are:

♢ Ability, in the case of exact arithmetic, to find solution in at most n steps, with a single matrix-vector multiplication and $O(n)$ additional operations per step.

$\Rightarrow$ *The cost of finding the solution is at most $O(n)L$, where L is the arithmetic price of matrix-vector multiplication.*

Note: When H is sparse, $L << n^2$, and the price of the solution becomes much smaller than the price $O(n^3)$ for the direct LA methods.

♢ In principle, there is no necessity to assemble H – all we need is the possibility to multiply by H

♢ The non-asymptotic error bound $f(x_t) - \min_x f(x) \leq 4 \left[\frac{\sqrt{Q_f}-1}{\sqrt{Q_f}+1}\right]^{2t} [f(x_0) - \min_x f(x)]$ indicates *rate of convergence completely independent of the dimension and depending only on the condition number of H.*

♠ **Illustrations:**

◇ System 1000×1000, $Q_f = 1.e2$:

Itr	$f - f_*$	$\|x - x_*\|_2$
1	2.297e+03	2.353e+01
11	1.707e+01	4.265e+00
21	3.624e-01	6.167e-01
31	6.319e-03	8.028e-02
41	1.150e-04	1.076e-02
51	2.016e-06	1.434e-03
61	3.178e-08	1.776e-04
71	5.946e-10	2.468e-05
81	9.668e-12	3.096e-06
91	1.692e-13	4.028e-07
94	4.507e-14	2.062e-07

◇ System 1000×1000, $Q_f = 1.e4$:

Itr	$f - f_*$	$\|x - x_*\|_2$
1	1.471e+05	2.850e+01
51	1.542e+02	1.048e+01
101	1.924e+01	4.344e+00
151	2.267e+00	1.477e+00
201	2.248e-01	4.658e-01
251	2.874e-02	1.779e-01
301	3.480e-03	6.103e-02
351	4.154e-04	2.054e-02
401	4.785e-05	6.846e-03
451	4.863e-06	2.136e-03
501	4.537e-07	6.413e-04
551	4.776e-08	2.109e-04
601	4.954e-09	7.105e-05
651	5.666e-10	2.420e-05
701	6.208e-11	8.144e-06
751	7.162e-12	2.707e-06
801	7.850e-13	8.901e-07
851	8.076e-14	2.745e-07
901	7.436e-15	8.559e-08
902	7.152e-15	8.412e-08

◇ System 1000×1000, $Q_f = 1.e6$:

Itr	$f - f_*$	$\|x - x_*\|_2$
1	9.916e+06	2.849e+01
1000	7.190e+00	2.683e+00
2000	4.839e-02	2.207e-01
3000	4.091e-04	1.999e-02
4000	2.593e-06	1.602e-03
5000	1.526e-08	1.160e-04
6000	1.159e-10	1.102e-05
7000	6.022e-13	7.883e-07
8000	3.386e-15	5.595e-08
8103	1.923e-15	4.236e-08

◇ System 1000×1000, $Q_f = 1.e12$:

Itr	$f - f_*$	$\|x - x_*\|_2$
1	5.117e+12	3.078e+01
1000	1.114e+07	2.223e+01
2000	2.658e+06	2.056e+01
3000	1.043e+06	1.964e+01
4000	5.497e+05	1.899e+01
5000	3.444e+05	1.851e+01
6000	2.343e+05	1.808e+01
7000	1.760e+05	1.775e+01
8000	1.346e+05	1.741e+01
9000	1.045e+05	1.709e+01
10000	8.226e+04	1.679e+01

♣ **Non-Quadratic Extensions:** CG in the form

$$\begin{array}{rcl} d_0 & = & -g_0 = -f'(x_0) \\ \gamma_t & = & \underset{\gamma}{\text{argmin}}\, f(x_{t-1} + \gamma d_{t-1}) \\ x_t & = & x_{t-1} + \gamma_t d_{t-1} \\ g_t & = & f'(x_t) \\ \beta_t & = & \frac{g_t^T g_t}{g_{t-1}^T g_{t-1}} \\ d_t & = & -g_t + \beta_t d_{t-1} \end{array}$$

can be applied to *whatever* function f, not necessarily quadratic one (Fletcher-Reevs CG), and similarly for another equivalent *in the quadratic case* form:

$$\begin{array}{rcl} d_0 & = & -g_0 = -f'(x_0) \\ \gamma_t & = & \underset{\gamma}{\text{argmin}}\, f(x_{t-1} + \gamma d_{t-1}) \\ x_t & = & x_{t-1} + \gamma_t d_{t-1} \\ g_t & = & f'(x_t) \\ \beta_t & = & \frac{(g_t - g_{t-1})^T g_t}{g_{t-1}^T g_{t-1}} \\ d_t & = & -g_t + \beta_t d_{t-1} \end{array}$$

(Polak-Ribiere CG).

♠ *Being equivalent in the quadratic case, these (and other) forms of CG become different in the non-quadratic case!*

♠ Non-quadratic extensions of CG can be used with and without *restarts*.

♢ In quadratic case CG, modulo rounding errors, terminates in at most n steps with exact solution. In non-quadratic case this is not so.

♢ In non-quadratic CG with restarts, execution is split into n-step *cycles*, and cycle $t+1$ starts from the last iterate x^t of the previous cycle as from the starting point (that is, set search direction to be minus the current gradient)

In contrast to this, with no restarts the recurrence like

$$\begin{array}{rcl} d_0 & = & -g_0 = -f'(x_0) \\ \gamma_t & = & \operatorname*{argmin}_{\gamma} f(x_{t-1} + \gamma d_{t-1}) \\ x_t & = & x_{t-1} + \gamma_t d_{t-1} \\ g_t & = & f'(x_t) \\ \beta_t & = & \frac{(g_t - g_{t-1})^T g_t}{g_{t-1}^T g_{t-1}} \\ d_t & = & -g_t + \beta_t d_{t-1} \end{array}$$

is never "refreshed".

Theorem: *Let the level set $\{x : f(x) \leq f(x_0)\}$ of f be compact and f be twice continuously differentiable in a neighbourhood of G. When minimizing f by Fletcher-Reevs or Polak-Ribiere Conjugate Gradients with exact linesearch and restarts,*

$\diamondsuit$ *the trajectory is well-defined and bounded*

$\diamondsuit$ *f never increases*

$\diamondsuit$ *all limiting points of the sequence x^t of concluding iterates of the subsequent cycles are critical points of f.*

$\diamondsuit$ *If, in addition, x^t converge to a nondegenerate local minimizer x_* of f and f is 3 times continuously differentiable around x_*, then x^t converge to x_* quadratically.*

Quasi-Newton Methods

♣ Quasi-Newton methods are variable metric methods of the generic form

$$x_{t+1} = x_t - \gamma_{t+1} \underbrace{S_{t+1}}_{=A_{t+1}^{-1}} f'(x_t)$$

where $S_{t+1} \succ 0$ and γ_{t+1} is given by linesearch.

♠ In contrast to Modified Newton methods, in Quasi-Newton algorithms one operates directly on matrix S_{t+1}, with the ultimate goal to ensure, under favourable circumstances, that

$$S_{t+1} - [f''(x_t)]^{-1} \to 0, \, t \to \infty. \qquad (*)$$

♠ In order to achieve $(*)$, in Quasi-Newton methods one updates S_t into S_{t+1} in a way which ensures that

♢ S_{t+1} is $\succ 0$

♢ $S_{t+1}(g_t - g_{t-1}) = x_t - x_{t-1}$, where $g_\tau = f'(x_\tau)$ [secant equation]

Note: *The second relation is motivated by what happens when* $f = \frac{1}{2}x^T H x - b^T x + c$ *is quadratic strongly convex and* $S_{t+1} = H^{-1}$

♣ **Generic Quasi-Newton method:**

Initialization: Choose somehow starting point x_0, matrix $S_1 \succ 0$, compute $g_0 = f'(x_0)$.

Step t: given x_{t-1}, $g_{t-1} = f'(x_{t-1})$ and $S_t \succ 0$, terminate when $g_{t-1} = 0$, otherwise

$\diamondsuit$ Set $d_t = -S_t g_{t-1}$ and perform exact line search from x_{t-1} in the direction d_t, thus getting new iterate

$$x_t = x_{t-1} + \gamma_t d_t;$$

$\diamondsuit$ compute $g_t = f'(x_t)$ and set

$$p_t = x_t - x_{t-1},\ q_t = g_t - g_{t-1};$$

$\diamondsuit$ update S_t into positive definite symmetric matrix S_{t+1} in such a way that

$$S_{t+1} q_t = p_t$$

and loop.

Note: $g_{t-1}^T d_t < 0$ (since $g_{t-1} \neq 0$ and $S_t \succ 0$) and $g_t^T d_t = 0$ (since x_t is a minimizer of f on the ray $\{x_{t-1} + \gamma d_t : \gamma > 0\}$)

$\Rightarrow p_t^T q_t > 0$. *This fact is instrumental when justifying positive definiteness of S_t's in the standard Quasi-Newton methods.*

♠ **Davidon-Fletcher-Powell method:**

$$S_{t+1} = S_t + \frac{1}{p_t^T q_t} p_t p_t^T - \frac{1}{q_t^T S_t q_t} S_t q_t q_t^T S_t.$$

♠ The Davidon-Fletcher-Powell method, as applied to a strongly convex quadratic form, finds exact solution in no more than n steps. The trajectory generated by the method initialized with $S_1 = I$ is exactly the one of the Conjugate Gradient method, so that the DFP (Davidon-Fletcher-Powell) method with the indicated initialization is a Conjugate Gradient method.

♣ The Broyden family.

Broyden-Fletcher-Goldfarb-Shanno updating formula:

$$S_{t+1}^{BFGS} = S_t + \frac{1 + q_t^T S_t q_t}{(p_t^T q_t)^2} p_t p_t^T - \frac{1}{p_t^T q_t} \left[p_t q_t^T S_t + S_t q_t p_t^T \right]$$

can be combined with the Davidon-Fletcher-Powell formula

$$S_{t+1}^{DFP} = S_t + \frac{1}{q_t^T p_t} p_t p_t^T - \frac{1}{q_t^T S_t q_t} S_t q_t q_t^T S_t.$$

to yield a single-parametric *Broyden* family of updating formulas

$$S_{t+1}^{\phi} = (1 - \phi) S_{t+1}^{DFP} + \phi S_{t+1}^{BFGS}$$

where $\phi \in [0, 1]$ is parameter.

- "Mixing"

$$S_t \mapsto S_{t+1}^{\phi} = (1-\phi) S_{t+1}^{(a)} + \phi S_{t+1}^{(b)} \qquad [0 \le \phi \le 1]$$

of two legitimate updating policies is legitimate policy as well: if

$$\{S_{t+1}^{(a)} \succ 0 \ \& \ S_{t+1}^{(a)} q_t = p_t\} \ \ \& \ \ \{S_{t+1}^{(b)} \succ 0 \ \& \ S_{t+1}^{(b)} q_t = p_t\}$$

then clearly

$$S_{t+1}^{\phi} \succ 0 \ \& \ S_{t+1}^{\phi} q_t = p_t$$

$$S_{t+1}^{BFGS} = S_t + \frac{1 + q_t^T S_t q_t}{(p_t^T q_t)^2} p_t p_t^T - \frac{1}{p_t^T q_t}\left[p_t q_t^T S_t + S_t q_t p_t^T\right] \qquad \text{(BFGS)}$$

♠ (BFGS) "mirrors" the Davidon-Fletcher-Powell updating:

• We are looking for a policy for updating $S_t \succ 0$ into $S_{t+1} \succ 0$ while ensuring $S_{t+1} q_t = p_t$. In terms of the inverses H of the S-matrices, this is a policy for updating $H_t = S_t^{-1} \succ 0$ into $H_{t+1} = S_{t+1}^{-1} \succ 0$ while ensuring $H_{t+1} p_t = q_t$.

• Using DFP (with p_t and q_t swapped!) as the policy for updating H-matrices and looking what this policy yields for S-matrices, one arrives at (BFGS).

♣ **Facts:**

♢ *As applied to a strongly convex quadratic form f, the Broyden method minimizes the form exactly in no more than n steps, n being the dimension of the design vector. If S_1 is proportional to the unit matrix, then the trajectory of the method on f is exactly the one of the Conjugate Gradient method.*

♢ *all Broyden methods, independently of the choice of the parameter ϕ, being started from the same pair (x_0, S_1) and equipped with the same exact line search and applied to the same problem, generate the same sequence of iterates (although not the same sequence of matrices S_t!).*

♣ *Broyden methods are thought to be the most efficient in practice versions of the Conjugate Gradient and quasi-Newton methods*, with the pure BFGS method ($\phi = 1$) seemingly being the best.

Convergence of Quasi-Newton methods

♣ **Global convergence** of Quasi-Newton methods *without restarts* is proved only for certain versions of the methods and only under strong assumptions on f.

• *For methods with restarts*, where the updating formulas are "refreshed" every m steps by setting $S = S_1$, one can easily prove that under our standard assumption that the level set $G = \{x : f(x) \leq f(x_0)\}$ is compact and f is continuously differentiable in a neighbourhood of G, the trajectory of *starting points* of the cycles is bounded, and all its limiting points are critical points of f.

♣ **Local convergence:**

◇ *For scheme with restarts,* one can prove that if $m = n$ and $S_1 = I$, then the trajectory of *starting points* x^t of cycles, **if** it converges to a nondegenerate local minimizer x_* of f such that f is 3 times continuously differentiable around x_*, converges to x_* quadratically.

◇ **Theorem** [Powell, 1976] *Consider the BFGS method without restarts and assume that the method converges to a nondegenerate local minimizer x^* of a three times continuously differentiable function f. Then the method converges to x^* superlinearly.*

Lecture 10:

Efficient Solvability of Convex Problems

Solving Convex Problems: Ellipsoid Algorithm

♣ There is a wide spectrum of algorithms capable to approximate *global* solutions of convex problems to *high accuracy* in *"reasonable" time.*
We will present one of the "universal" algorithms of this type – the *Ellipsoid method* imposing only minimal additional to convexity requirements on the problem.

♣ The Ellipsoid method is aimed at solving convex problem in the form

$$\mathrm{Opt} = \min_{x \in X \subset \mathbb{R}^n} f(x)$$

where

• f is a real-valued continuous convex function on X which admits subgradients at every point of X.

f is given by *First Order oracle* – a procedure ("black box") which, given on input a point $x \in X$, returns the value $f(x)$ and a subgradient $f'(x)$ of f at x.

For example, when f is differentiable, it is enough to be able to compute the value and the gradient of f at a point from X.

• X is a *closed and bounded* convex set in $\mathbb{R}^n$ with *nonempty interior*.

X is given by *Separation oracle* – a procedure Sep_X which, given on input a point $x \in \mathbb{R}^n$, reports whether $x \in X$, and if it is not the case, returns a *separator* – a *nonzero* vector $e \in \mathbb{R}^n$ such that

$$\max_{y \in X} e^T y \leq e^T x.$$

$$\mathrm{Opt} = \min_{x \in X \subset \mathbb{R}^n} f(x)$$

♠ Usually, the original description of the feasible domain X of the problem is as follows:

$$X = \{x \in Y : g_i(x) \leq 0,\ 1 \leq i \leq m\}$$

where

A Y is a nonempty convex set admitting a simple Separation oracle Sep_Y.

Example: Let Y be nonempty and given by a list of linear inequalities $a_k^T x \leq b_k$, $1 \leq k \leq K$. Here Sep_Y is as follows:

Given a query point x, we check validity of the inequalities $a_k^T x \leq b_k$. If all of them are satisfied, we claim that $x \in Y$, otherwise claim that $x \notin Y$, take a violated inequality – one with $a_k^T x > b_k$ – and return a_k as the required separator e.

Note: We have $\max_{y \in Y} a_k^T y \leq b_k < a_k^T x$, implying that $e := a_k$ separates x and Y and is nonzero (since $Y \neq \emptyset$).

B. $g_i : Y \to \mathbb{R}$ are convex functions on Y given by First Order oracles and such that *given $x \in Y$, we can check whether $g_i(x) \leq 0$ for all i, and if it is not the case, we can find $i_* = i_*(x)$ such that $g_{i_*}(x) > 0$.*

♠ *Under assumptions* **A**, **B**, *assuming X nonempty, it is easy to build a Separation oracle* Sep_X *for* X, namely, as follows:

Given query point $x \in \mathbb{R}^n$, we

— call Sep_Y to check whether $x \in Y$. If it is not the case, $x \notin X$, and the separator of x and Y separates x and X as well. Thus, when Sep_Y reports that $x \notin Y$, we are done.

— when Sep_Y reports that $x \in Y$, we check whether $g_i(x) \leq 0$ for all i. If it is the case, $x \in X$, and we are done. Otherwise we claim that $x \notin X$, find a constraint $g_{i_*}(\cdot) \leq 0$ violated at x: $g_{i_*}(x) > 0$, call First Oracle to compute a subgradient e of $g_{i_*}(\cdot)$ at x and return this e as the separator of x and X.

Note: In the latter case, e is nonzero and separates x and X: since $g_{i_*}(y) \geq g_{i_*}(x) + e^T(y-x) > e^T(y-x)$ and $g_{i_*}(y) \leq 0$ when $y \in X$, we have

$$y \in X \Rightarrow e^T(y-x) < 0$$

It follows that $e \neq 0$ (X is nonempty!) and $\max_{y \in X} e^T y \leq e^T x$.

$$\text{Opt} = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

Assumptions:

• X is convex, closed and bounded set with $\text{int}\, X \neq \emptyset$ given by Separation oracle Sep_X.

• f is convex and continuous function on X given by First Order oracle $\mathcal{O}_f$.

• [new] *We have an "upper bound" on X – we know $R < \infty$ such that the ball B of radius R centered at the origin contains X,*

(?) *How to solve (P) ?*

To get an idea, let us start with univariate case.

Univariate Case: Bisection

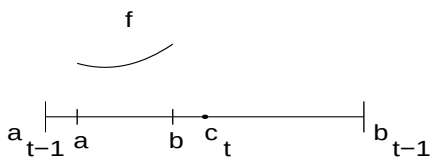
♣ When solving a problem

$$\min_x \{f(x) : x \in X = [a, b] \subset [-R, R]\},$$

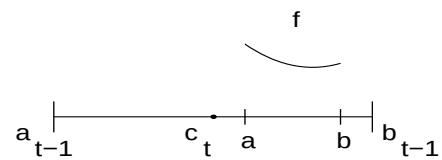
by bisection, we recursively update *localizers* – segments $\Delta_t = [a_{t-1}, b_{t-1}]$ containing the optimal set X_{opt}.

• **Initialization:** Set $\Delta_1 = [-R, R]$ $[\supset X_{\text{opt}}]$

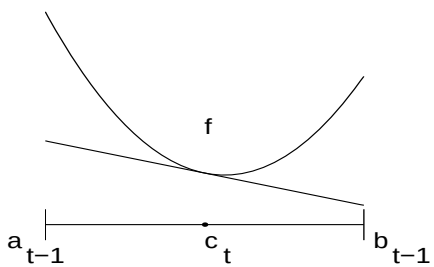
• **Step** t**:** Given $\Delta_t \supset X_{\text{opt}}$ let c_t be the midpoint of Δ_t. Calling Separation and First Order oracles at c_t, we replace Δ_t by *twice smaller* localizer Δ_{t+1}.



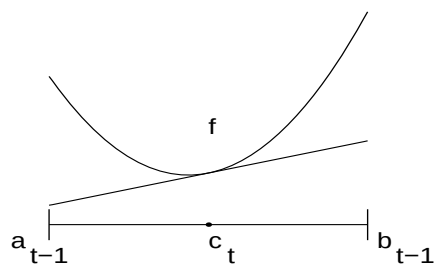
1.a)



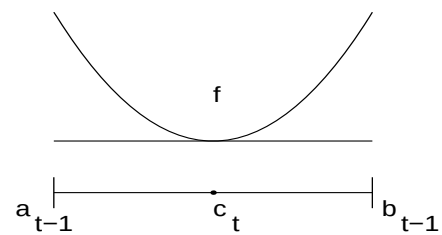
1.b)



2.a)



2.b)



2.c)

1)	Sep_X says that $c_t \notin X$ and reports, via separator e, on which side of c_t X is. 1.a): $\Delta_{t+1} = [a_t, c_t]$; 1.b): $\Delta_{t+1} = [c_t, b_t]$
2)	Sep_X says that $c_t \in X$, and $\mathcal{O}_f$ reports, via $\mathsf{sign} f'(c_t)$, on which side of c_t X_{opt} is. 2.a): $\Delta_{t+1} = [a_t, c_t]$; 2.b): $\Delta_{t+1} = [c_t, b_t]$; 2.c): $c_t \in X_{\mathsf{opt}}$

♠ *Since the localizers rapidly shrink and X is of positive length, eventually some of search points will become feasible, and the nonoptimality of the best found so far feasible search point will rapidly converge to 0 as process goes on.*

♠ Bisection admits multidimensional extension, called *Generic Cutting Plane Algorithm*, where one builds a sequence of "shrinking" *localisers* G_t – closed and bounded convex domains containing the optimal set X_{opt} of (P).

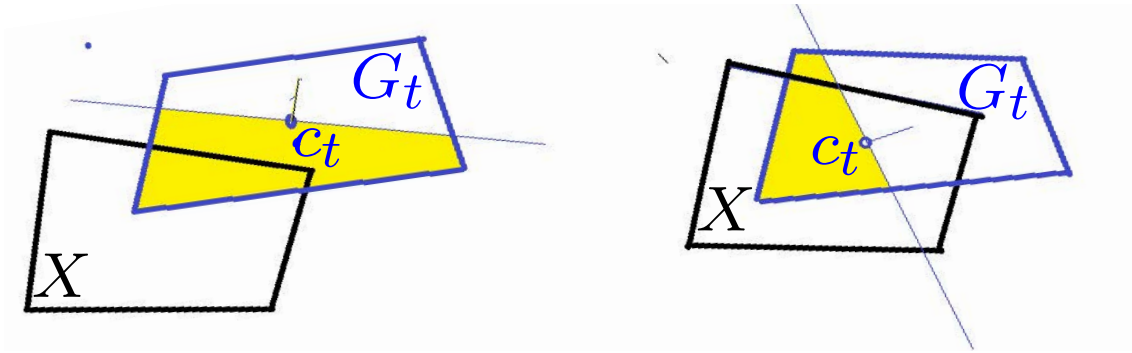
Generic Cutting Plane Algorithm is as follows:

♠ **Initialization** Select as G_1 a closed and bounded convex set containing X and thus being a localizer.

♠ **Step** $t = 1, 2, ...$**:** Given current localizer G_t,

• Select current *search point* $c_t \in G_t$ and call Separation and First Order oracles to form a *cut* – to find $e_t \neq 0$ such that

$$X_{\text{opt}} \subset \widehat{G}_t := \{x \in G_t : e_t^T x \leq e_t^T c_t\}$$



Left: $c_t \notin X$ (case A); right: $c_t \in X$ (case B). Yellow polygon: $\widehat{G}_t$.

— call Sep_X, c_t being the input. If Sep_X says that $c_t \notin X$ and returns a separator, take it as e_t (case A on the picture).

Note: $c_t \notin X \Rightarrow$ *all points from* $G_t \backslash \widehat{G}_t$ *are infeasible*

— if $c_t \in X_t$, call $\mathcal{O}_f$ to compute $f(c_t)$, $f'(c_t)$. If $f'(c_t) = 0$, terminate, otherwise set $e_t = f'(c_t)$ (case B on the picture).

Note: *When* $f'(c_t) = 0$, c_t *is optimal for* (P), *otherwise* $f(x) > f(c_t)$ *at all feasible points from* $G_t \backslash \widehat{G}_t$

• By the two "Note" above, $\widehat{G}_t$ is a localizer along with G_t. Select a closed and bounded convex set $G_{t+1} \supset \widehat{G}_t$ (it also will be a localizer) and pass to step $t + 1$.

$$\mathrm{Opt} = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

♠ *Approximate solution x^t built in course of $t = 1, 2, ...$ steps is the best – with the smallest value of f – of the feasible search points $c_1, ..., c_t$ built so far.*

If in course of the first t steps no feasible search points were built, x^t is undefined.

♣ **Analysing Cutting Plane algorithm**

• Let $\mathrm{Vol}(G)$ be the n-dimensional volume of a closed and bounded convex set $G \subset \mathbb{R}^n$.

Note: For convenience, we use, as the unit of volume, the volume of n-dimensional unit ball $\{x \in \mathbb{R}^n : \|x\|_2 \leq 1\}$, and not the volume of n-dimensional unit box.

• Let us call the quantity $\rho(G) = [\mathrm{Vol}(G)]^{1/n}$ the *radius* of G. $\rho(G)$ is the radius of n-dimensional ball with the same volume as G, and this quantity can be thought of as the average linear size of G.

Theorem. *Let convex problem (P) satisfying our standing assumptions be solved by Generic Cutting Plane Algorithm generating localizers G_1, G_2,... and ensuring that $\rho(G_t) \to 0$ as $t \to \infty$. Let $\bar{t}$ be the first step where $\rho(G_{t+1}) < \rho(X)$. Starting with this step, approximate solution x^t is well defined and obeys the "error bound"*

$$f(x^t) - \mathrm{Opt} \leq \min_{\tau \leq t} \left[\frac{\rho(G_{\tau+1})}{\rho(X)}\right] \left[\max_X f - \min_X f\right]$$

$$\mathrm{Opt}(P) = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

Explanation: Since $\mathrm{int}\, X \neq \emptyset$, $\rho(X)$ is positive, and since X is closed and bounded, (P) is solvable. Let x_* be an optimal solution to (P).

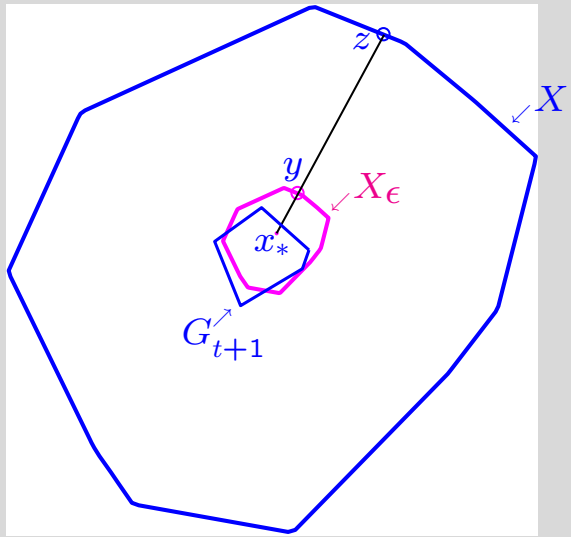
• Let us fix $\epsilon \in (0,1)$ and set $X_\epsilon = x_* + \epsilon(X - x_*)$.

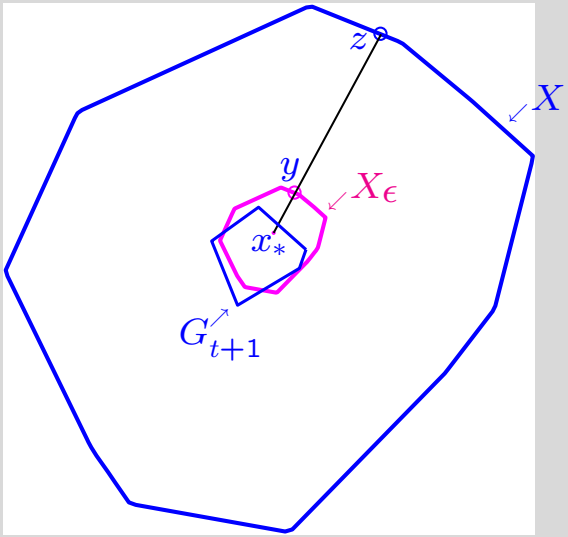
X_ϵ is obtained X by similarity transformation which keeps x_* intact and "shrinks" X towards x_* by factor ϵ. This transformation multiplies volumes by $\epsilon^n \Rightarrow \rho(X_\epsilon) = \epsilon\rho(X)$.

• Let t be such that $\rho(G_{t+1}) < \epsilon\rho(X) = \rho(X_\epsilon)$. Then $\mathrm{Vol}(G_{t+1}) < \mathrm{Vol}(X_\epsilon) \Rightarrow$ *the set* $X_\epsilon \backslash G_{t+1}$ *is nonempty* $\Rightarrow$ *for some* $z \in X$, *the point*

$$y = x_* + \epsilon(z - x_*) = (1-\epsilon)x_* + \epsilon z$$

does not belong to G_{t+1}.





• G_1 contains X and thus y, and G_{t+1} does not contain y, implying that *for some* $\tau \leq t$, *it holds*

$$e_\tau^T y > e_\tau^T c_\tau \qquad (!)$$

• We definitely have $c_\tau \in X$ – otherwise e_τ separates c_τ and $X \ni y$, and (!) witnesses otherwise.

$\Rightarrow c_\tau \in X \Rightarrow e_\tau = f'(c_\tau) \Rightarrow f(c_\tau) + e_\tau^T(y - c_\tau) \leq f(y)$

$\Rightarrow$ [by (!)]

$f(c_\tau) \leq f(y) = f((1-\epsilon)x_* + \epsilon z) \leq (1-\epsilon)f(x_*) + \epsilon f(z)$

$\Rightarrow f(c_\tau) - f(x_*) \leq \epsilon[f(z) - f(x_*)] \leq \epsilon\left[\max_X f - \min_X f\right]$.

Bottom line: *If* $0 < \epsilon < 1$ *and* $\rho(G_{t+1}) < \epsilon\rho(X)$, *then* x^t *is well defined* (since $\tau \leq t$ and c_τ is feasible) *and* $f(x^t) - \mathrm{Opt}(P) \leq \epsilon\left[\max_X f - \min_X f\right]$.

$$\mathrm{Opt}(P) = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

"Starting with the first step $\bar{t}$ where $\rho(G_{t+1}) < \rho(X)$, x^t is well defined, and

$$f(x^t) - \mathrm{Opt} \leq \underbrace{\min_{\tau \leq t} \left[\frac{\rho(G_{\tau+1})}{\rho(X)} \right]}_{\epsilon_t} \underbrace{\left[\max_X f - \min_X f \right]}_{V}"$$

♣ We are done. Let $t \geq \bar{t}$, so that $\epsilon_t < 1$, and let $\epsilon \in (\epsilon_t, 1)$. Then for some $t' \leq t$ we have

$$\rho(G_{t'+1}) < \epsilon \rho(X)$$

$\Rightarrow$ [by bottom line] $x^{t'}$ *is well defined and*

$$f(x^{t'}) - \mathrm{Opt}(P) \leq \epsilon V$$

$\Rightarrow$ [since $f(x^t) \leq f(x^{t'})$ due to $t \geq t'$] x^t *is well defined and* $f(x^t) - \mathrm{Opt}(P) \leq \epsilon V$

$\Rightarrow$ [passing to limit as $\epsilon \to \epsilon_t + 0$] x^t *is well defined and* $f(x^t) - \mathrm{Opt}(P) \leq \epsilon_t V$

□

$$\mathrm{Opt} = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

♠ **Corollary:** *Let (P) be solved by cutting Plane Algorithm which ensures, for some $\vartheta \in (0,1)$, that*

$$\rho(G_{t+1}) \leq \vartheta \rho(G_t)$$

Then, for every desired accuracy $\epsilon > 0$, finding feasible ϵ-optimal solution x_ϵ to (P) (i.e., a feasible solution x_ϵ satisfying $f(x_\epsilon) - \mathrm{Opt} \leq \epsilon$) *takes at most*

$$N = \frac{1}{\ln(1/\vartheta)} \ln\left(\mathcal{R}\left[1 + \frac{V}{\epsilon}\right]\right) + 1$$

steps of the algorithm. Here

$$\mathcal{R} = \frac{\rho(G_1)}{\rho(X)}$$

says how well, in terms of volume, the initial localizer G_1 approximates X, and

$$V = \max_X f - \min_X f$$

is the variation of f on X.

Note: $\mathcal{R}$, and V/ϵ are under log, implying that high accuracy and poor approximation of X by G_1 cost "nearly nothing."

What matters, is the factor *at the log* which is the larger the closer $\vartheta < 1$ is to 1.

"Academic" Implementation: Centers of Gravity

♠ Volumes in high dimensions exhibit counter-intuitive behavior. For example:

• High-dimensional water-mellon with thickness of skin just 1% of the radius is "nearly skin only:"

dimension	3	10	100	750	1000
fraction of watermellon's volume in the skin	0.0297	0.0956	0.6340	0.9995	1.0000

• Large in linear sizes spherical hat $\{x : \|x\|_2 \leq 1, x_1 \geq 0.1\}$ of n-dimensional unit ball $\{x \in \mathbb{R}^n : \|x\|_2 \leq 1\}$ in volume is, for large n, negligibly small part of the ball:

dimension	3	10	100	1000	10000
fraction of ball's volume in the hat	0.4253	0.3727	0.1528	$8.06 \cdot 10^{-4}$	$1.07 \cdot 10^{-23}$

$\Rightarrow$ In high dimensions, to ensure progress in volumes of subsequent localizers in a Cutting Plane algorithm is not an easy task: we do *not* know how the cut through c_t will pass, and thus should select c_t in G_t in such a way that *whatever be the cut*, it cuts off the current localizer G_t a "meaningful" part of its volume.

♠ The most natural choice of c_t in G_t is the *center of gravity*:

$$c_t = \left[\int_{G_t} x dx\right] / \left[\int_{G_t} 1 dx\right],$$

the expectation of the random vector uniformly distributed on G_t.

Good news: The Center of Gravity policy with $G_{t+1} = \widehat{G}_t$ results in

$$\vartheta = \left(1 - \left[\frac{n}{n+1}\right]^n\right)^{1/n} \leq [0.632...]^{1/n} \qquad (*)$$

This results in the complexity bound (# of steps needed to build ϵ-solution)

$$N = 2.2n \ln\left(\mathcal{R}\left[1 + \frac{V}{\epsilon}\right]\right) + 1$$

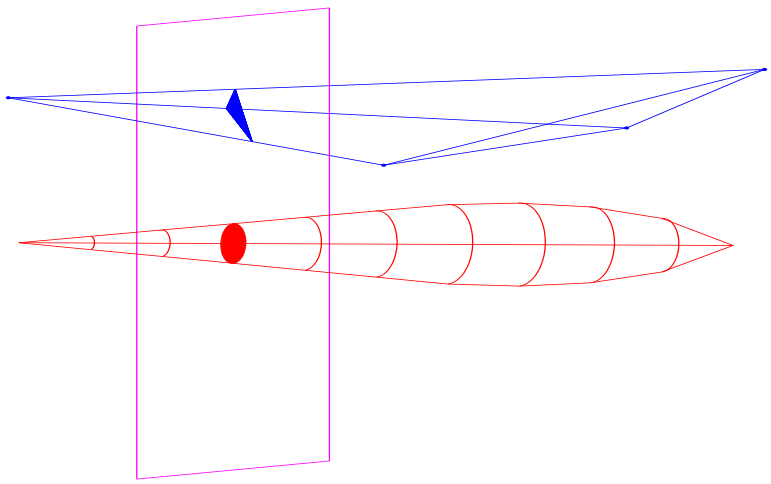
Note: It can be proved that *within absolute constant factor*, like 4, *this is the best complexity bound achievable by whatever algorithm for convex minimization which can "learn" the objective via First Order oracle only.*

♣ **Reason for (*): Brunn-Minkowski Symmeterization Principle:**

Let Y be a convex compact set in $\mathbb{R}^n$, e be a unit direction and Z be "equi-cross-sectional" to X body symmetric w.r.t. e, so that

- *Z is rotationally symmetric w.r.t. the axis e*
- *for every hyperplane $H = \{x : e^T x = \text{const}\}$, one has*

$$\mathrm{Vol}_{n-1}(X \cap H) = \mathrm{Vol}_{n-1}(Z \cap H)$$



Then Z is a convex compact set.

Equivalently: Let U, V be convex compact nonempty sets in $\mathbb{R}^n$. Then

$$\mathrm{Vol}^{1/n}(U + V) \geq \mathrm{Vol}^{1/n}(U) + \mathrm{Vol}^{1/n}(V).$$

In fact, convexity of U, V is redundant!

Disastrously bad news: Centers of Gravity are *not* implementable, unless the dimension n of the problem is like 2 or 3.

Reason: In the method, we have no control on the shape of localizers. Perhaps the best we can say is that if we started with a polytope G_1 given by M linear inequalities, even as simple as a box, then G_t, for meaningful t's, is a more or less arbitrary polytope given by at most $M+t-1$ linear inequalities. And computing center of gravity of a general-type high-dimensional polytope is a computationally intractable task – it requires astronomically many computations already in the dimensions like 5 – 10.

Remedy: *Maintain the shape of G_t simple and convenient for computing centers of gravity,* sacrificing, if necessary, the value of ϑ.

The most natural implementation of this remedy is enforcing G_t to be *ellipsoids*. As a result,

- c_t becomes computable in $O(n^2)$ operations (nice!)
- $\vartheta = [0.632...]^{1/n} \approx \exp\{-0.367/n\}$ increases to $\vartheta \approx \exp\{-0.5/n^2\}$, spoiling the complexity bound

$$N = 2.2n \ln\left(\mathcal{R}\left[1+\frac{V}{\epsilon}\right]\right)+1$$

to

$$N = 4n^2 \ln\left(\mathcal{R}\left[1+\frac{V}{\epsilon}\right]\right)+1$$

(unpleasant, but survivable...)

Practical Implementation - Ellipsoid Method

♠ *Ellipsoid in* $\mathbb{R}^n$ is the image of the unit n-dimensional ball under one-to-one affine mapping:

$$E = E(B, c) = \{x = Bu + c : u^T u \leq 1\}$$

where B is $n \times n$ nonsingular matrix, and $c \in \mathbb{R}^n$.

• c is the center of ellipsoid $E = E(B, c)$: when $c + h \in E$, $c - h \in E$ as well

• When multiplying by $n \times n$ matrix B, n-dimensional volumes are multiplied by $|\mathrm{Det}(B)|$

$\Rightarrow \mathrm{Vol}(E(B, c)) = |\mathrm{Det}(B)|$, $\rho(E(B, c)) = |\mathrm{Det}(B)|^{1/n}$.

Simple fact: *Let $E(B,c)$ be ellipsoid in $\mathbb{R}^n$ and $e \in \mathbb{R}^n$ be a nonzero vector. The "half-ellipsoid"*

$$\widehat{E} = \{x \in E(B,c) : e^T x \leq e^T c\}$$

is covered by the ellipsoid $E^+ = E(B^+, c^+)$ given by

$$\begin{aligned} c^+ &= c - \tfrac{1}{n+1} Bp, \; p = B^T e / \sqrt{e^T B B^T e} \\ B^+ &= \tfrac{n}{\sqrt{n^2-1}} B + \left(\tfrac{n}{n+1} - \tfrac{n}{\sqrt{n^2-1}} \right) (Bp) p^T, \end{aligned}$$

- *$E(B^+, c^+)$ is the ellipsoid of the smallest volume containing the half-ellipsoid $\widehat{E}$, and the volume of $E(B^+, c^+)$ is strictly smaller than the one of $E(B,c)$:*

$$\vartheta := \frac{\rho(E(B^+,c^+))}{\rho(E(B,c))} \leq \exp\{-\tfrac{1}{2n^2}\}.$$

- *Given B, c, e, computing B^+, c^+ costs $O(n^2)$ arithmetic operations.*

$$\mathrm{Opt} = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

♣ **Ellipsoid method** is the Cutting Plane Algorithm where

- all localizers G_t are ellipsoids:
$$G_t = E(B_t, c_t) = \{x = c_t + B_t u : u^T u \leq 1\},$$
- the search point at step t is c_t, and
- G_{t+1} is the smallest volume ellipsoid containing the half-ellipsoid
$$\widehat{G}_t = \{x \in G_t : e_t^T x \leq e_t^T c_t\}$$

Computationally, at every step of the algorithm we once call the Separation oracle Sep_X, (at most) once call the First Order oracle $\mathcal{O}_f$ and spend $O(n^2)$ operations to update (B_t, c_t) into (B_{t+1}, c_{t+1}) by explicit formulas.

♠ **Complexity bound** of the Ellipsoid algorithm is
$$N = 4n^2 \ln\left(\mathcal{R}\left[1 + \frac{V}{\epsilon}\right]\right) + 1$$
$$\mathcal{R} = \frac{\rho(G_1)}{\rho(X)}, \; V = \max_{x \in X} f(x) - \min_{x \in X} f(x)$$

Pay attention:

- $\mathcal{R}, V, \epsilon$ are under log $\Rightarrow$ *large magnitudes in data entries and high accuracy are not issues*
- *the factor at the log depends only on the structural parameter of the problem* (its design dimension n) *and is independent of the remaining data.*

What is Inside Simple Fact

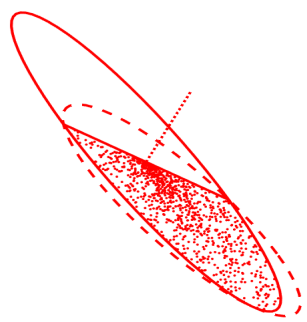
♠ Messy formulas describing the updating

$$(B_t, c_t) \to (B_{t+1}, c_{t+1})$$

in fact are easy to get.

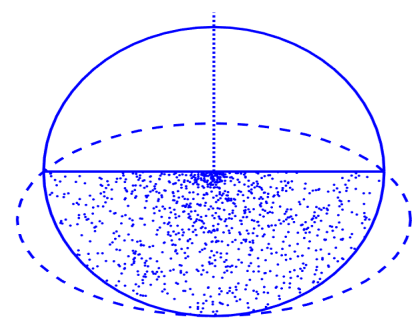
• Ellipsoid E is the image of the unit ball U under affine transformation $u \mapsto c + Bu$. *Affine transformation preserves ratio of volumes*

$\Rightarrow$ *Finding the smallest volume ellipsoid containing a given half-ellipsoid* $\widehat{E}$ *reduces to finding the smallest volume ellipsoid* U^+ *containing a given half-ball* $\widehat{U}$*:*



E, $\widehat{E}$ and E^+

$\overset{\Leftrightarrow}{x = c + Bu}$



U, $\widehat{U}$ and U^+

• The "ball" problem is highly symmetric, and solving it reduces to a simple exercise in elementary Calculus.

Why Ellipsoids?

(?) *When enforcing the localizers to be of "simple and stable" shape, why we make them ellipsoids (i.e., affine images of the unit Euclidean ball), and not something else, say parallelotopes (affine images of the unit box)?*

Answer: In a "simple stable shape" version of Cutting Plane Scheme all localizers are affine images of some fixed n-dimensional *solid* $\mathbf{C}$ (closed and bounded convex set in $\mathbb{R}^n$ with a nonempty interior). To allow for reducing step by step volumes of localizers, $\mathbf{C}$ cannot be arbitrary. What we need is the following property of $\mathbf{C}$:

One can fix a point $\mathbf{c}$ *in* $\mathbf{C}$ *in such a way that whatever be a cut*

$$\widehat{\mathbf{C}} = \{x \in \mathbf{C} : e^T x \leq e^T \mathbf{c}\} \quad [e \neq 0]$$

this cut can be covered by the affine image of $\mathbf{C}$ *of volume less than the one of* $\mathbf{C}$*:*

$$\exists B, b : \widehat{\mathbf{C}} \subset B\mathbf{C} + b \ \& \ |\mathrm{Det}(B)| < 1 \qquad (!)$$

♠ In the Ellipsoid algorithm, $\mathbf{C}$ is the unit Euclidean ball $\Rightarrow |\mathrm{Det}(B)| \leq \exp\{-\frac{1}{2n}\}$.

• Solids $\mathbf{C}$ with the above property are "rare commodity." For example, n-dimensional box does *not* possess it.

• Another "good" solid is n-dimensional simplex (this is not that easy to see!). Here (!) can be satisfied with $|\mathrm{Det}(B)| \leq \exp\{-O(1/n^2)\}$, finally yielding $\vartheta = (1 - O(1/n^3))$.

$\Rightarrow$ *From the complexity viewpoint, "simplex" Cutting Plane algorithm is worse than the Ellipsoid method.*

The same is true for handful of other known so far (and quite exotic) "good solids."

Ellipsoid Method: pro's & con's

♣ **Academically speaking,** *Ellipsoid method is an indispensable tool underlying basically all results on efficient solvability of generic convex problems*, most notably, the famous theorem of L. Khachiyan (1978) on *efficient* (scientifically: *polynomial time*, whatever it means) *solvability of Linear Programming with rational data* – the first ever mathematical result which made the C2 page of *New York Times* (Nov 27, 1979).

♠ *What matters from theoretical perspective,* is "universality" of the algorithm (nearly no assumptions on the problem except for convexity) and complexity bound of the form *"structural parameter outside of log, all else, including required accuracy, under the log."*

♠ Another theoretical (and to some extent, also practical) advantage of the Ellipsoid algorithm is that *as far as the representation of the feasible set X is concerned, all we need is a Separation oracle, and not the list of constraints describing X.* The number of these constraints can be astronomically large, making impossible to check feasibility by looking at the constraints one by one; however, in many important situations the constraints are "well organized," allowing to implement Separation oracle efficiently.

♠ Theoretically, the only (and minor!) drawback of the algorithm is the necessity for the feasible set X to be bounded, with known "upper bound," and to possess nonempty interior.

As of now, there is not way to cure the first drawback without sacrificing universality. The second "drawback" is artifact: given nonempty set

$$X = \{x : g_i(x) \leq 0, 1 \leq i \leq m\},$$

we can extend it to

$$X^\epsilon = \{x : g_i(x) \leq \epsilon, 1 \leq i \leq m\},$$

thus making the interior nonempty, and minimize the objective within accuracy ϵ on this larger set, seeking for ϵ-optimal ϵ-feasible solution instead of ϵ-optimal and *exactly feasible* one.

This is quite natural: to find a feasible solution is, in general, not easier than to find an optimal one. Thus, *either ask for exactly feasible and exactly optimal solution* (which beyond LO is unrealistic), or allow for controlled violation in *both* feasibility and optimality!

♠ **From practical perspective**, theoretical drawbacks of the Ellipsoid method become irrelevant: for all practical purposes, bounds on the magnitude of variables like 10^{100} is the same as no bounds at all, and infeasibility like 10^{-10} is the same as feasibility. And since the bounds on the variables and the infeasibility are under log in the complexity estimate, 10^{100} and 10^{-10} are not a disaster.

♠ **Practical limitations** (rather severe!) of Ellipsoid algorithm stem from method's sensitivity to problem's design dimension n. Theoretically, with $\epsilon, V, \mathcal{R}$ fixed, the number of steps grows with n as n^2, and the effort per step is *at least* $O(n^2)$ a.o.

$\Rightarrow$ *Theoretically, computational effort grows with* n *at least as* $O(n^4)$,

$\Rightarrow$ n *like 1000 and more is beyond the "practical grasp" of the algorithm.*

Note: *Nearly all modern applications of Convex Optimization deal with* n *in the range of tens and hundreds of thousands!*

♠ By itself, growth of *theoretical* complexity with n as n^4 is not a big deal: for Simplex method, this growth is exponential rather than polynomial, and nobody dies – in reality, Simplex does *not* work according to its disastrous theoretical complexity bound.

Ellipsoid algorithm, unfortunately, works more or less according to its complexity bound.

$\Rightarrow$ *Practical scope of Ellipsoid algorithm is restricted to convex problems with few tens of variables.*

However: Low-dimensional convex problems from time to time do arise in applications. More importantly, these problems arise "on a permanent basis" as auxiliary problems within some modern algorithms aimed at solving *extremely large-scale* convex problems.

$\Rightarrow$ *The scope of practical applications of Ellipsoid algorithm is nonempty, and within this scope, the algorithm, due to its ability to produce high-accuracy solutions* (and surprising stability to rounding errors) *can be considered as the method of choice.*

How It Works

$$\text{Opt} = \min_x f(x),\ X = \{x \in \mathbb{R}^n : a_i^T x - b_i \le 0,\ 1 \le i \le m\}$$

♠ Real-life problem with $n = 10$ variables and $m = 81,963,927$ "well-organized" linear constraints:

CPU, sec	t	$f(x^t)$	$f(x^t) - \text{Opt} \le$	$\rho(G_t)/\rho(G_1)$
0.01	1	0.000000	6.7e4	1.0e0
0.53	63	0.000000	6.7e3	4.2e-1
0.60	176	0.000000	6.7e2	8.9e-2
0.61	280	0.000000	6.6e1	1.5e-2
0.63	436	0.000000	6.6e0	2.5e-3
1.17	895	-1.615642	6.3e-1	4.2e-5
1.45	1250	-1.983631	6.1e-2	4.7e-6
1.68	1628	-2.020759	5.9e-3	4.5e-7
1.88	1992	-2.024579	5.9e-4	4.5e-8
2.08	2364	-2.024957	5.9e-5	4.5e-9
2.42	2755	-2.024996	5.7e-6	4.1e-10
2.66	3033	-2.024999	9.4e-7	7.6e-11

Note: My implementation of Ellipsoid algorithm utilizes several simple tricks, including on-line upper bounding of "optimality gaps" $f(x^t) - \text{Opt}$.

♠ Similar problem with $n = 30$ variables and $m = 1,462,753,730$ "well-organized" linear constraints:

CPU, sec	t	$f(x^t)$	$f(x^t) - \text{Opt} \leq$	$\rho(G_t)/\rho(G_1)$
0.02	1	0.000000	5.9e5	1.0e0
1.56	649	0.000000	5.9e4	5.0e-1
1.95	2258	0.000000	5.9e3	8.1e-2
2.23	4130	0.000000	5.9e2	8.5e-3
5.28	7080	-19.044887	5.9e1	8.6e-4
10.13	10100	-46.339639	5.7e0	1.1e-4
15.42	13308	-49.683777	5.6e-1	1.1e-5
19.65	16627	-50.034527	5.5e-2	1.0e-6
25.12	19817	-50.071008	5.4e-3	1.1e-7
31.03	23040	-50.074601	5.4e-4	1.1e-8
37.84	26434	-50.074959	5.4e-5	1.0e-9
45.61	29447	-50.074996	5.3e-6	1.2e-10
52.35	31983	-50.074999	1.0e-6	2.0e-11

Lecture 11:

Algorithms for Constrained Optimization, I: Penalty/Barrier Methods

Algorithms for Constrained Optimization

♣ Traditional methods for general constrained problems

$$\min_x \left\{ f(x) : \begin{array}{l} g_j(x) \leq 0,\ j = 1, ..., m \\ h_i(x) = 0,\ i = 1, ..., k \end{array} \right\} \qquad (P)$$

can be partitioned into

♢ **Primal** methods, where one mimics unconstrained approach, travelling along the feasible set in a way which ensures progress in objective at every step

♢ **Penalty/Barrier** methods, which reduce constrained minimization to solving a sequence of essentially unconstrained problems

♢ **Lagrange Multiplier** methods, where one focuses on dual problem associated with (P). A posteriori the Lagrange multiplier methods, similarly to the penalty/barrier ones, reduce (P) to a sequence of unconstrained problems, but in a "smart" manner different from the penalty/barrier scheme

♢ **Sequential Quadratic Programming** methods,
where one directly solves the KKT system associated with (P) by a kind of Newton method.

Penalty/Barrier Methods

♣ **Penalty Scheme, Equality Constraints.** Consider equality constrained problem

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \qquad (P)$$

and let us "approximate" it by unconstrained problem

$$\min_x f_\rho(x) = f(x) + \underbrace{\frac{\rho}{2}\sum_{i=1}^{k} h_i^2(x)}_{\substack{\text{penalty}\\ \text{term}}} \qquad (P[\rho])$$

$\rho > 0$ is penalty parameter.

Note: (A) On the feasible set, the penalty term vanishes, thus $f_\rho \equiv f$; (B) When ρ is large and x is infeasible, $f_\rho(x)$ is large:

$$\lim_{\rho\to\infty} f_\rho(x) = \begin{cases} f(x), & x \text{ is feasible} \\ +\infty, & \text{otherwise} \end{cases}$$

$\Rightarrow$ *It is natural to expect that solution of* $(P[\rho])$ *approaches, as* $\rho \to \infty$, *the optimal set of* (P).

♣ **Penalty Scheme, General Constraints.** In the case of general constrained problem

$$\min_x \left\{ f(x) : \begin{array}{l} h_i(x) = 0,\ i = 1, ..., k \\ g_j(x) \leq 0,\ j = 1, ..., m \end{array} \right\}, \qquad (P)$$

the same idea of penalizing the constraint violations results in approximating (P) by unconstrained problem

$$\min_x f_\rho(x) = f(x) + \underbrace{\frac{\rho}{2}\left[\sum_{i=1}^{k} h_i^2(x) + \sum_{j=1}^{m} [g_j(x)^+]^2\right]}_{\text{penalty term}} \qquad (P[\rho])$$

where

$$g_j^+(x) = \max[g_j(x), 0]$$

and $\rho > 0$ is penalty parameter. Here again

$$\lim_{\rho \to \infty} f_\rho(x) = \begin{cases} f(x), & x \text{ is feasible} \\ +\infty, & \text{otherwise} \end{cases}$$

and we again may expect that the solutions of $(P[\rho])$ approach, as $\rho \to \infty$, the optimal set of (P).

♣ **<u>Barrier scheme</u>** normally is used for *inequality constrained* problems

$$\min_x \left\{ f(x) : g_j(x) \leq 0,\ j = 1, ..., m \right\} \qquad (P)$$

satisfying "Slater condition": *the feasible set*

$$G = \left\{ x : g_j(x) \leq 0,\ j \leq m \right\}$$

of (P) *possesses a nonempty interior* $\operatorname{int} G$ *which is dense in* G*, and* $g_j(x) < 0$ *for* $x \in \operatorname{int} G$*.*

♠ Given (P), one builds a *barrier* ($\equiv$interior penalty) for G – a function F which is well-defined and smooth on $\operatorname{int} G$ and blows up to $+\infty$ along every sequence of points $x_i \in \operatorname{int} G$ converging to a boundary point of G:

$$x_i \in \operatorname{int} G,\ \lim_{i \to \infty} x_i = x \notin \operatorname{int} G \Rightarrow F(x_i) \to \infty,\ i \to \infty.$$

<u>Examples:</u>

♢ Log-barrier $F(x) = -\sum_j \ln(-g_j(x))$

♢ Carrol Barrier $F(x) = -\sum_j \frac{1}{g_j(x)}$

$$\min_x \left\{ f(x) : g_j(x) \leq 0,\ j = 1, ..., m \right\} \qquad (P)$$

♠ After interior penalty F for the feasible domain of (P) is chosen, the problem is approximated by the "essentially unconstrained" problem

$$\min_{x \in \operatorname{int} G} F^\rho(x) = f(x) + \frac{1}{\rho} F(x) \qquad (P[\rho])$$

When *penalty parameter* ρ is large, the function F^ρ is close to f everywhere in G, except for a thin stripe around the boundary.

$\Rightarrow$ It is natural to expect that solutions of $(P[\rho])$ approach the optimal set of (P) as $\rho \to \infty$,

Investigating Penalty Scheme

♣ Let us focus on equality constrained problem

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \qquad (P)$$

and associated penalized problems

$$\min_x f_\rho(x) = f(x) + \frac{\rho}{2}\|h(x)\|_2^2 \qquad (P[\rho])$$

(results for general case are similar).

♠ **Questions of interest:**

♢ Whether indeed unconstrained minimizers of the penalized objective f_ρ converge, as $\rho \to \infty$, to the optimal set of (P)?

♢ What are our possibilities to minimize the penalized objective?

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \qquad (P)$$

$$\Downarrow$$

$$\min_x f_\rho(x) = f(x) + \frac{\rho}{2}\|h(x)\|_2^2 \qquad (P[\rho])$$

Simple fact: *Let (P) be feasible, the objective and the constraints in (P) be continuous and let f possess bounded level sets $\{x : f(x) \leq a\}$. Let, further X_* be the set of global solutions to (P). Then X_* is nonempty, approximations problems $(P[\rho])$ are solvable, and their global solutions approach X_* as $\rho \to \infty$:*

$$\forall \epsilon > 0 \exists \rho(\epsilon) : \rho \geq \rho(\epsilon), x_*(\rho) \quad \textit{solves } (P[\rho])$$
$$\Rightarrow \mathrm{dist}(x_*(\rho), X_*) \equiv \min_{x_* \in X_*} \|x_*(\rho) - x_*\|_2 \leq \epsilon$$

$$\min_x \{f(x) : h_i(x) = 0,\, i = 1, ..., k\} \qquad (P)$$

$$\Downarrow$$

$$\min_x f_\rho(x) = f(x) + \frac{\rho}{2}\|h(x)\|_2^2 \qquad (P[\rho])$$

Proof. **1^0.** By assumption, the feasible set of (P) is nonempty and closed, f is continuous and $f(x) \to \infty$ as $\|x\|_2 \to \infty$. It follows that f attains its minimum on the feasible set, and the set X_* of global minimizers of f on the feasible set is bounded and closed.

2^0. The objective in $(P[\rho])$ is continuous and goes to $+\infty$ as $\|x\|_2 \to \infty$; consequently, $(P[\rho])$ is solvable.

$$\min_x \{f(x) : h_i(x) = 0,\, i = 1, ..., k\} \qquad (P)$$
$$\Downarrow$$
$$\min_x f_\rho(x) = f(x) + \frac{\rho}{2}\|h(x)\|_2^2 \qquad (P[\rho])$$

3^0. It remains to prove that, for every $\epsilon > 0$, the solutions of $(P[\rho])$ with large enough value of ρ belong to ϵ-neighbourhood of X_*. Assume, on the contrary, that for certain $\epsilon > 0$ there exists a sequence $\rho_i \to \infty$ such that an optimal solution x_i to $(P[\rho_i])$ is at the distance $> \epsilon$ from X_*, and let us lead this assumption to contradiction.

$\diamondsuit$ Let f_* be the optimal value of (P). We clearly have

$$f(x_i) \leq f_{\rho_i}(x_i) \leq f_*, \tag{1}$$

whence $\{x_i\}$ is bounded. Passing to a subsequence, we may assume that $x_i \to \bar{x}$ as $i \to \infty$.

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \qquad (P)$$
$$\Downarrow$$
$$\min_x f_\rho(x) = f(x) + \tfrac{\rho}{2}\|h(x)\|_2^2 \qquad (P[\rho])$$

$$x_i \in \operatorname{Argmin}_x f_{\rho_i}(x), x_i \to \bar{x} \notin X_*$$
$$\Rightarrow f(x_i) \le f_{\rho_i}(x_i) \le f_* \qquad (1)$$

$\diamondsuit$ We claim that $\bar{x} \in X_*$, which gives the desired contradiction. Indeed,
— $\bar{x}$ is feasible, since otherwise

$$\lim_{i\to\infty} \underbrace{[f(x_i) + \frac{\rho_i}{2}\|h(x_i)\|_2^2]}_{f_{\rho_i}(x_i)}$$
$$= \quad f(\bar{x}) + \lim_{i\to\infty} \frac{\rho_i}{2} \underbrace{\|h(x_i)\|_2^2}_{\to\|h(\bar{x})\|_2^2>0} = +\infty,$$

in contradiction to (1);
— $f(\bar{x}) = \lim_{i\to\infty} f(x_i) \le f_*$ by (1); since $\bar{x}$ is feasible for (P), we conclude that $\bar{x} \in X_*$.

♠ **Shortcoming of Simple Fact:** *In non-convex case, we cannot find/approximate* **global** *minimizers of the penalized objective, so that Simple Fact is "unsubstantial"...*

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \qquad (P)$$

$$\Downarrow$$

$$\min_x f_\rho(x) = f(x) + \frac{\rho}{2}\|h(x)\|_2^2 \qquad (P[\rho])$$

Theorem. *Let x_* be a nondegenerate locally optimal solution to (P), i.e., a feasible solution such that*

$\diamondsuit$ *f, h_i are twice continuously differentiable in a neighbourhood of x_*,*

$\diamondsuit$ *the gradients of the constraints taken at x_* are linearly independent,*

$\diamondsuit$ *at x_*, the Second Order Sufficient Optimality condition is satisfied, so that x_* is the best, in terms of the objective, among nearby feasible solutions.*

Then there exists a neighbourhood V of x_* and $\bar{\rho} > 0$ such that

$\diamondsuit$ *for every $\rho \geq \bar{\rho}$, f_ρ possesses in V exactly one critical point $x_*(\rho)$;*

$\diamondsuit$ *$x_*(\rho)$ is a nondegenerate local minimizer of f_ρ and global minimizer of f_ρ on V;*

$\diamondsuit$ *$x_*(\rho) \to x_*$ as $\rho \to \infty$.*

In addition,

- *The local "penalized optimal value"*

$$f_\rho(x_*(\rho)) = \min_{x \in V} f_\rho(x)$$

is nondecreasing in ρ

Indeed, $f_\rho(\cdot) = f(\cdot) + \frac{\rho}{2}\|h(\cdot)\|_2^2$ grows with ρ

- *The constraint violation $\|h(x_*(\rho))\|_2$ monotonically goes to 0 as $\rho \to \infty$*

Indeed, let $\rho'' > \rho'$, and let $x' = x_*(\rho')$, $x'' = x_*(\rho'')$. Then

$$\begin{array}{l}
f(x') + \frac{\rho''}{2}\|h(x')\|_2^2 \geq f(x'') + \frac{\rho''}{2}\|h(x'')\|_2^2 \\
f(x'') + \frac{\rho'}{2}\|h(x'')\|_2^2 \geq f(x') + \frac{\rho'}{2}\|h(x')\|_2^2 \\
\Rightarrow f(x') + f(x'') + \frac{\rho''}{2}\|h(x')\|_2^2 + \frac{\rho'}{2}\|h(x'')\|_2^2 \\
\quad \geq f(x') + f(x'') + \frac{\rho''}{2}\|h(x'')\|_2^2 + \frac{\rho'}{2}\|h(x')\|_2^2 \\
\Rightarrow \frac{\rho''-\rho'}{2}\|h(x')\|_2^2 \geq \frac{\rho''-\rho'}{2}\|h(x'')\|_2^2
\end{array}$$

- *The true value of the objective $f(x_*(\rho))$ at $x_*(\rho)$ is nondecreasing in ρ*

Explanation: $x_*(\rho)$ is "super-optimal:" $f(x_*(\rho)) \leq f(x_*)$, with super-optimality achieved at the price of violating the constraints. As the penalty ρ goes to ∞, the constraint violation $\|h(x_*(\rho))\|_2$ and "super-optimality" $f(x_*) - f(x_*(\rho))$ monotonically go to 0.

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \qquad (P)$$

• The quantities $\rho h_i(x_*(\rho))$ converge to optimal Lagrange multipliers μ_i^* associated locally optimal solution x_*.

Indeed,

$$0 = f'_\rho(x_*(\rho)) = f'(x_*(\rho)) + \sum_i (\rho h_i(x_*(\rho)))h'_i(x_*(\rho)).$$

while

$$0 = f'(x_*) + \sum_i \mu_i^* h'_i(x_*) \ \& \ \lim_{\rho\to\infty} x_*(\rho) = x_*$$

$\Rightarrow$ *If not all optimal Lagrange multipliers μ_i^* for x_* are zeros, the violations of (some of) constraints at $x_*(\rho)$ are of order of $1/\rho$*

$\Rightarrow$ *To get small constraint violations, we <u>must</u> work with large penalties ρ !*

♣ Solving penalized problem

$$\min_x f_\rho(x) \equiv f(x) + \frac{\rho}{2}\|h(x)\|_2^2 \qquad (P[\rho])$$

◇ *In principle*, one can solve $(P[\rho])$ by whatever method for unconstrained minimization.

◇ **However:** *The conditioning of f deteriorates as $\rho \to \infty$.*

Indeed, as $\rho \to \infty$, we have

$$d^T f''_\rho(\underbrace{x_*(\rho)}_{x})d = d^T \underbrace{\left[f''(x) + \sum_i \rho h_i(x) h''_i(x)\right]}_{\to \nabla^2_x L(x_*,\mu^*)} d + \underbrace{\rho \sum_i (d^T h'_i(x))^2}_{\substack{\to \infty,\, \rho \to \infty \\ \text{except for } d^T h'(x_*) = 0}}$$

⇒ slowing down the convergence and/or severe numerical difficulties when working with large penalties...

Barrier Methods

$$\min_x \{f(x) : x \in G \equiv \{x : g_j(x) \le 0,\ j = 1, ..., m\}\} \qquad (P)$$

$$\Downarrow$$

$$\min_{x \in \operatorname{int} G} F^\rho(x) \equiv f(x) + \tfrac{1}{\rho} F(x) \qquad (P[\rho])$$

F is *interior penalty* for $G = \mathrm{cl}(\operatorname{int} G)$:

$\diamondsuit$ F is smooth on $\operatorname{int} G$

$\diamondsuit$ F tends to ∞ along every sequence $x_i \in \operatorname{int} G$ converging to a boundary point of G.

<u>Theorem.</u> *Assume that $G = \mathrm{cl}(\operatorname{int} G)$ is bounded and f, g_j are continuous on G. Then the set X_* of optimal solutions to (P) and the set $X_*(\rho)$ of optimal solutions to $(P[\rho])$ are nonempty, and the second set converges to the first one as $\rho \to \infty$: for every $\epsilon > 0$, there exists $\rho = \rho(\epsilon)$ such that*

$$\rho \ge \rho(\epsilon), x_*(\rho) \in X_*(\rho) \Rightarrow \mathrm{dist}(x_*(\rho), X_*) \le \epsilon.$$

♣ In the case of **<u>convex</u>** program

$$\min_{x\in G} f(x) \qquad (P)$$

with closed and bounded convex G and convex objective f, the domain G can be in many ways equipped with a twice continuously differentiable *strongly convex* penalty $F(x)$.

♠ Assuming f twice continuously differentiable on $\operatorname{int} G$, the aggregate

$$F_\rho(x) = \rho f(x) + F(x)$$

is strongly convex on $\operatorname{int} G$ and therefore attains its minimum at a single point

$$x_*(\rho) = \operatorname*{argmin}_{x\in \operatorname{int} G} F_\rho(x) \qquad [= \operatorname*{argmin}_{x\in \operatorname{int} G} F^\rho(x) := f(x) + \tfrac{1}{\rho}F(x)]$$

♠ It is easily seen that the *path* $x_*(\rho)$ is continuously differentiable and converges, as $\rho \to \infty$, to the optimal set of (P).

$$\min_{x\in G} f(x) \qquad (P)$$

$$\Downarrow$$

$$\min_{x\in \operatorname{int} G} F_\rho(x) = \rho f(x) + F(x) \qquad (P[\rho])$$

$$\Downarrow$$

$$x_*(\rho) = \operatorname{argmin}_{x\in \operatorname{int} G} F_\rho(x) \underset{\rho\to\infty}{\to} \operatorname{Argmin}_G f$$

♣ In *classical path-following scheme (Fiacco and McCormic, 1967)*, one traces the path $x_*(\rho)$ as $\rho \to \infty$ according to the following generic scheme:

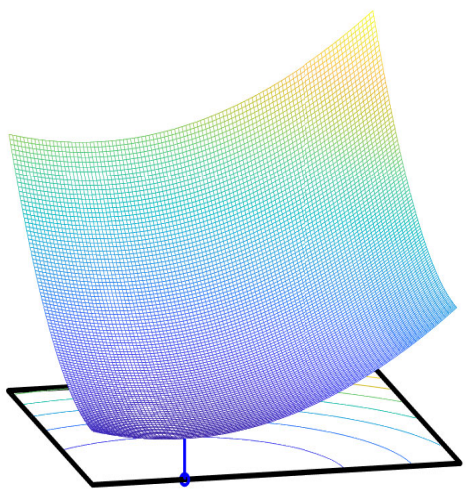
♢ Given $(x_i \in \operatorname{int} G, \rho_i > 0)$ with x_i close to $x_*(\rho_i)$,

— update ρ_i into a larger value ρ_{i+1} of the penalty

— minimize $F_{\rho_{i+1}}(\cdot)$, x_i being the starting point, until a new iterate x_{i+1} close to

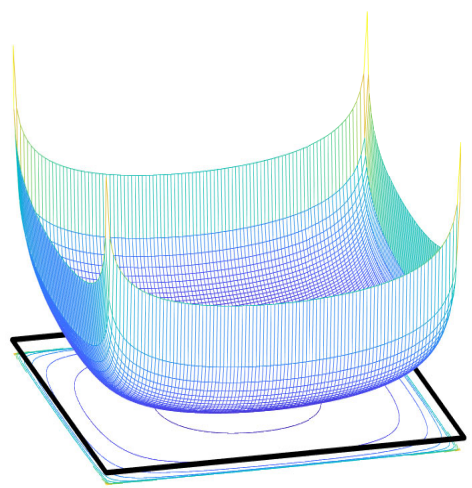
$$x_*(\rho_{i+1}) = \operatorname{argmin}_{x\in \operatorname{int} G} F_{\rho_{i+1}}(x)$$

is built, and loop.

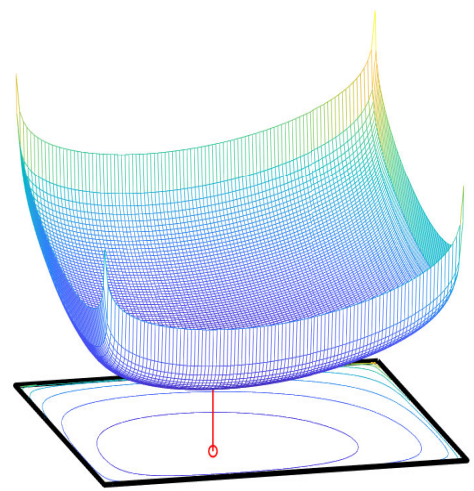


$f(x)$

blue dot: $x_* = \underset{x \in G}{\operatorname{argmin}}\, f(x)$



$F(x)$



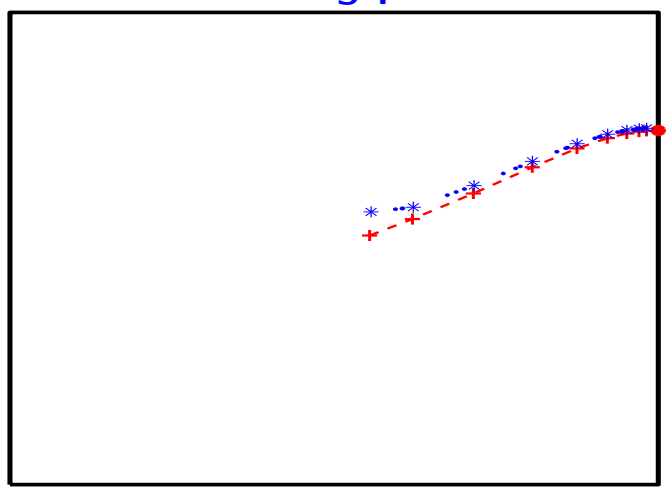
$F_{10}(x) = 10 f(x) + F(x)$

red dot: $x_*(10) = \underset{x \in \operatorname{int} G}{\operatorname{argmin}}\, F_{10}(x)$

G : black 2D rectangle

♠ To update a tight approximation x_i of $\operatorname{argmin} F_{\rho_i}(x)$ into a tight approximation x_{i+1} of $\operatorname{argmin} F_{\rho_{i+1}}(x)$, one can apply to $F_{\rho_{i+1}}(\cdot)$ a method for "essentially unconstrained" minimization, preferably, the Newton method
♠ When Newton method is used, one can try to increase penalty at a "safe" rate, keeping x_i in the domain of quadratic convergence of the Newton method as applied to $F_{\rho_{i+1}}(\cdot)$ and thus making use of fast local convergence of the method.

Tracing path



black rectangle	feasible domain G
bullet $\bullet$	optimal solution
dashed line	path $x_*(\rho) = \operatorname{argmin}_{\operatorname{int} G}[\rho f(x) + F(x)]$
pluses $+$	"target" points $x_*(\rho_i)$ on the path
austericks $*$	approximations x_i to $x_*(\rho_i)$ built by path tracing
dots .	iterates of Newton minimization of $F_{\rho_i}(\cdot)$ updating x_{i-1} into x_i

♠ To update a tight approximation x_i of $\operatorname{argmin} F_{\rho_i}(x)$ into a tight approximation x_{i+1} of $\operatorname{argmin} F_{\rho_{i+1}}(x)$, one can apply to $F_{\rho_{i+1}}(\cdot)$ a method for "essentially unconstrained" minimization, preferably, the Newton method

♠ When Newton method is used, one can try to increase penalty at a "safe" rate, keeping x_i in the domain of quadratic convergence of the Newton method as applied to $F_{\rho_{i+1}}(\cdot)$ and thus making use of fast local convergence of the method.

Questions: • How to choose F?

• How to measure closeness to the path?

• *How to ensure "safe" penalty updating without slowing the method down?*

Note: As $\rho \to \infty$, the condition number of $F''_\rho(x_*(\rho))$ may blow up to ∞, which, according to the traditional theory of the Newton method, makes the problems of updating x_i into x_{i+1} more and more difficult. Thus, slowing down seems to be unavoidable...

♣ In late 80's, it was discovered that *the classical path-following scheme, associated with properly chosen barriers, admits "safe" implementation without slowing down*. This discovery led to invention of *Polynomial Time Interior Point methods* for convex programs.

♣ Majority of Polynomial Time Interior Point methods heavily exploit the classical path-following scheme; the novelty is in what are the underlying barriers – these are specific *self-concordant* functions especially well suited for Newton minimization.

♠ When speaking about Newton method as applied to a strongly convex smooth function f, we saw that

— the algorithm, started close to the global minimizer x_*, converges to x_* quadratically

— the algorithm is *affine invariant*: passing from $f(x)$ to $g(y) = f(Ay+b)$, with invertible A, applying the Newton algorithm to $g(y)$, and translating the resulting trajectory y_t into x-coordinates: $y_t \mapsto x_t = Ay_t + b$, *we get exactly the trajectory we would get when applying the Newton method to $f(x)$ directly.*

♠ In spite of the affine invariance of the algorithm, the classical description of the region of quadratic convergence of Newton method as applied to smooth strongly convex $f(x)$ is frame-dependent. It is expressed in terms of the largest and the smallest eigenvalues of $f''(x_*)$ and the magnitude of Lipschitz constant of $f''(x)$ and is <u>not</u> affine invariant – if A is not orthogonal, it may happen that this description when translated from x-variables to y-variables specifies domain much larger, or much smaller, than the same description as applied to $g(y)$ directly.

♠ **Question:** *Where to take "good coordinates" (scientifically: good Euclidean structure) to describe qualitatively in affine invariant fashion the behaviour of the Newton method as applied to strongly convex smooth function f ?*

♠ **Answer:** The Hessian of f at a point x defines Euclidean structure $\langle g, h\rangle_x = g^T f''(x)h$ and Euclidean norm

$$\|h\|_x = \sqrt{h^T f''(x)h} = \sqrt{\left.\frac{d^2}{dt^2}\right|_{t=0} f(x+th)}.$$

In coordinates orthonormal in this Euclidean structure $f''(x)$ becomes as good as it could be – just the unit matrix. Imposing an upper bound on the third directional derivative of f, taken at x, in terms of the $\|\cdot\|_x$-norm of the direction, we arrive at a family of strongly convex smooth objectives perfectly well suited for Newton minimization. *On this family, the behavior of Newton method, including description of its domain of quadratic convergence, becomes quite transparent and frame-independent!*

♣ Let $G \subset \mathbb{R}^n$ be a closed convex domain with nonempty interior which does not contain lines. A 3 times continuously differentiable convex function

$$F(x) : \operatorname{int} G \to \mathbb{R}$$

is called *self-concordant*, if

◇ F is an interior penalty for G: $x_i \in \operatorname{int} G, x_i \to x \in \partial G \Rightarrow F(x_i) \to \infty$

◇ F satisfies the relation

$$\forall (x \in \operatorname{int} G, h \in \mathbb{R}^n) : \left| \frac{d^3}{dt^3}\Big|_{t=0} F(x+th) \right| \leq 2 \underbrace{\left(\frac{d^2}{dt^2}\Big|_{t=0} F(x+th) \right)^{3/2}}_{\|h\|_x^3} \qquad (*)$$

Equivalently: The third order directional derivative taken at $x \in \operatorname{int} G$ along any direction h of *unit* $\|\cdot\|_x$-length, i.e., such that $\frac{d^2}{dt^2}\Big|_{t=0} F(x+th) = 1$, does not exceed 2.

Standard example: *$F(x) = -\ln(x)$ is self-concordant on $G = \mathbb{R}_+$.* In this case $(*)$ becomes identity.

Extension: *Assume domain $G = \operatorname{cl}\{x \in \mathbb{R}^n : a_i^T x < b_i, i \leq m\}$ is nonempty and does not contain lines. Then the function $F(x) = -\sum_{i=1}^m \ln(b_i - a_i^T x)$ is self-concordant on G.*

$$\forall(x \in \operatorname{int} G, h \in \mathbb{R}^n) : \left|\frac{d^3}{dt^3}\Big|_{t=0} F(x+th)\right| \leq 2 \underbrace{\left(\frac{d^2}{dt^2}\Big|_{t=0} F(x+th)\right)^{3/2}}_{\|h\|_x^3} \qquad (*)$$

Note: 3/2 in $(*)$ is a must — both sides in $(*)$ should be of the same degree of homogeneity in h.

Note: There is nothing special in factor 2 in front of $(...)^{3/2}$ in the right hand side of $(*)$ – it is just a convenient normalization.

Indeed, the sides of $(*)$ are of *different degree of homogeneity w.r.t.* F, so that scaling F, we can make this factor whatever we want (or, equivalently, can convert factor 2 into whatever constant factor we want, same as can convert whatever constant factor in front of $(...)^{3/2}$ into factor 2).

Note: Convenience of constant 2 stems from the fact that with this constant in (*) the important barriers

- $-\ln(x)$ for $\mathbb{R}_+$, and $-\sum_i \ln(b_i - a_i^T x)$ for polytope $\{x : a_i^T x \leq b_i, i \leq m\}$,
- $-\ln(x_m^2 - x_1^2 - x_2^2 - ... - x_{m-1}^2)$ for the Lorentz cone $\mathbf{L}^m = \{x \in \mathbb{R}^m : x_m \geq \sqrt{\sum_{i=1}^{m-1} x_i^2}\}$,
- $-\ln \operatorname{Det} X$ for the cone $\mathbf{S}_+^m$ of positive semidefinite $m \times m$ matrices X

become self-concordant "as is," without scaling.

Newton Method on Self-Concordant functions

♠ Let G. $\operatorname{int} G \neq \emptyset$, be a closed convex domain not containing lines, and $F(x) : \operatorname{int} G \to \mathbb{R}$ be self-concordant. Then $F''(x) \succ 0$, $x \in \operatorname{int} G$. Defining the *Newton decrement* of F at a point $x \in \operatorname{int} G$ as

$$\lambda(x, F) = \max_h \left\{ \frac{d}{dt}\Big|_{t=0} F(x + th) : \|h\|_x \leq 1 \right\} \quad \left[= \sqrt{F'(x)[F''(x)]^{-1}F'(x)} \right]$$

and *Damped Newton iterate of* x as the point

$$x_+ = x_+(x) = x - \frac{1}{1+\lambda(x,F)}[F''(x)]^{-1}F'(x)$$

one has:

- $x_+ \in \operatorname{int} G$ and $F(x_+) \leq F(x) - [\lambda - \ln(1 + \lambda)] \leq F(x) - \frac{\lambda^2}{2(1+\lambda)}$, $\lambda = \lambda(x, F)$;
- F attains its minimum on $\operatorname{int} G$ iff $\lambda(x, F) < 1$ for some x, and

$$\lambda := \lambda(x, F) < 1 \Rightarrow F(x) - \min_{\operatorname{int} G} F \leq -\log(1 - \lambda) - \lambda \leq \frac{\lambda^2}{2(1-\lambda)}$$

- Region of fast convergence of Damped Newton method $x_t \mapsto x_{t+1} = x_+(x_t)$ is given by $\lambda_0 := \lambda(x_0, F) < 1$. When $\lambda_0 < 1$, it takes $T \leq O(1)/(1 - \lambda_0)$ steps to get $\lambda(x_T, F) < 0.1$, and $t \geq T \Rightarrow \lambda(x_{t+1}, F) \leq 2\lambda^2(x_t, F) \leq \frac{\lambda(x_t,F)}{2}$ & $\|x_{t+1} - x_*\|_{x_*} \leq 2\|x_t - x_*\|_{x_*}^2 \leq \frac{\|x_t - x_*\|_{x_*}}{2}$, where $x_* = \operatorname{Argmin}_{\operatorname{int} G} F$.

♠ Let $\vartheta \geq 1$. F is called *ϑ-self-concordant barrier* for G, if, in addition to being self-concordant on G, F satisfies the relation

$$\left|\frac{d}{dt}\Big|_{t=0} F(x+th)\right| \leq \sqrt{\vartheta}\left(\frac{d^2}{dt^2}\Big|_{t=0} F(x+th)\right)^{1/2}$$

or, equivalently,

$$\lambda(x,F) \leq \sqrt{\vartheta}\ \forall x \in \operatorname{int} G.$$

ϑ is called the *parameter* of s.-c.b. F.

Examples:

• Assume domain $G = \mathrm{cl}\{x \in \mathbb{R}^n : a_i^T x < b_i, i \leq m\}$ is nonempty and does not contain lines. Then the function $F(x) = -\sum_{i=1}^m \ln(b_i - a_i^T x)$ is m-self-concordant barrier for G.

• The function $F(x) = -\ln(x_m^2 - x_1^2 - x_2^2 - \ldots - x_{m-1}^2)$ is 2-self-concordant barrier for the Lorentz cone $\mathbf{L}^m$

• The function $F(X) = -\ln \operatorname{Det} X$ is m-self-concordant barrier for the positive semidefinite cone $\mathbf{S}_+^m$

♣ Every convex program

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

can be converted into a convex program with *linear* objective, namely,

$$\min_{t,x} \{t : x \in G, f(x) \leq t\}.$$

Assuming that this transformation has been done at the very beginning, we can w.l.o.g. focus on convex program with *linear* objective

$$\min_{x \in G} c^T x \qquad (P)$$

$$\mathrm{Opt} = \min_{x \in G} c^T x \qquad (P)$$

♣ Assume that G is a closed and bounded convex set with a nonempty interior, and let F be a ϑ-s.c.b. barrier for G.

♢ **<u>Fact I:</u>** *The path*

$$x_*(\rho) = \operatorname*{argmin}_{x \in \mathrm{int}\, G} \left[F_\rho(x) = \rho c^T x + F(x) \right], \ \rho > 0$$

is well defined, and $\lambda(x_*(\rho), F_\rho) = 0 \Rightarrow \lambda(x, F_\rho)$ *can be thought of as a measure of proximity of* $x \in \mathrm{int}\, G$ *to* $x_*(\rho)$.

$$\mathrm{Opt} = \min_{x \in G} c^T x \qquad (P)$$

♣ Assume that G is a closed and bounded convex set with a nonempty interior, and let F be a ϑ-s.c.b. barrier for G and

$$F_\rho(x) = \rho c^T x + F(x)$$

♢ **<u>Fact II:</u>** *Assuming* $\lambda(x_0, F_{\rho_0}) \leq 0.1$, *consider path-following algorithm where*

- *penalty updating rule is* $\rho_{t-1} \mapsto \rho_t = \left(1 + \frac{\gamma}{\sqrt{\vartheta}}\right)\rho_{t-1}$
- x_t *is obtained from* x_{t-1} *by running on* $F_{\rho_t}(\cdot)$ *Damped Newton method, started at* x_{t-1}, *until an iterate with Newton decrement* ≤ 0.1 *is built; this iterate is taken as* x_t.

For this algorithm,

— *the number of Damped Newton steps in updating* $x_{t-1} \mapsto x_t$ *depends solely on* γ *and is just one when* $\gamma = 0.1$;

— *for all* t, *one has* $c^T x_t - \mathrm{Opt} \leq \frac{2\vartheta}{\rho_t} \leq \frac{2\vartheta}{\rho_0} \exp\{-\gamma t/\sqrt{\vartheta}\}$

$\Rightarrow$ *It takes* $O(\sqrt{\vartheta})$ *Newton steps to increase* ρ *by absolute constant factor and reduce inaccuracy* $c^T x - \mathrm{Opt}$ *by absolute constant factor!*

No slow down as $\rho \to \infty$!

♣ **<u>Fact III:</u>** Every convex domain $G \subset \mathbb{R}^n$ admits $O(n)$-s.-c.b. For typical feasible domains arising in Convex Programming, one can point out explicit "computable" s.-c.b.'s. For example,

♠ Let G be given by m convex quadratic constraints:

$$G = \{x : \underbrace{x^T A_j^T A_j x + 2b_j^T x + c_j}_{g_j(x)} \leq 0,\ 1 \leq j \leq m\}$$

satisfying the Slater condition. When G does not contain lines, the logarithmic barrier $F(x) = -\sum\limits_{j=1}^{m} \ln(-g_j(x))$ is m-s.-c.b. for G.

♠ Let A_i be $m \times m$ symmetric matrices and G be given by *Linear Matrix Inequality*

$$G = \{x : \underbrace{A_0 + x_1 A_1 + ... + x_n A_n}_{\mathcal{A}(x)} \succeq 0\}$$

satisfying the Slater condition: $\mathcal{A}(\bar{x}) \succ 0$ for some $\bar{x}$. When G does not contain lines, the log-det barrier $F(x) = -\ln \mathrm{Det}(\mathcal{A}(x))$ is m-s.-c.b. for G.

Primal-Dual Path Following Methods for LP

♣ **Preliminaries**

♠ m-self-concordant log-barrier $F(z) = -\sum_{i=1}^m \ln(a_i^T z - b)$ for polytope $\{z : Az - b \geq 0\}$ ($A = [a_1^T; ...; a_m^T]$, $\mathrm{Null}(A) = \{0\}$, $Az - b \geq 0$ strictly feasible) is

$$F(z) = \Phi(Az - b), \ \Phi(x) = -\sum_{i=1}^m \ln(x_i)$$

♠ **Facts:**

- $\Phi(x)$ is m-self-concordant barrier for $\mathbb{R}^m_+$
- When $t > 0$ and $x > 0$, we have

$$\Phi'(x) = -[1/x_1; ...; 1/x_m], \ \Phi'(tx) = t^{-1}\Phi'(x), \ x^T\Phi'(x) \equiv -m$$

- Nonlinear mapping $x \mapsto -\Phi'(x) = [1/x_1; ...; 1/x_m]$ is a one-to-one smooth mapping of $\mathrm{int}\,\mathbb{R}^m_+$ onto $\mathrm{int}\,\mathbb{R}^m_+$. This mapping is self-inverse:

$$x > 0, y = -\Phi'(x) \Leftrightarrow y > 0, x = -\Phi'(y) \Leftrightarrow x > 0, y > 0, x_s y_s = 1, s \leq m$$

and

$$\boxed{\rho > 0 \ \& \ x > 0 \ \& \ y = -\Phi'(x)/\rho} \Leftrightarrow \boxed{\rho > 0 \ \& \ y > 0 \ \& \ x = -\Phi'(y)/\rho}$$

♣ Consider an LP

$$\min_z \left\{ c^T z : Az - b \geq 0 \right\} \qquad (P)$$

with $m \times n$ matrix A, $\mathrm{Null}(A) = \{0\}$, along with the dual problem

$$\max_y \left\{ b^T y : A^T y = c,\ y \geq 0 \right\} \qquad (D)$$

and assume that both problems are strictly feasible:

$$\exists \bar{z} : A\bar{z} - b > 0 \ \& \ \exists \bar{y} > 0 : A^T \bar{y} = c$$

<u>Note:</u> Passing from z to "primal slack" $x = Az - b$, we can rewrite (P) as

$$\min_x \left\{ e^T x : x \geq 0, x \in L = \mathrm{Im} A - b \right\} \qquad (P')$$

where e is a vector satisfying $A^T e = c$, so that

$$e^T x = e^T (Az - b) = (A^T e)^T z - \mathrm{const} = c^T z - \mathrm{const}$$

$$\boxed{\begin{array}{rcl} & \min\limits_z \left\{c^T z : Az - b \geq 0\right\} & (P) \\ \Leftrightarrow & \min\limits_x \left\{e^T x : x + b \in \mathrm{Im}A, x \geq 0\right\} & (P') \end{array}}$$

$$\Downarrow$$

$$\boxed{\max\limits_y \left\{b^T y : \underbrace{A^T y = c \equiv A^T e}_{\Leftrightarrow y - e \in (\mathrm{Im}A)^\perp}, y \geq 0\right\} \quad (D)}$$

♠ Let $\Phi(x) = -\sum\limits_{i=1}^m \ln x_i$. Equipping the domain of (P) with m-s.c.b. $F(z) = \Phi(Az - b)$, consider

$$z_*(\rho) \;=\; \mathrm{argmin}_z[\rho c^T z + F(z)] = \mathrm{argmin}_z[\rho e^T(Az - b) + \Phi(Az - b)]$$

Observation: *The point $x_* = x_*(\rho) := Az_*(\rho) - b$ minimizes $\rho e^T x + \Phi(x)$ over the feasible set of (P'), i.e.,*

$$x_* > 0, \;\; x_* + b \in \mathrm{Im}A, \;\; \rho e + \Phi'(x_*) \in (\mathrm{Im}A)^\perp.$$

$\Rightarrow$ The point $y_* := y_*(\rho) := -\rho^{-1}\Phi'(x_*(\rho))$ satisfies

$$y_* > 0, \qquad \underbrace{y_* - e}_{=-[\rho e + \Phi'(x_*)]/\rho} \in (\mathrm{Im}A)^\perp, \; \underbrace{-\rho b + \Phi'(y_*)}_{=-\rho(x_* + b)} \in \mathrm{Im}A$$

[Note: as we know, $y_* = -\Phi'(x_*)/\rho \Leftrightarrow x_* = -\Phi'(y_*)/\rho$]

i.e., the point $y_*(\rho)$ minimizes $-\rho b^T y + \Phi(y)$ over the feasible set of (D).

♣ We arrive at a nice symmetric picture:

♣ The *primal central path* $x_* = x_*(\rho)$ minimizing the *primal aggregate*

$$\rho e^T x + \Phi(x) \qquad [\Phi(x) = -\sum_i \ln x_i]$$

over the primal feasible set is given by

$$x_* > 0, x_* + b \in \mathrm{Im}A, \rho e + \Phi'(x_*) \in (\mathrm{Im}A)^\perp$$

♣ The *dual central path* $y_* = y_*(\rho)$ minimizing the *dual aggregate*

$$-\rho b^T y + \Phi(y)$$

over the dual feasible set is given by

$$y_* > 0, y_* - e \in (\mathrm{Im}A)^\perp, -\rho b + \Phi'(y_*) \in \mathrm{Im}A$$

♣ The paths (together called the primal-dual central path $(x_*(\rho), y_*(\rho))$) are linked by

$$y_*(\rho) = -\rho^{-1}\Phi'(x_*(\rho)) \Leftrightarrow x_*(\rho) = -\rho^{-1}\Phi'(y_*(\rho)) \Leftrightarrow [x_*(\rho)]_s[y_*(\rho)]_s = \frac{1}{\rho} \forall s \le m.$$

$\Rightarrow$ *On the primal-dual path* $x = x_*(\rho)$, $y = y_*(\rho)$, *setting* $z = z_*(\rho)$, *so that* $x = Az - b$, *we have*

$$\mathrm{DualityGap}(x, y) := [c^T z - \mathrm{Opt}(P)] + [\mathrm{Opt}(D) - b^T y] = x^T y = m/\rho$$
$$\left[x = Az - b, \textit{whence } x^T y = [A^T y]^T z - b^T y = c^T z - b^T y\right]$$

— *we know exactly how the sum of non-optimalities of strictly feasible primal and dual solutions* $x_*(\rho)$, $y_*(\rho)$ *in the respective problems* (P'), (D) *goes to* 0 *as* $\rho \to \infty$ *!*

$$\min_z \left\{c^T z : Az - b \geq 0\right\} \qquad (P)$$

$$\Leftrightarrow \quad \min_x \left\{e^T x : x + b \in \mathrm{Im}A, x \geq 0\right\} \qquad (P')$$

$$\Downarrow$$

$$\max_y \left\{b^T y : \underbrace{A^T y = c \equiv A^T e}_{\equiv y - e \in (\mathrm{Im}A)^\perp}\right\} \qquad (D)$$

♣ **Generic Primal-Dual Interior Point Method for LP** is obtained by tracing the primal-dual central path:

♢ Given current iterate — primal-dual strictly feasible pair x^i, y^i and value ρ_i of penalty, update it into new iterate $x^{i+1}, y^{i+1}, \rho_{i+1}$ by

♢ Updating $\rho_i \mapsto \rho_{i+1} \geq \rho_i$

♢ Applying a Newton step to the system

$$x > 0, \ x + b \in \mathrm{Im}A; \quad y > 0, \ y - e \in (\mathrm{Im}A)^\perp$$

$$\mathrm{Diag}\{x\}y = \frac{1}{\rho_{i+1}} \underbrace{(1, ..., 1)^T}_{\mathbf{e}} \left[\Leftrightarrow x_s y_s = \frac{1}{\rho_{i+1}}, \ 1 \leq s \leq m\right]$$

defining the primal-dual central path, i.e., linearizing at x^i, y^i the nonlinear constraints $x_s y_s = \frac{1}{\rho_{i+1}}$ and passing to the solution of the resulting *linear* system.

$$x > 0,\ x + b \in \mathrm{Im}A;\quad y > 0,\ y - e \in (\mathrm{Im}A)^\perp$$
$$\mathrm{Diag}\{x\}y = \frac{1}{\rho_{i+1}}\underbrace{[1;...;1]}_{\mathbf{e}}\left[\Leftrightarrow x_s y_s = \frac{1}{\rho_{i+1}},\ 1 \le s \le m\right]$$

- Newton step as applied to the system results in

$$x^{i+1} = x^i + \Delta x,\ y^{i+1} = y^i + \Delta y$$

where $\Delta x, \Delta y$ solve the linear system

$$\Delta x \in \mathrm{Im}A,\ \Delta y \in (\mathrm{Im}A)^\perp,$$
$$\mathrm{Diag}\{x^i\}y^i + \mathrm{Diag}\{x^i\}\Delta y + \mathrm{Diag}\{y^i\}\Delta x = \frac{\mathbf{e}}{\rho_{i+1}}$$
$$\underbrace{\left[\Leftrightarrow x_s^i y_s^i + x_s^i \cdot [\Delta y]_s + y_s^i \cdot [\Delta x]_s = \frac{1}{\rho_{i+1}},\ 1 \le s \le m\right]}_{\text{linearization of the nonlinear system } [x^i + \Delta x]_s[y^i + \Delta y]_s = \frac{1}{\rho_{i+1}}, s \le m, \text{ in } \Delta x,\ \Delta y}$$

$$
\begin{array}{rll}
 & \min\limits_z \left\{ c^T z : Az - b \geq 0 \right\} & (P) \\
\Leftrightarrow & \min\limits_x \left\{ e^T x : x + b \in \mathrm{Im} A, x \geq 0 \right\} & (P') \\
 & \Downarrow & \\
 & \max\limits_y \left\{ b^T y : \underbrace{A^T y = c \equiv A^T e}_{\equiv y - e \in (\mathrm{Im} A)^\perp} \right\} & (D)
\end{array}
$$

♣ The classical path-following scheme as applied to (P) and the m-s.c.b. $F(z) = \Phi(Az - b)$ allows to trace the path $z_*(\rho)$ (and thus the primal central path $x_*(\rho) = Az_*(\rho) - b$).

More advanced *primal-dual* path-following methods *simultaneously trace the primal and the dual central paths, staying close* (in certain precise sense) *to it*, which results in algorithmic schemes with better practical performance than the one of the "purely primal" scheme.

♣ Both approaches, with proper implementation, result in the best known so far theoretical complexity bounds for LP. According to these bounds, the "arithmetic cost" of generating ϵ-solution to a primal-dual pair of strictly feasible LP's with $m \times n$ matrix A is

$$O(1)mn^2 \ln\left(\frac{mn\Theta}{\epsilon}\right)$$

operations, where $O(1)$ is an absolute constant and Θ is a data-dependent constant.

♣ In practice, properly implemented primal-dual methods by far outperform the purely primal ones and solve in few tens of Newton iterations real-world LPs with tens and hundreds of thousands of variables and constraints. *In modern commercial LP solvers, primal-dual path-following is the default choice...*

♣ Primal-dual path-following methods are developed and routinely used for general *conic* problems on "nice" cones, e.g., Second Order Conic programs and Semidefinite programs (whatever it means...)

Lecture 12:

Algorithms for Constrained Optimization, II: Augmented Lagrangians

Augmented Lagrangian methods

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \qquad (P)$$

♣ **Shortcoming of penalty scheme:** in order to solve (P) to high accuracy, one should work with large values of penalty, which makes the penalized objective

$$f_\rho(x) = f(x) + \frac{\rho}{2}\|h(x)\|_2^2$$

difficult to minimize.

♠ **Augmented Lagrangian methods** use the penalty mechanism in a "smart way," which allows to avoid the necessity to work with very large values of ρ.

Local Lagrange Duality

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \qquad (P)$$

♣ Let x_* be a nondegenerate local solution to (P), so that there exists μ^* such that

$$\begin{array}{ll} (a) & \nabla_x L(x_*, \mu^*) = 0 \\ (b) & d^T \nabla_x^2 L(x_*, \mu^*) d > 0 \ \forall d \in T_{x_*} \backslash \{0\} \end{array}$$

$$\left[\begin{array}{c} L(x, \mu) = f(x) + \sum_i \mu_i h_i(x) \\ T_{x_*} = \{d : d^T h_i'(x_*) = 0,\ i = 1, ..., k\} \end{array} \right]$$

♠ Assume for the time being that instead of (b), a stronger condition hods true:

(!) *the entire matrix* $\nabla_x^2 L(x_*, \mu^*)$ *is positive definite*

♣ Under assumption (!), x_* is a nondegenerate unconstrained local minimizer of the smooth function $L(x, \mu^*)$ of x and as such can be found by methods for unconstrained minimization.

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \qquad (P)$$

♠ **Intermediate Summary:** *If*

♢ *(a) we are clever enough to guess the vector μ^* of Lagrange multipliers,*

♢ *(b) we are lucky to have $\nabla_x^2 L(x_*, \mu^*) \succ 0$,*

then x_* can be found by unconstrained optimization technique.

♠ **How to become smart when being lucky: Local Lagrange Duality**.

Situation: x_* is a nondegenerate locally optimal solution to

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \qquad (P)$$

and we are lucky:

$$\exists \mu^* : \nabla_x L(x_*, \mu^*) = 0,\ \nabla_x^2 L(x_*, \mu^*) \succ 0 \qquad (!)$$

Fact: *Under assumption (!), there exist an open convex neighbourhood V of x_* and an open convex neighbourhood $\mathcal{M}$ of μ^* such that*
(i) For every $\mu \in \mathcal{M}$, function $L(x, \mu)$ is strongly convex in $x \in V$ and possesses uniquely defined critical point $x_*(\mu)$ in V which is continuously differentiable in $\mu \in \mathcal{M}$. $x_*(\mu)$ is a nondegenerate local minimizer of $L(\cdot, \mu)$;
(ii) The function

$$\underline{\mathbf{L}}(\mu) = L(x_*(\mu), \mu) = \min_{x \in V} L(x, \mu)$$

is C^2-smooth and concave in $\mathcal{M}$,

$$\underline{\mathbf{L}}'(\mu) = h(x_*(\mu)),$$

and μ_* is a nondegenerate maximizer of $\underline{\mathbf{L}}(\mu)$ on $\mathcal{M}$.

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \quad (P)$$
$$\Rightarrow \qquad L(x, \mu) = f(x) + \sum_i \mu_i h_i(x)$$

<u>Situation:</u> $\nabla_x L(x_*, \mu^*) = 0$, $\nabla_x^2 L(x_*, \mu^*) \succ 0$

$$\begin{array}{rcl} \mu^* & = & \text{argmax}_{\mu \in \mathcal{M}} \left[\underline{\mathbf{L}}(\mu) := \min_{x \in V} L(x, \mu) \right] \\ x_* & = & \text{argmin}_{x \in V} L(x, \mu^*) \end{array}$$

$\Rightarrow$ We can solve (P) by maximizing $\underline{\mathbf{L}}(\mu)$ over $\mu \in \mathcal{M}$ by a first order method for "essentially unconstrained" minimization.

The first order information on $\underline{\mathbf{L}}(\mu)$ required by the method can be obtained by solving auxiliary "essentially unconstrained" problems

$$x_*(\mu) = \operatorname*{argmin}_{x \in V} L(x, \mu)$$

via

$$\begin{array}{rcl} \underline{\mathbf{L}}(\mu) & = & L(x_*(\mu), \mu) \\ \underline{\mathbf{L}}'(\mu) & = & h(x_*(\mu)) \end{array}$$

<u>Note:</u> In this scheme, there are no "large parameters"!

<u>However:</u> How to ensure luck?

♣ How to ensure luck: convexification by penalization

Observe that the problem of interest

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \qquad (P)$$

for every $\rho \geq 0$ is **exactly equivalent** to

$$\min_x \left\{ f_\rho(x) = f(x) + \frac{\rho}{2}\|h(x)\|_2^2 : h_i(x) = 0, i \leq k \right\} \qquad (P_\rho)$$

It turns out that

(!) *If x_* is a nondegenerate locally optimal solution of (P) and ρ is large enough, then x_* is a locally optimal and "lucky" solution to (P_ρ).*

$\Rightarrow$ We can solve (P) by applying the outlined "primal-dual" scheme to (P_ρ), provided that ρ is appropriately large!

Note: *Although in our new scheme we do have penalty parameter which should be "large enough", we still have an advantage over the straightforward penalty scheme: in the latter, ρ should go to ∞ as $O(1/\epsilon)$ as required inaccuracy ϵ of solving (P) goes to 0, while in our new scheme* **a single** *"large enough" value of ρ will do!*

Problem

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \qquad (P)$$

or every $\rho \geq 0$ is **exactly equivalent** to

$$\min_x \left\{ f_\rho(x) = f(x) + \frac{\rho}{2}\|h(x)\|_2^2 : h_i(x) = 0, i \leq k \right\} \qquad (P_\rho)$$

(!) *If x_* is a nondegenerate locally optimal solution of (P) and ρ is large enough, then x_* is a locally optimal and "lucky" solution to (P_ρ).*

Verification of (!) boils down to verifying he following fact:

- We are given a *positive definite* $m \times m$ matrix Δ and a symmetric $n \times n$ matrix split into blocks: $A = \left[\begin{array}{c|c} P & S \\ \hline S^T & R \end{array}\right]$ with $m \times m$ block P. We want to find a positive ρ such that the matrix $A[\rho] = \left[\begin{array}{c|c} P + \rho\Delta & S \\ \hline S^T & R \end{array}\right]$ is positive definite.

Note: If $A[\rho] \succ 0$, then clearly $A[\rho'] \succ 0$ when $\rho' \geq \rho$, due to $A[\rho'] \succeq A[\rho]$.
Question: *When our goal is achievable?*
Answer: *Our goal is achievable if and only if R is positive definite.*
Necessity: All principal submatrices in a positive definite matrix are positive definite. R is a principal submatrix of every one of ther matrices $A[\rho]$, so that if our goal is achievable, R must be positive definite.

• We are given a *positive definite* $m \times m$ matrix Δ and a symmetric $n \times n$ matrix split into blocks: $A = \left[\begin{array}{c|c} P & S \\ \hline S^T & R \end{array}\right]$ with $m \times m$ block P. We want to find a positive ρ such that the matrix $A[\rho] = \left[\begin{array}{c|c} P + \rho\Delta & S \\ \hline S^T & R \end{array}\right]$ is positive definite.

Claim: *Our goal is achievable if and only if R is positive definite.*
Sufficiency follows from extremely important by its own right

Schur Complement Lemma: *Symmetric block matrix*

$$B = \left[\begin{array}{c|c} E & S \\ \hline S^T & R \end{array}\right]$$

with positive definite R is positive (semi)definite if and only if the matrix

$$E - SR^{-1}S^T \qquad (*)$$

is positive (semi)definite.
SCL⇒ Sufficiency: When $R \succ 0$ and $B = A[\rho]$, matrix $(*)$ becomes

$$P + \rho\Delta - SR^{-1}S^T. \qquad (\#)$$

Since $\Delta \succ 0$, matrix $(\#)$ is positive definite for all large enough values of ρ, implying by SCL that $A[\rho] \succ 0$ for large ρ.

Schur Complement Lemma: *Symmetric block matrix*

$$B = \left[\begin{array}{c|c} E & S \\ \hline S^T & R \end{array}\right]$$

with positive definite R is positive (semi)definite if and only if the matrix

$$E - SR^{-1}S^T \qquad (*)$$

is positive (semi)definite.

Proof. B is positive semidefinite if and only if the quadratic form

$$[u; v]^T B[u; v] = u^T Eu + 2u^T Sv + v^T Rv$$

of $[u; v] \in \mathbb{R}^m \times \mathbb{R}^{n-m}$ is everywhere nonnegative, or, which is the same, if

$$\min_v [u^T Eu + 2u^T Sv + v^T Rv] \geq 0 \ \forall u.$$

Since $R \succ 0$, $\min_v$ is achieved when $0 = \nabla_v [u^T Eu + 2u^T Sv + v^T Rv] = 2[S^T u + Rv]$, resulting in $v = -R^{-1}S^T u$ and

$$\min_v [u^T Eu + 2u^T Sv + v^T Rv] = u^T Eu - 2u^T SR^{-1}S^T u + u^T SR^{-1}S^T u = u^T [E - SR^{-1}S^T]u.$$

Thus, the minimum in question is ≥ 0 for all u if and only if the matrix $E - SR^{-1}S^T$ is positive semidefinite.

The same reasoning (where one should replace "nonnegative" with "positive whenever $u \neq 0$") shows that B is positive definite if and only if $E - SR^{-1}S^T$ is so. $\square$

$$\min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., k\} \qquad (P)$$

$$\Updownarrow$$

$$\min_x \left\{ f_\rho(x) = f(x) + \tfrac{\rho}{2}\|h(x)\|_2^2 : \begin{array}{c} h_i(x) = 0, \\ i \le k \end{array} \right\} \quad (P_\rho)$$

Let

$$L_\rho(x, \mu) = f(x) + \frac{\rho}{2}\|h(x)\|_2^2 + \sum_i \mu_i h_i(x)$$

be the Lagrange function of (P_ρ); the Lagrange function of (P) is then $L_0(x, \mu)$. Given nondegenerate locally optimal solution x_* to (P), let μ^* be the corresponding Lagrange multipliers.

Claim: *When $\rho > 0$ is large enough, one has $\nabla^2 L_\rho(x_*, \mu^*) \succ 0$.*

Justifying the claim. We have

$$\begin{array}{rcl}
\nabla_x L_\rho(x_*, \mu^*) & = & \nabla_x L_0(x_*, \mu^*) + \rho \sum_i h_i(x_*) h_i'(x_*) = \nabla_x L_0(x_*, \mu^*) = 0 \\
\nabla_x^2 L_\rho(x_*, \mu^*) & = & \nabla_x^2 L(x_*, \mu^*) + \rho \sum_i h_i(x_*) h_i''(x_*) + \rho \sum_i h_i'(x_*)[h_i'(x_*)]^T \\
& = & \nabla_x^2 L_0(x_*, \rho^*) + \rho H^T H, \\
H & = & \begin{bmatrix} [h_1'(x_*)]^T \\ \cdots \\ [h_k'(x_*)]^T \end{bmatrix}
\end{array}$$

$$\nabla_x^2 L_\rho(x_* \mu^*) = \nabla_x^2 L_0(x_*, \rho^*) + \rho H^T H$$
$$H = \begin{bmatrix} [h_1'(x_*)]^T \\ \cdots \\ [h_k'(x_*)]^T \end{bmatrix}$$

Directions d orthogonal to $h_i'(x_*)$, $i = 1, ..., k$, are exactly the directions d such that $Hd = 0$. Since x_* is nondegenerate local solution to (P), we have

$$Hd = 0 \ \& \ d \neq 0 \Rightarrow d^T \underbrace{\nabla_x^2 L(x_*, \mu^*)}_{Q} d > 0$$

Thus,
$\diamondsuit$ For all $\rho \geq 0$, at x_* the Second Order sufficient optimality condition for (P_ρ) holds true:

$$Hd = 0 \ \& \ d \neq 0 \Rightarrow d^T \nabla_x^2 L_\rho(x_*, \mu^*) d = d^T [Q + \rho H^T H] d > 0$$

$\Rightarrow$ All we need in order to prove that x_* is a "lucky" solution for large ρ, is to apply to $Q = \nabla_x^2 L(x_*, \mu^*)$ and H the following Linear Algebra fact:
Let Q be a symmetric $n \times n$ matrix, and H be a $k \times n$ matrix. Assume that Q is positive definite on the null space of H:

$$Hd = 0 \ \& \ d \neq 0 \Rightarrow d^T Q d > 0.$$

Then for all large enough values of ρ the matrix $Q + \rho H^T H$ is positive definite.

Claim: *Let Q be a symmetric $n \times n$ matrix, and H be a $k \times n$ matrix. Assume that Q is positive definite on the null space of H:*

$$Hd = 0 \ \& \ d \neq 0 \Rightarrow d^T Q d > 0.$$

Then for all large enough values of ρ the matrix $Q + \rho H^T H$ is positive definite.

Proof. Properly selecting orthonormal coordinates in $\mathbb{R}^n$, we can assume w.l.o.g. that the null space of H is spanned by the last $n - m$ basic orths, that is,

$$H = [G, 0_{k \times (n-m)}]$$

with $k \times m$ matrix G with linearly independent columns. Representing

$$Q = \left[\begin{array}{c|c} P & S \\ \hline S^T & R \end{array}\right] \qquad [P : m \times m]$$

positive definiteness of Q on the null space of H means that $R \succ 0$. Next,

$$H^T H = \left[\begin{array}{c|c} G^T G & \\ \hline & \end{array}\right]$$

with $G^T G \succ 0$ due to the linear independence of the columns in G. Consequently,

$$Q_\rho := Q + \rho H^T H = \left[\begin{array}{c|c} P + \rho G^T G & S \\ \hline S^T & R \end{array}\right].$$

Since $R \succ 0$, positive definiteness of Q_ρ by Schur Complement Lemma is equivalent to

$$P + \rho G^T G \succeq S R^{-1} S^T,$$

and since $G^T G \succ 0$, this relation indeed takes place for all large enough ρ. $\square$

Let Q be a symmetric $n \times n$ matrix, and H be a $k \times n$ matrix. Assume that Q is positive definite on the null space of H:

$$Hd \ \& \ d \neq 0 \Rightarrow d^T Q d > 0.$$

Then for all large enough values of ρ the matrix $Q + \rho H^T H$ is positive definite.

Alternative proof: Assume, on the contrary, that there exists a sequence $\rho_i \to \infty$ and d_i, $\|d_i\|_2 = 1$:

$$d_i^T [Q + \rho_i H^T H] d_i \leq 0 \ \forall i.$$

Passing to a subsequence, we may assume that $d_i \to d$, $i \to \infty$. Let $d_i = h_i + h_i^\perp$ be the decomposition of d_i into the sum of its projections onto Null(H) and [Null$(H)]^\perp$, and similarly $d = h + h^\perp$. Then

$$d_i^T H^T H d_i = \|Hd_i\|_2^2 = \|Hh_i^\perp\|_2^2 \to \|Hh^\perp\|_2^2 \Rightarrow$$
$$0 \geq d_i^T [Q + \rho_i H^T H] d_i = \underbrace{d_i^T Q d_i}_{\to d^T Q d} + \rho_i \underbrace{\|Hh_i^\perp\|_2^2}_{\to \|Hh^\perp\|_2^2} \ (*)$$

If $h^\perp \neq 0$, then $\|Hh^\perp\|_2 > 0$, and the right hand side in $(*)$ tends to $+\infty$ as $i \to \infty$, which is impossible. Thus, $h^\perp = 0$. But then $0 \neq d \in$ Null(H) and therefore $d^T Q d > 0$, so that the right hand side in $(*)$ is positive for large i, which again is impossible.

Putting things together:
Augmented Lagrangian Scheme

$$\min_x \left\{ f(x) + \tfrac{\rho}{2}\|h(x)\|_2^2 : h_i(x) = 0, i \le k \right\} \qquad (P_\rho)$$
$$\Rightarrow L_\rho(x,\mu) = f(x) + \tfrac{\rho}{2}\|h(x)\|_2^2 + \sum_i \mu_i h_i(x)$$

♣ **Generic Augmented Lagrangian Scheme:** *For a given value of ρ, solve the dual problem*

$$\begin{array}{c} \max_\mu \underline{\mathbf{L}}_\rho(\mu) \\ \left[\underline{\mathbf{L}}_\rho(\mu) = \min_x L_\rho(x,\mu) \right] \end{array} \qquad (D)$$

by a first order method for unconstrained minimization, getting the first order information for (D) from solving the auxiliary problems

$$x_\rho(\mu) = \operatorname{argmin}_x L_\rho(x,\mu) \qquad (P^\mu)$$

via the relations

$$\underline{\mathbf{L}}_\rho(\mu) = L_\rho(x_\rho(\mu),\mu), \quad \underline{\mathbf{L}}'_\rho(\mu) = h(x_\rho(\mu))$$

$$\min_x \left\{ f(x) + \tfrac{\rho}{2}\|h(x)\|_2^2 : \begin{array}{c} h_i(x) = 0 \\ i \le k \end{array} \right\} \qquad (P_\rho)$$

$$\Rightarrow L_\rho(x,\mu) = f(x) + \tfrac{\rho}{2}\|h(x)\|_2^2 + \sum_i \mu_i h_i(x)$$

$$\Rightarrow \max_\mu \left\{ \underline{\mathbf{L}}_\rho(\mu) \equiv \underbrace{\min_x L_\rho(x,\mu)}_{\text{problem } (P^\mu)} \right\} \qquad (D)$$

Note: *If ρ is large enough and the optimizations in (P^μ) and in (D) and are restricted to appropriate convex neighbourhoods of nondegenerate locally optimal solution x_* to (P_ρ) and the corresponding vector μ^* of Lagrange multipliers, respectively, then*
— the objective in (D) is concave and C^2, and μ^* is a nondegenerate solution to (D)
— the objectives in (P^μ) are convex and C^2, and $x_*(\mu) = \operatorname{argmin}_x L_\rho(x,\mu)$ are nondegenerate locally optimal solutions to (P^μ)
— as the "master method" working on (D) converges to μ^*, the corresponding primal iterates $x_*(\mu)$ converge to x_*.

♣ **Implementation issues:**

♢ **Solving auxiliary problems**

$$x_\rho(\mu) = \operatorname*{argmin}_x L_\rho(x, \mu) \qquad (P^\mu)$$

— the best choices are Newton method with linesearch or Modified Newton method, provided that the second order information is available; otherwise, one can use Quasi-Newton methods, Conjugate Gradients, etc.

♢ Solving the master problem

$$\max_{\mu} \left\{ \underline{\mathbf{L}}_\rho(\mu) \equiv \min_x L_\rho(x,\mu) \right\} \qquad (D)$$

Surprisingly, the method of choice here is the simplest gradient ascent method with constant step:

$$\mu^t = \mu^{t-1} + \rho \underline{\mathbf{L}}'_\rho(\mu^{t-1}) = \mu^{t-1} + \rho h(x^{t-1}),$$

where x^{t-1} is (approximate) minimizer of $L_\rho(x, \mu^{t-1})$ in x.

Motivation: We have

$$\begin{array}{rcl} 0 & \approx & \nabla_x L_\rho(x^{t-1}, \mu^{t-1}) \\ & = & f'(x^{t-1}) + \sum_i [\mu_i^{t-1} + \rho h_i(x^{t-1})] h'_i(x^{t-1}) \end{array}$$

which resembles the KKT condition

$$0 = f'(x_*) + \sum_i \mu_i^* h'_i(x_*).$$

$$\max_{\mu}\left\{\underline{\mathbf{L}}_\rho(\mu) \equiv \min_x L_\rho(x,\mu)\right\} \qquad (D)$$

$$\Rightarrow \left\{\ \mu^t = \mu^{t-1} + \rho h(x^{t-1}),\ x^{t-1} = \mathrm{argmin}_x\, L_\rho(x,\mu^{t-1})\ \right\} \qquad (*)$$

<u>Justification:</u> Direct computation shows that

$$\Psi_\rho \equiv \nabla^2_\mu \underline{\mathbf{L}}_\rho(\mu^*) = -H[Q+\rho H^T H]^{-1}H^T,$$
$$\left[Q = \nabla^2_x L_0(x_*,\mu^*),\ H = \left[\begin{array}{c}[h_1'(x_*)]^T\\ \cdots \\ [h_k'(x_*)]^T\end{array}\right]\right]$$

whence $-\rho\Psi_\rho \to I$ as $\rho\to\infty$.

Consequently, *when ρ is large enough and the starting point μ_0 in $(*)$ is close enough to μ^*, $(*)$ ensures linear convergence of μ^t to μ^* with the ratio tending to 0 as $\rho\to+\infty$.* Indeed, asymptotically the behaviour of $(*)$ is as if $\underline{\mathbf{L}}_\rho(\mu)$ were the quadratic function $\Phi(\mu) = \mathrm{const} + \frac{1}{2}(\mu-\mu^*)^T\Psi_\rho(\mu-\mu^*)$, and we were maximizing this function by the gradient ascent $\mu\mapsto\mu+\rho\Phi'(\mu)$. This recurrence is $\mu^t-\mu^* = \underbrace{(I+\rho\Psi_\rho)}_{\to 0,\rho\to\infty}(\mu^{t-1}-\mu^*)$.

♣ **Adjusting penalty parameter:**

$$\left\{ \begin{array}{rcl} \mu^t & = & \mu^{t-1} + \rho h(x^{t-1}) \\ x^{t-1} & = & \text{argmin}_x \, L_\rho(x, \mu^{t-1}) \end{array} \right. \qquad (*)$$

When ρ is "large enough", so that $(*)$ converges linearly with reasonable convergence ratio, $\|\underline{\mathbf{L}}'_\rho(\mu^t)\|_2 = \|h(x^t)\|_2$ should go to 0 linearly with essentially the same convergence ratio.

$\Rightarrow$ We can use progress in $\|h(\cdot)\|_2$ to control ρ, e.g., as follows: when

$$\|h(x^t)\|_2 \leq 0.25 \|h(x^{t-1})\|_2,$$

we keep the current value of ρ intact, otherwise we increase penalty by factor 10 and recompute x^t with the new value of ρ.

Incorporating Inequality Constraints

♣ Given a general-type constrained problem

$$\min_x \left\{ f(x) : \begin{array}{l} h_i = 0, i \le m \\ g_j(x) \le 0, j \le m \end{array} \right\}$$

we can transform it equivalently into the equality constrained problem

$$\min_{x,s} \left\{ f(x) : \begin{array}{l} h_i(x) = 0, i \le m \\ g_j(x) + s_j^2 = 0, j \le k \end{array} \right\}$$

and apply the Augmented Lagrangian scheme to the reformulated problem, thus arriving at Augmented Lagrangian

$$\begin{aligned} L_\rho(x, s; \mu, \nu) &= f(x) + \tfrac{\rho}{2} \left[\sum_i h_i^2(x) + \sum_j [g_j(x) + s_j^2]^2 \right] \\ &\quad + \sum_i \mu_i h_i(x) + \sum_j \nu_j [g_j(x) + s_j^2] \end{aligned}$$

The corresponding dual problem is

$$\max_{\mu,\nu} \left\{ \underline{\mathbf{L}}_\rho(\mu, \nu) := \min_{x,s} L_\rho(x, s; \mu, \nu) \right\} \qquad (D)$$

$$L_\rho(x,s;\mu,\nu) \;=\; f(x) + \tfrac{\rho}{2}\left[\sum_i h_i^2(x) + \sum_j [g_j(x)+s_j^2]^2\right] + \sum_i \mu_i h_i(x) + \sum_j \nu_j [g_j(x)+s_j^2]$$

$$\Rightarrow \max_{\mu,\nu}\left\{\underline{\mathbf{L}}_\rho(\mu,\nu) := \min_{x,s} L_\rho(x,s;\mu,\nu)\right\}$$

We can carry out the minimization in s analytically, arriving at

$$\underline{\mathbf{L}}_\rho(\mu,\nu) \;=\; \min_x \left\{ f(x) + \tfrac{\rho}{2}\left[\sum_{i=1}^k h_i^2 + \sum_{j=1}^m \left(g_j(x) + \tfrac{\nu_j}{\rho}\right)_+^2\right] + \sum_{i=1}^k \mu_i h_i(x)\right\} - \sum_{j=1}^m \frac{\nu_j^2}{2\rho}$$

where $a_+ = \max[0,a]$.

$\Rightarrow$ *The auxiliary problems arising in the Augmented Lagrangian Scheme are problems in the initial design variables!*

$$\min_x \left\{ f(x) : \begin{array}{c} h_i(x) = 0,\ i \le k \\ g_j(x) \le 0,\ j \le m \end{array} \right\} \qquad (P)$$

$$\Rightarrow \min_{x,s} \left\{ f(x) : \begin{array}{c} h_i(x) = 0,\ i \le k \\ g_j(x) + s_j^2 = 0,\ j \le m \end{array} \right\} \qquad (P')$$

♣ Theoretical analysis of Augmented Lagrangian scheme *for problems with equality constraints* was based on assumption that we are trying to approximate *nondegenerate* locally optimal solution. Is it true that when reducing the inequality constrained problem to an equality constrained one, we preserve nondegeneracy of the locally optimal solution?

Yes!

Theorem: *Let x_* be a nondegenerate locally optimal solution to (P). Then the point*

$$(x_*, s^*) : s_j^* = \sqrt{-g_j(x_*)},\ j = 1, ..., m$$

is a **nondegenerate** *locally optimal solution to (P').*

Convex case: Augmented Lagrangians

♣ Consider a **convex** optimization problem

$$\min_x \{f(x) : g_j(x) \leq 0,\ j = 1, ..., m\} \qquad (P)$$

(f, g_j are convex and C^2 on $\mathbb{R}^n$).
Assumption: (P) is solvable and satisfies the Slater condition:

$$\exists \bar{x} : g_j(\bar{x}) < 0 \ j = 1, ..., m$$

♠ In the convex situation, the previous local considerations can be globalized due to the Lagrange Duality Theorem.

$$\min_x \{f(x) : g_j(x) \leq 0,\ j = 1, ..., m\} \qquad (P)$$

Theorem: *Let (P) be convex, solvable and satisfy the Slater condition. Then the dual problem*

$$\max_{\lambda \geq 0} \left\{ \underline{\mathbf{L}}(\lambda) := \min_x \underbrace{\left[f(x) + \sum_j \lambda_j g_j(x) \right]}_{L(x,\lambda)} \right\} \qquad (D)$$

possess the following properties:

♢ *dual objective $\underline{\mathbf{L}}$ is concave*

♢ *(D) is solvable*

♢ *for every optimal solution λ^* of (D), all optimal solutions of (P) are contained in the set* $\mathrm{Argmin}_x L(x, \lambda^*)$.

♣ **Implications:**

♢ Sometimes we can build (D) explicitly (e.g., in Linear, Linearly Constrained Quadratic and Geometric Programming). In these cases, we may gain a lot by solving (D) and then recovering solutions to (P) from solution to (D).

$$\min_x \{f(x) : g_j(x) \le 0,\ j = 1, ..., m\} \qquad (P)$$

$$\Downarrow$$

$$\max_{\lambda \ge 0} \underline{\mathbf{L}}(\lambda) \equiv \min_x \underbrace{\left[f(x) + \sum_j \lambda_j g_j(x) \right]}_{L(x,\lambda)} \qquad (D)$$

♢ In the general case one can solve (D) numerically by an appropriate first order method. To this end we should be able to compute the first order information for $\underline{\mathbf{L}}$. This can be done via solving the auxiliary problems

$$x_* = x_*(\lambda) = \min_x L(x, \lambda) \qquad (P_\lambda)$$

due to

$$\underline{\mathbf{L}}(\lambda) = L(x_*(\lambda), \lambda),\ \underline{\mathbf{L}}'(\lambda) = g(x_*(\lambda))$$

<u>Note:</u> (P_λ) is a convex unconstrained program with smooth objective!

♣ *In all cases, passing from (P) to (D) reduces a convex problem with general convex constraints to one with simple linear constraints.*

$$\min_x \{f(x) : g_j(x) \le 0,\ j = 1, ..., m\} \qquad (P)$$

$$\Rightarrow \max_{\lambda \ge 0} \underline{\mathbf{L}}(\lambda) \equiv \min_x \underbrace{\left[f(x) + \sum_j \lambda_j g_j(x) \right]}_{L(x,\lambda)} \qquad (D)$$

♠ **<u>Potential difficulties:</u>**

♢ $\underline{\mathbf{L}}(\cdot)$ can be $-\infty$ at some points; how to solve (D)?

♢ After λ^* is found, how to recover optimal solution to (P)? We know that the set X_* of optimal solutions to (P) is contained in the set $\mathrm{Argmin}_x L(x, \lambda^*)$, but it may happen that the latter set is much larger than the former!

<u>Example: LP.</u> (P) : $\min_x \left\{c^T x : Ax - b \leq 0\right\}$. Here

$$\begin{aligned} \underline{\mathbf{L}}(\lambda) &= \min_x \left[c^T x + (A^T\lambda)^T x - b^T\lambda\right] \\ &= \begin{cases} -b^T\lambda, & A^T\lambda + c = 0 \\ -\infty, & \text{otherwise} \end{cases} \end{aligned}$$

— how to solve (D) ???
At the same time, for every λ the function $L(x,\lambda)$ is linear in x; thus, $\operatorname{Argmin}_x L(x,\lambda)$ is either $\emptyset$, or $\mathbb{R}^n$ – how to recover x_* given λ^* ???

♠ **<u>Observation:</u>** *Both outlined difficulties come from possible non-existence/non-uniqueness of solutions to the auxiliary problems*

$$\min_x L(x,\lambda) \equiv \min_x [f(x) + \sum_j \lambda_j g_j(x)] \qquad (P_\lambda)$$

Indeed, if solution $x_*(\lambda)$ to (P_λ) exists and is unique and continuous in λ on certain set Λ, then $\underline{\mathbf{L}}(\lambda)$ is finite and continuously differentiable on Λ due to

$$\begin{array}{rcl} \underline{\mathbf{L}}(\lambda) & = & L(x_*(\lambda),\lambda) \\ \underline{\mathbf{L}}'(\lambda) & = & g(x_*(\lambda)) \end{array}$$

Besides this, if $\lambda^* \in \mathrm{Argmax}_{\lambda\geq 0}\, \underline{\mathbf{L}}(\lambda)$ belongs to Λ, then there is no problem with recovering an optimal solution to (P) from λ^*.

Example: Assume that the function

$$r(x) = f(x) + \sum_{j=1}^{m} g_j(x)$$

is locally strongly convex ($r''(x) \succ 0 \ \forall x$) and is such that

$$r(x)/\|x\|_2 \to \infty, \ \ \|x\|_2 \to \infty.$$

Then $x_*(\lambda)$ exists, is unique and is continuous in λ on the set $\Lambda = \{\lambda > 0\}$.

When f itself is locally strongly convex and $f(x)/\|x\|_2 \to \infty$ as $\|x\|_2 \to \infty$, the conclusion holds true with $\Lambda = \{\lambda \geq 0\}$.

♣ In *Augmented Lagrangian* scheme, we ensure local strong convexity of

$$r(\cdot) = f(x) + \text{sum of constraints}$$

by passing from the original problem

$$\min_x \{f(x) : g_j(x) \leq 0,\ j = 1, ..., m\} \qquad (P)$$

to the *equivalent* problem

$$\min_x \{f(x) : \theta_j(g_j(x)) \leq 0,\ j = 1, ..., m\} \qquad (P')$$

where $\theta_j(\cdot)$ are *increasing strongly convex* smooth functions satisfying the normalization

$$\theta_j(0) = 0,\ \theta_j'(0) = 1,$$

e.g.,

$$\theta_j(t) = \mathrm{e}^t - 1$$

or

$$\theta_j(t) = 2\ln(1 + \mathrm{e}^t) - 2\ln 2.$$

$$\min_x \{f(x) : g_j(x) \le 0,\ j = 1, ..., m\} \qquad (P)$$
$$\Downarrow$$
$$\min_x \{f(x) : \theta_j(g_j(x)) \le 0,\ j = 1, ..., m\} \quad (P')$$
$$\left[\theta_j(0) = 0,\ \theta_j'(0) = 1\right]$$

Facts:

$\diamondsuit$ (P') is convex and equivalent to (P)

$\diamondsuit$ optimal Lagrange multipliers for (P) and (P') are the same: if λ^* is the vector of Lagrange multipliers justifying that x_* is optimal for (P), the same λ^* justifies that x_* is optimal for (P') (due to $\theta_j'(0) = 1$):

$$\lambda^* \ge 0 \ \&\ \overbrace{\nabla_x\big|_{x=x_*}[f(x) + \sum_j \lambda_j^* g_j(x)]}^{f'(x_*) + \sum_j \lambda_j^* g'(x_*)} = 0 \ \&\ \lambda_j^* g_j(x_*) = 0\,\forall j$$
$$\Updownarrow$$
$$\lambda^* \ge 0 \ \&\ \underbrace{\nabla_x\big|_{x=x_*}[f(x) + \sum_j \lambda_j^* \theta_j(g_j(x))]}_{f'(x_*) + \sum_j \lambda_j^* \theta_j'(g_j(x_*)) g'(x_*)} = 0 \ \&\ \lambda_j^* \theta_j(g_j(x_*)) = 0\,\forall j$$

$\diamondsuit$ under mild regularity assumptions,

$$r(x) = f(x) + \sum_j \theta_j(g_j(x))$$

is locally strongly convex and $r(x)/\|x\|_2 \to \infty$ as $\|x\|_2 \to \infty$.

$$\min_x \{f(x) : g_j(x) \le 0,\ j = 1, ..., m\} \qquad (P)$$

$$\Downarrow$$

$$\min_x \{f(x) : \theta_j(g_j(x)) \le 0,\ j = 1, ..., m\} \quad (P')$$

$$\left[\theta_j(0) = 0,\ \theta_j'(0) = 1\right]$$

♣ With the outlined scheme, one passes from the classical Lagrange function of (P)

$$L(x, \lambda) = f(x) + \sum_j \lambda_j g_j(x)$$

to the *augmented Lagrange function*

$$\widetilde{L}(x, \lambda) = f(x) + \sum_j \lambda_j \theta_j(g_j(x))$$

of the problem, which yields the dual problem

$$\max_{\lambda \ge 0} \underline{\mathbf{L}}(\lambda) \equiv \max_{\lambda \ge 0} \min_x \widetilde{L}(x, \lambda)$$

better suited for numerical solution and recovering a solution to (P) than the usual Lagrange dual of (P).

$$L(x,\lambda) = f(x) + \sum_j \lambda_j g_j(x)$$
$$\Rightarrow \widetilde{L}(x,\lambda) = f(x) + \sum_j \lambda_j \theta_j(g_j(x))$$
$$\Rightarrow \max_{\lambda \geq 0} \left[\min_x \widetilde{L}(x,\lambda) \right] \qquad (\widetilde{D})$$

♠ Further flexibility is added by penalty mechanism:

$$\widetilde{L}(x,\lambda) \Rightarrow f(x) + \sum_j \lambda_j \rho^{-1} \theta_j(\rho g_j(x))$$

equivalent to "rescaling"

$$\theta_j(s) \Rightarrow \theta_{j,\rho}(s) = \rho^{-1}\theta_j(\rho s) \qquad \Rightarrow [\theta'_{j,\rho}(0) = 1]$$

When $\rho > 1$, this rescaling increases "the curvature" of the rescaled constraint at a point where the constraint is active:

$$g_j(x) = 0 \Rightarrow \nabla_x^2[\theta_{j,\rho}(g_j(x))] = \rho\theta''_j(0) \cdot \nabla_x g_j(x)[\nabla_x g_j(x)]^T + \nabla_x^2 g_j(x).$$

As a result, the larger is ρ, the faster is convergence of the first order methods as applied to $(\widetilde{D})$ **<u>and</u>** the more difficult become the auxiliary problems

$$\min_x \left[f(x) + \sum_j \lambda_j \rho^{-1} \theta_j(\rho g_j(x)) \right]$$

Lecture 13:

Algorithms for Constrained Optimization, III: Sequential Quadratic Programming

Sequential Quadratic Programming

♣ SQP is thought of to be the most efficient technique for solving general-type optimization problems with smooth objective and constraints.
♣ SQP methods directly solve the KKT system of the problem by a Newton-type iterative process.

♣ Consider an equality constrained problem

$$\min_x \left\{ f(x) : h(x) = (h_1(x), ..., h_k(x))^T = 0 \right\} \quad (P)$$
$$\Rightarrow L(x, \mu) = f(x) + h^T(x)\mu$$

The KKT system of the problem is

$$\begin{array}{rcl} \nabla_x L(x,\mu) & \equiv & f'(x) + [h'(x)]^T \mu = 0 \\ \nabla_\mu L(x,\mu) & \equiv & h(x) = 0 \end{array}$$
$$\left[h'(x) = \left[\begin{array}{c} [\nabla h_1(x)]^T \\ \cdots\cdots\cdots \\ [\nabla h_k(x)]^T \end{array} \right] \right] \qquad \text{(KKT)}$$

Every locally optimal solution x_* of (P) which is regular (that is, the gradients $\{h_i'(x_*)\}_{i=1}^k$ are linearly independent) can be extended by properly chosen $\mu = \mu^*$ to a solution of (KKT).

♠ (KKT) is a system of nonlinear equations with $n + k$ equations and $n + k$ unknowns, where n is the dimension of x. We can try to solve this system by Newton method.

Newton method for solving nonlinear systems of equations

♣ To solve a system of N nonlinear equations with N unknowns

$$P(u) \equiv (p_1(u), ..., p_N(u))^T = 0,$$

with C^1 real-valued functions p_i, we act as follows:
Given current iterate $\bar{u}$, we linearize the system at the iterate, thus arriving at the linearized system

$$P(\bar{u}) + P'(\bar{u})(u - \bar{u}) \equiv \begin{bmatrix} p_1(\bar{u}) + [p_1'(\bar{u})]^T(u - \bar{u}) \\ \vdots \\ p_N(\bar{u}) + [p_N'(\bar{u})]^T(u - \bar{u}) \end{bmatrix} = 0.$$

Assuming the $N \times N$ matrix

$$P'(\bar{u}) = \begin{bmatrix} [p_1'(x)]^T \\ [p_2'(x)]^T \\ \cdots \\ [p_N'(x)]^T \end{bmatrix}$$

nonsingular, we solve the linearized system, thus getting the new iterate

$$\bar{u}^+ = \bar{u} \underbrace{-[P'(\bar{u})]^{-1}P(\bar{u})}_{\substack{\text{Newton} \\ \text{displacement}}};$$

$$\bar{u} \mapsto \bar{u}^+ = \bar{u} - [P'(\bar{u})]^{-1} P(\bar{u}) \qquad (N)$$

Note: The Basic Newton method for unconstrained minimization is nothing but the outlined process as applied to the Fermat equation

$$P(x) \equiv \nabla f(x) = 0.$$

♣ Same as in the optimization case, the Newton method possesses fast local convergence:

Theorem. *Let $u_* \in \mathbb{R}^N$ be a solution to the square system of nonlinear equations*

$$P(u) = 0$$

with components of P being C^1 in a neighbourhood of u_*. Assuming that u_* is nondegenerate (i.e., $\mathrm{Det}(P'(u_*)) \neq 0$), the Newton method (N), started close enough to u_*, is well defined and converges to u_* superlinearly.
If, in addition, the components of P are C^2 in a neighbourhood of u_*, then the above convergence is quadratic.

♣ Applying the outlined scheme to the KKT system

$$\begin{array}{rcl} \nabla_x L(x,\mu) & \equiv & f'(x) + [h'(x)]^T \mu = 0 \\ \nabla_\mu L(x,\mu) & \equiv & h(x) = 0 \end{array} \qquad \text{(KKT)}$$

we should answer first of all the following crucial question:
(?) When a KKT point (x_*, μ^*) is a *nondegenerate* solution to (KKT)?
Let us set

$$P(x,\mu) = \nabla_{x,\mu} L(x,\mu) = \begin{bmatrix} \nabla_x L(x,\mu) \equiv f'(x) + [h'(x)]^T \mu \\ \nabla_\mu L(x,\mu) \equiv h(x) \end{bmatrix}$$

Note that

$$P'(x,\mu) = \begin{bmatrix} \nabla_x^2 L(x,\mu) & [h'(x)]^T \\ h'(x) & 0 \end{bmatrix}$$

$$\min_x \left\{ f(x) : h(x) = (h_1(x), ..., h_k(x))^T = 0 \right\} \qquad (P)$$

$$\Rightarrow \qquad L(x,\mu) = f(x) + h^T(x)\mu$$

$$\Rightarrow \quad P(x,\mu) = \nabla_{x,\mu} L(x,\mu) = \begin{bmatrix} \nabla_x L(x,\mu) \equiv f'(x) + [h'(x)]^T \mu \\ \nabla_\mu L(x,\mu) \equiv h(x) \end{bmatrix}$$

$$\Rightarrow \qquad P'(x,\mu) = \begin{bmatrix} \nabla_x^2 L(x,\mu) & [h'(x)]^T \\ h'(x) & 0 \end{bmatrix}$$

Theorem. *Let x_* be a nondegenerate locally optimal solution to (P) and μ^* be the corresponding vector of Lagrange multipliers. Then (x_*, μ^*) is a nondegenerate solution to the KKT system*

$$P(x,\mu) = 0,$$

that is, the matrix $P' \equiv P'(x_*, \mu^*)$ is nonsingular.

$$\min_x \left\{ f(x) : h(x) = (h_1(x), ..., h_k(x))^T = 0 \right\} \qquad (P)$$

$$\Rightarrow \qquad L(x,\mu) = f(x) + h^T(x)\mu$$

$$\Rightarrow \quad P(x,\mu) = \nabla_{x,\mu} L(x,\mu) = \begin{bmatrix} \nabla_x L(x,\mu) \equiv f'(x) + [h'(x)]^T \mu \\ \nabla_\mu L(x,\mu) \equiv h(x) \end{bmatrix}$$

$$\left[h'(x) = \begin{bmatrix} [\nabla h_1(x)]^T \\ ..... \\ [\nabla h_k(x)]^T \end{bmatrix} \right]$$

$$\Rightarrow \qquad P'(x,\mu) = \begin{bmatrix} \nabla_x^2 L(x,\mu) & [h'(x)]^T \\ h'(x) & 0 \end{bmatrix}$$

Claim: *Let x_* be a nondegenerate locally optimal solution to (P) and μ^* be the corresponding vector of Lagrange multipliers. Then (x_*, μ^*) is a nondegenerate solution to the KKT system $P(x,\mu) = 0$, that is, the matrix $P' \equiv P'(x_*, \mu^*)$ is nonsingular.*

Proof. Setting $Q = \nabla_x^2 L(x_*, \mu^*)$, $H = h'(x_*)$, we have

$$P' = \begin{bmatrix} Q & H^T \\ H & 0 \end{bmatrix}.$$

We know that $d \neq 0, Hd = 0 \Rightarrow d^T Q d > 0$ and that rows of H are linearly independent.
We should prove that if

$$0 = P' \begin{bmatrix} d \\ g \end{bmatrix} \equiv \begin{bmatrix} Qd + H^T g \\ Hd \end{bmatrix},$$

then $d = 0$, $g = 0$.

Given that the rows of H are linearly independent and $d^TQd > 0$ whenever $d \neq 0$ and $Hd = 0$, we should prove that

$$\underbrace{0 = P' \begin{bmatrix} d \\ g \end{bmatrix} \equiv \begin{bmatrix} Qd + H^Tg \\ Hd \end{bmatrix}}_{\Leftrightarrow Qd+H^Tg=0, Hd=0} \Rightarrow d = 0, g = 0$$

We have $Hd = 0$ and

$$0 = Qd + H^Tg \Rightarrow 0 = d^T[Qd + H^Tg] = d^TQd + (\underbrace{Hd}_{0})^Tg = d^TQd,$$

which, as we know, for d satisfying $Hd = 0$ is possible iff $d = 0$.
We now have $H^Tg = Qd + H^Tg = 0$; since the rows of H are linearly independent, it follows that $g = 0$. $\square$

Structure and interpretation of the Newton displacement

♣ In our case the Newton system

$$P'(u)\Delta = -P(u) \qquad [\Delta = u^+ - u]$$

becomes

$$\begin{array}{rcl} [\nabla_x^2 L(\bar{x},\bar{\mu})]\Delta x \;+\; [h'(\bar{x})]^T\Delta\mu & = & -f'(\bar{x}) - [h'(\bar{x})]^T\bar{\mu} \\ [h'(\bar{x})]\Delta x & = & -h(\bar{x}) \end{array},$$

where $(\bar{x},\bar{\mu})$ is the current iterate.
Passing to the variables Δx, $\mu^+ = \bar{\mu} + \Delta\mu$, the system becomes

$$\begin{array}{rcl} [\nabla_x^2 L(\bar{x},\bar{\mu})]\Delta x \;+\; [h'(\bar{x})]^T\mu^+ & = & -f'(\bar{x}) \\ h'(\bar{x})\Delta x & = & -h(\bar{x}) \end{array}$$

$$\boxed{\begin{array}{rcl}[\nabla_x^2 L(\bar{x},\bar{\mu})]\Delta x + [h'(\bar{x})]^T\mu^+ & = & -f'(\bar{x}) \\ h'(\bar{x})\Delta x & = & -h(\bar{x})\end{array}}$$
$$(\bar{x},\bar{\mu}) \mapsto (\bar{x}+\Delta x, \mu^+)$$

Interpretation:

♣ Let x_* be a nondegenerate locally optimal solution to

$$\min_x \left\{ f(x) : h(x) = (h_1(x), ..., h_k(x))^T = 0 \right\} \qquad (P)$$

Assume for a moment that *we know the optimal Lagrange multipliers* μ^* **and** *the tangent plane* T *to the feasible surface at* x_**:*

$$T = \{y = x_* + \Delta x : h'(x_*)\Delta x + h(x_*) = 0\}.$$

Since $\nabla_x^2 L(x_*, \mu^*)$ is positive definite on $T - x_*$ and $\nabla_x L(x_*, \mu^*) = 0$, x_* is a nondegenerate local minimizer of $L(x, \mu^*)$ *over* $x \in T$, and we could find x_* by applying the Newton minimization method to the function $L(x, \mu^*)$ *restricted onto* T, the iterations being

$$\bar{x} \mapsto \bar{x} + \mathrm{argmin}_{\Delta x : \bar{x}+\Delta x \in T} \left[L(\bar{x}, \mu^*) + \Delta x^T \nabla_x L(\bar{x}, \mu^*) + \tfrac{1}{2}\Delta x^T \nabla_x^2 L(\bar{x}, \mu^*)\Delta x \right]$$

♣ In reality we do *not* know μ^* and T, we know only current approximations $\bar{x}$, $\bar{\mu}$ of x_* and μ^*. We can use these approximations to *approximate* the outlined scheme:

- Given $\bar{x}$, we approximate T by the plane

$$\overline{T} = \{y = \bar{x} + \Delta x : h'(\bar{x})\Delta x + h(\bar{x}) = 0\}$$

- We apply the outlined step with μ^*, T replaced with $\bar{\mu}$ and $\overline{T}$:

$$\bar{x} \mapsto \bar{x} + \text{argmin}_{\Delta x : \bar{x} + \Delta x \in \overline{T}} \left[L(\bar{x}, \bar{\mu}) + \Delta x^T \nabla_x L(\bar{x}, \bar{\mu}) + \tfrac{1}{2} \Delta x^T \nabla_x^2 L(\bar{x}, \bar{\mu}) \Delta x \right] \quad (A)$$

Note: Step can be simplified to

$$\bar{x} \mapsto \bar{x} + \text{argmin}_{\Delta x : \bar{x} + \Delta x \in \overline{T}} \left[f(\bar{x}) + \Delta x^T f'(\bar{x}) + \tfrac{1}{2} \Delta x^T \nabla_x^2 L(\bar{x}, \bar{\mu}) \Delta x \right] \quad (B)$$

due to the fact that for $\bar{x} + \Delta x \in \overline{T}$ one has

$$\begin{array}{rcl} \Delta x^T \nabla_x L(\bar{x}, \bar{\mu}) & = & \Delta x^T f'(\bar{x}) + \Delta x^T [h'(\bar{x})]^T \bar{\mu} \\ & = & \Delta x^T f'(\bar{x}) + \bar{\mu}^T h'(\bar{x}) \Delta x \\ & = & \Delta x^T f'(\bar{x}) - \bar{\mu}^T h(\bar{x}) \end{array}$$

$\Rightarrow$ *When $\bar{x} + \Delta x \in \overline{T}$, the functions of Δx we are minimizing in (A) and in (B) differ by a constant.*

♣ We have arrived at the following scheme:
Given approximation $(\bar{x}, \bar{\mu})$ to a nondegenerate KKT point (x_*, μ^*) of equality constrained problem

$$\min_x \left\{ f(x) : h(x) \equiv (h_1(x), ..., h_k(x))^T = 0 \right\} \tag{P}$$

solve the auxiliary quadratic program

$$\Delta x_* = \operatorname*{argmin}_{\Delta x} \left\{ f(\bar{x}) + \Delta x^T f'(\bar{x}) + \tfrac{1}{2} \Delta x^T \nabla_x^2 L(\bar{x}, \bar{\mu}) \Delta x : \right. \\ \left. h(\bar{x}) + h'(\bar{x}) \Delta x = 0 \right\} \tag{QP}$$

and replace $\bar{x}$ with $\bar{x} + \Delta x_*$.
<u>Note:</u> (QP) is a nice Linear Algebra problem, provided that $\nabla_x^2 L(\bar{x}, \bar{\mu})$ is positive definite on the linear subspace $\{\Delta x : h'(\bar{x}) \Delta x = 0\}$ parallel to the feasible plane of (QP) (which indeed is the case when $(\bar{x}, \bar{\mu})$ is close enough to (x_*, μ^*)).

$$\min_x \left\{ f(x) : h(x) \equiv (h_1(x), ..., h_k(x))^T = 0 \right\} \qquad (P)$$

♣ **Step of the Newton method** as applied to the KKT system of (P):

$$(\bar{x}, \bar{\mu}) \mapsto (\bar{x}^+ = \bar{x} + \Delta x, \mu^+) :$$
$$\left[\begin{array}{rcl} [\nabla_x^2 L(\bar{x}, \bar{\mu})]\Delta x \ + \ [h'(\bar{x})]^T \mu^+ & = & -f'(\bar{x}) \\ h'(\bar{x})\Delta x & = & -h(\bar{x}) \end{array} \right] \qquad (N)$$

♣ **Associated quadratic program:**

$$\min_{\Delta x} \left\{ f(\bar{x}) + \Delta x^T f'(\bar{x}) + \tfrac{1}{2} \Delta x^T \nabla_x^2 L(\bar{x}, \bar{\mu}) \Delta x : h(\bar{x}) + h'(\bar{x}) \Delta x = 0 \right\} \qquad \text{(QP)}$$

Crucial observation: *Let the Newton system underlying (N) be a system with non-singular matrix. Then the Newton displacement Δx given by (N) is the unique KKT point of the quadratic program (QP), and μ^+ is the corresponding vector of Lagrange multipliers.*

$$\begin{array}{rcl} [\nabla_x^2 L(\bar{x},\bar{\mu})]\Delta x \;+\; [h'(\bar{x})]^T\mu^+ & = & -f'(\bar{x}) \\ h'(\bar{x})\Delta x & = & -h(\bar{x}) \end{array} \qquad (N)$$

$$\min_{\Delta x}\left\{ f(\bar{x}) + \Delta x^T f'(\bar{x}) + \tfrac{1}{2}\Delta x^T \nabla_x^2 L(\bar{x},\bar{\mu})\Delta x : h'(\bar{x})\Delta x = -h(\bar{x}) \right\} \qquad \text{(QP)}$$

Proof of Crucial Observation: Let z be a KKT point of (QP), and μ be the corresponding vector of Lagrange multipliers. The KKT system for (QP) reads

$$\begin{array}{rcl} f'(\bar{x}) + \nabla_x^2 L(\bar{x},\bar{\mu})z \;+\; [h'(\bar{x})]^T\mu & = & 0 \\ h'(\bar{x})z & = & -h(\bar{x}) \end{array}$$

which are exactly the equations in (N). Since the matrix of system (N) is nonsingular, we have $z = \Delta x$ and $\mu = \mu^+$.

$$\min_x \left\{ f(x) : h(x) \equiv (h_1(x), ..., h_k(x))^T = 0 \right\} \qquad (P)$$

♣ The Newton method as applied to the KKT system of (P) works as follows:
Given current iterate $(\bar{x}, \bar{\mu})$, we linearize the constraints, thus getting "approximate tangent plane to the feasible set"

$$\overline{T} = \{\bar{x} + \Delta x : h'(\bar{x})\Delta x = -h(\bar{x})\},$$

and minimize over this set the quadratic function

$$f(\bar{x}) + (x - \bar{x})^T f'(\bar{x}) + \frac{1}{2}(x - \bar{x})^T \nabla_x^2 L(\bar{x}, \bar{\mu})(x - \bar{x}).$$

The solution of the resulting quadratic problem with linear equality constraints is the new x-iterate, and the vector of Lagrange multipliers associated with this solution is the new μ-iterate.
Note: The quadratic part in the auxiliary quadratic objective comes from the Lagrange function of (P), and not from the objective of (P)!

General constrained case

♣ "Optimization-based" interpretation of the Newton method as applied to the KKT system of equality constrained problem can be extended onto the case of general constrained problem

$$\min_x \left\{ f(x) : \begin{array}{l} h(x) = (h_1(x), ..., h_k(x))^T = 0 \\ g(x) = (g_1(x), ..., g_m(x))^T \leq 0 \end{array} \right\} \qquad (P)$$

and results in the **Basic SQP scheme**:

Given current approximations x_t, μ_t, $\lambda_t \geq 0$ to a nondegenerate locally optimal solution x_* of (P) and corresponding optimal Lagrange multipliers μ^*, λ^*, we solve auxiliary linearly constrained quadratic problem

$$\Delta x_* = \operatorname*{argmin}_{\Delta x} \left\{ f(x_t) + \Delta x^T f'(x_t) + \tfrac{1}{2} \Delta x^T \nabla_x^2 L(x_t; \mu_t, \lambda_t) \Delta x : \begin{array}{l} h'(x_t)\Delta x = -h(x_t) \\ g'(x_t)\Delta x \leq -g(x_t) \end{array} \right\} \qquad (QP_t)$$

$$L(x; \mu, \lambda) = f(x) + h^T(x)\mu + g^T(x)\lambda$$

set $x_{t+1} = x_t + \Delta x_*$ and define μ_{t+1}, λ_{t+1} as the optimal Lagrange multipliers of (QP_t).

Theorem. *Let $(x_*; \mu^*, \lambda^*)$ be a nondegenerate locally optimal solution to (P) and the corresponding optimal Lagrange multipliers. The Basic SQP method, started close enough to $(x_*; \mu^*, \lambda^*)$, and restricted to work with appropriately small Δx, is well defined and converges to $(x_*; \mu^*, \lambda^*)$ quadratically.*

♣ **Difficulty:** From the "global" viewpoint, the auxiliary quadratic problem to be solved may be bad (e.g., infeasible or below unbounded). In the *equality constrained* case, this never happens when we are close to the nondegenerate locally optimal solution; in the general case, bad things may happen even close to a nondegenerate locally optimal solution.

$$\min_x \left\{ f(x) : \begin{array}{l} h(x) = (h_1(x), ..., h_k(x))^T = 0 \\ g(x) = (g_1(x), ..., g_m(x))^T \leq 0 \end{array} \right\} \qquad (P)$$

$$\min_{\Delta x} \left\{ f(x_t) + \Delta x^T f'(x_t) + \tfrac{1}{2} \Delta x^T \nabla_x^2 L(x_t; \mu_t, \lambda_t) \Delta x : \begin{array}{l} h'(x_t)\Delta x = -h(x_t) \\ g'(x_t)\Delta x \leq -g(x_t) \end{array} \right\}$$

♣ **Cure:** replace the matrix $\nabla_x^2 L(x_t; \mu_t, \lambda_t)$ when it is not positive definite on the entire space by a positive definite matrix B_t, thus arriving at the method where the auxiliary quadratic problem is

$$\min_{\Delta x} \left\{ f(x_t) + \Delta x^T f'(x_t) + \tfrac{1}{2} \Delta x^T B_t \Delta x : \begin{array}{l} h'(x_t)\Delta x = -h(x_t) \\ g'(x_t)\Delta x \leq -g(x_t) \end{array} \right\} \qquad (\mathrm{QP}_t)$$

With this modification, the auxiliary problems are convex and solvable with unique optimal solution (*provided they are feasible*, which indeed is the case when x_t is close to a nondegenerate solution to (P)).

Ensuring global convergence

♣ "Cured" Basic SQP scheme possesses nice local convergence properties; however, it in general is not globally converging.
Indeed, in the simplest unconstrained case SQP becomes the basic/modified Newton method, which is not necessarily globally converging, unless linesearch is incorporated.

♠ To ensure global convergence of SQP, we incorporate linesearch. In the scheme with linesearch, the optimal solution Δx_* to the auxiliary quadratic problem

$$\Delta x_* = \underset{\Delta x}{\operatorname{argmin}} \left\{ f(x_t) + \Delta x^T f'(x_t) + \tfrac{1}{2}\Delta x^T B_t \Delta x : \begin{array}{c} h'(x_t)\Delta x = -h(x_t) \\ g'(x_t)\Delta x \leq -g(x_t) \end{array} \right\} \qquad (\mathrm{QP}_t)$$

and the associated Lagrange multipliers μ^+, λ^+ are used as *search direction* rather than as a new iterate. The new iterate is

$$\begin{array}{rcl} x_{t+1} & = & x_t + \gamma_{t+1}\Delta x_* \\ \mu_{t+1} & = & \mu_t + \gamma_{t+1}(\mu^+ - \mu_t) \\ \lambda_{t+1} & = & \lambda_t + \gamma_{t+1}(\lambda^+ - \lambda_t) \end{array}$$

where $\gamma_{t+1} > 0$ is the stepsize given by linesearch.
Note: In (QP_t), we do not see μ_t and λ_t. They, however, could present in this problem *implicitly* – as the data utilized when building B_t.
Question: What should be minimized by the linesearch?

♣ In the constrained case, the auxiliary objective to be minimized by the linesearch cannot be chosen as the objective of the problem of interest. In the case of SQP, a good auxiliary objective ("merit function") is

$$M(x) = f(x) + \theta \left[\sum_{i=1}^{k} |h_i(x)| + \sum_{j=1}^{m} g_j^+(x) \right]$$
$$\left[g_j^+(x) = \max[0, g_j(x)] \right]$$

where $\theta > 0$ is parameter.

<u>Fact:</u> *Let x_t be current iterate, B_t be a positive definite matrix used in the auxiliary quadratic problem, Δx be a solution to this problem and $\mu \equiv \mu_{t+1}$, $\lambda \equiv \lambda_{t+1}$ be the corresponding Lagrange multipliers. Assume that θ is large enough:*

$$\theta \geq \max\{|\mu_1|, ..., |\mu_k|, \lambda_1, \lambda_2, ..., \lambda_m\}$$

Then either $\Delta x = 0$, and then x_t is a KKT point of the original problem, or $\Delta x \neq 0$, and then Δx is a direction of decrease of $M(\cdot)$, that is,

$$M(x + \gamma \Delta x) < M(x)$$

for all small enough $\gamma > 0$.

SQP Algorithm with Merit Function

♣ Generic SQP algorithm with merit function is as follows:

♢ **Initialization**: Choose $\theta_1 > 0$ and starting point x_1

♢ **Step t:** Given current iterate x_t,

— choose a matrix $B_t \succ 0$ and form and solve auxiliary problem

$$\min_{\Delta x} \left\{ f(x_t) + \Delta x^T f'(x_t) + \tfrac{1}{2}\Delta x^T B_t \Delta x : \begin{array}{l} h'(x_t)\Delta x = -h(x_t) \\ g'(x_t)\Delta x \leq -g(x_t) \end{array} \right\} \qquad (\mathrm{QP}_t)$$

thus getting the optimal Δx along with associated Lagrange multipliers μ, λ.

— if $\Delta x = 0$, terminate: x_t is a KKT point of the original problem, otherwise proceed as follows:

— check whether

$$\theta_t \geq \bar{\theta}_t \equiv \max\{|\mu_1|, ..., |\mu_k|, \lambda_1, ..., \lambda_m\}.$$

if it is the case, set $\theta_{t+1} = \theta_t$, otherwise set

$$\theta_{t+1} = \max[\bar{\theta}_t, 2\theta_t];$$

— Find the new iterate

$$x_{t+1} = x_t + \gamma_{t+1}\Delta x$$

by linesearch aimed to minimize the merit function

$$M_{t+1}(x) = f(x) + \theta_{t+1}\left[\sum_{i=1}^{k} |h_i(x)| + \sum_{j=1}^{m} g_j^+(x)\right]$$

on the search ray $\{x_t + \gamma\Delta x \mid \gamma \geq 0\}$. Replace t with $t+1$ and loop.

$$\min_x \left\{ f(x) : \begin{array}{l} h(x) = (h_1(x), ..., h_k(x))^T = 0 \\ g(x) = (g_1(x), ..., g_m(x))^T \leq 0 \end{array} \right\} \qquad (P)$$

Theorem: *Let general constrained problem be solved by SQP algorithm with merit function. Assume that*

- *there exists a compact $\Omega \subset \mathbb{R}^n$ such that for $x \in \Omega$ the solution set $D(x)$ of the system of linear inequality constraints*

$$S(x): \quad h'(x)\Delta x = -h(x),\ g'(x)\Delta x \leq -g(x)$$

with unknowns Δx is nonempty, and each vector $\Delta x \in D(x)$ is a regular solution of system $S(x)$;

- *the trajectory $\{x_t\}$ of the algorithm belongs to Ω and is infinite (i.e., the method does not terminate with exact KKT point);*
- *the matrices B_t used in the method are uniformly bounded and uniformly positive definite: $cI \preceq B_t \preceq CI$ for all t, with some $0 < c \leq C < \infty$.*

Then all accumulation points of the trajectory of the method are KKT points of (P).

Lecture 14:
Frontiers, Challenges, Perspectives

Methods for Nonlinear Optimization: Frontiers, Challenges and Perspectives

♣ **Disclaimer:** *All opinions to follow (in contrast to facts) are personal and do not pretend to be ultimate truth !*

♣ **Apology:** Some of you are ISyE students who *are obliged* to take the 6663 course. However, *I suspect* than many of you took the course due to extreme today popularity of Optimization beyond Optimization/Operations Research Communities *per se.*

• *I suspect* that today popularity of Nonlinear Optimization stems from unprecedented interest in and successes of *Machine Learning* where Continuous Optimization is an important, to say the least, element of "computational toolbox."

• Students who took 6663 because of the role of Optimization in Machine Learning can think that they were cheated and should "request their money back:" instead of Deep Learning, Stochastic Gradient Descent, and other "hot" ML-related issues they were taught

— in the "theoretical" part – things like Caratheodory Theorem, Separation of convex sets, Optimality Conditions known, for something in-between 150 and 50 years;

— in the "algorithmic" part – algorithms of "age" in-between 15 (Newton method with cubic regularization) and 60+ (gradient descent) years.

♠ It would take too much time to explain why you were taught what you were taught. Short explanation is: *because I believe that the concepts and results you were taught, especially in the theoretical part of the course, are everlasting components of Optimization and will serve your Optimization-related needs for tens of years to come*.
The "value" of Pythagoras Theorem today is as high as it was 2300+ years ago when Theorem was discovered. Farkas Lemma, Theorem on Alternative, Separation of convex sets, KKT conditions, etc., albeit younger, are in the same category of *eternal ultimate truths*, and *I believe* truths of this type should be the primary focus of a *basic graduate* university course.

♠ It is easy to explain *why you were not taught Deep Learning, Stochastic Subgradient Descent, and other hot topics*. The reason is that *I believe* that the fantastic real life successes of today Machine Learning technologies are *brilliant engineering achievements* which do not have much to do with Math in general and Nonlinear Optimization in particular.

My beliefs are no more than my beliefs, but I am not the only one with these beliefs. I strongly recommend you YouTube lecture of an outstanding Stanford statistician Prof. David Donoho

https://www.youtube.com/watch?v=1-cAT73NRwM&feature=youtu.be

The lecture is Intro to Stanford STATS 285 course and is fantastic, definitely worthy of viewing from the very beginning to the very end; the part on Deep Learning starts at about min 45 of the video.

♠ As about Stochastic Subgradient Descent, to present its nearly complete theory to you would require something like half an hour. However, this theory does not explain *when and why* this algorithm as applied to training Deep Neural Nets produces useful results, and these "when and why" questions go far beyond my (and not only my!) understanding...

End of Apology

♣ In the last decade or so, the traditional Mathematical Programming paradigm of what is an MP program and what is a solutions algorithm was essentially extended in at least two directions:

- On-Line Optimization
- Distributed Optimization

♣ **On-Line Optimization:** In traditional MP, a solution algorithm is an *off-line* process: all we want is to learn the optimization program of interest in order to get as fast as possible a good approximate solution; this solution is what actually will be used "in real life."

Since the search points generated in the learning process are *not* used in "real life", we *pay nothing* for their "heavy infeasibility" or "heavy nonoptimality."

On-Line Optimization is about "learning in real time," where the search points *are* the subsequent "real life" decisions we make, so that their quality matters. A typical setting is as follows:

• at time t, $1 \leq t \leq T$, we select search point $x_t \in X \subset \mathbb{R}^n$, and the nature (or an adversary) selects current objective $f_t(\cdot) \in \mathcal{F}$, where X is a known in advance (usually, convex) subset of $\mathbb{R}^n$, and $\mathcal{F}$ is a known in advance family of functions (usually, convex) on X.

• At step t, our loss is $f_t(x_t)$, and this loss (or its unbiased stochastic estimate $g_t(x_t)$) and perhaps some additional information on f_t (e.g., subgradient of f_t at x_t, or unbiased stochastic estimate of this subgradient) become known. We can select $x_{t+1} \in X$ as we want, based on information accumulated so far.

• The standard goal is to find a policy of generating x_1, x_2,...,x_T which results in as small as possible *regret*

$$\frac{1}{T}\mathbf{E}\left\{\sum_{t=1}^{T} g_t(x_t)\right\} - \frac{1}{T}\min_{x\in X}\mathbf{E}\left\{\sum_{t=1}^{T} f_t(x)\right\}$$

In other words, we do pay for nonoptimality of search points x_t and want to make our average payment close to the one of "clairvoyant" who knows the future but "cannot move" – sticks to time-invariant solution.

Fact: In the convex case with (unbiased stochastic estimates of) subgradients of f_t at x_t available, online regret minimization can be handled by standard tools of Convex Optimization (Stochastic Subgradient/Mirror Descent). The "bandit" setting where the only on-line available information is given by (unbiased stochastic estimates of) $f_t(x_t)$ is much more difficult and is subject of intensive research.

♣ **Distributed Optimization:** Traditional solution algorithms in MP are "black box oriented" and *sequential* — the next search point is specified in terms of local information on objective and constraints acquired at the preceding search points.
Moreover, for typical classes of MP problems possibility of "parallelization" – generating at a step M search points instead of just one and acquiring local information at all these points in parallel – does not allow to accelerate the learning process, unless M is unrealistically large (an exponent of the number of variables).
$\Rightarrow$ *as far as learning is concerned, access to several processors instead of a single one usually does not help*.
Note: Such an access can be useful when implementing a step (by parallelizing matrix-vector multiplications, matrix inversions, etc.)
♠ **Distributed Optimization** is inspired by modern Cloud storage of data and computations and is about solving optimization problems (usually, convex and well-structured) in *distributed setting*, where there are several interacting processors ("agents") and
- problem's data is somehow distributed among the processors
- we should take into account the cost of communicating information between the agents.

Example: We want to minimize $f(x) = \sum_{i=1}^{N} f_i(x)$ in the situation when
— i-th agent, $i = 1, ..., N$, has direct access to information on i-th term f_i only (say, can call First Order oracle reporting the values and the subgradients of f_i at query points)
— the agents form nodes in a graph, and in a single interchange act (which takes unit time) an agent i can forward information to agent j iff the nodes i and j are adjacent.

♠ *The necessity to account for communication costs results in significant and highly novel challenges in design and analysis of optimization algorithms*, and these challenges are the subject of intensive ongoing research.

Disclaimer: *In what follows, I restrict myself with the traditional MP paradigm.*

♣ **Claim:** *Algorithmic and computational toolbox for solving general-type Mathematical Programming problems*

$$\min_x \left\{ f(x) : \begin{array}{l} g_1(x) \leq 0, \ldots, g_m(x) \leq 0 \\ h_1(x) = 0, \ldots, h_k(x) = 0 \end{array} \right\}$$

is essentially complete, and its further development seems to be a relatively dead research area.

At least, I am not aware of any essential progress in this area during the last 15 years, except for *Newton method with cubic regularization* for smooth unconstrained minimization (Yu. Nesterov, B. Polyak, 2005) and *primal-dual interior point method(s) for smooth nonconvex constrained minimization* (software IPOPT, A. Waechter et al.).

$$\min_x \left\{ f(x) : \begin{array}{l} g_1(x) \le 0, \ ..., g_m(x) \le 0 \\ h_1(x) = 0, ..., h_k(x) = 0 \end{array} \right\} \qquad (*)$$

Note:

- Generality means that *all* we intend to use when building an algorithm is that

  — the objective is called f, the constraints are called $g_1, ..., g_m$, $h_1, ..., h_k$, and these functions are smooth;
  — we can compute the values and the derivatives (first, second,...) of the objective and the constraints at any point.

- Stagnation in the area comes from the fact that *optimizers ran out of novel ideas*, and *not* from the fact that the existing algorithms satisfy all our needs.

However: *Never say "never"!*

Note: What seems to be dead, is *creating* novel general-purpose algorithms for Mathematical Programming problems, not *developing new software* and *application* of existing MP algorithms (perhaps properly adjusted) to novel optimization models arising in applications.

♡ *Modeling* real-world situations as optimization problems in many cases poses highly challenging theoretical questions, and thus is a quite respectful and rapidly developing research area.

♡ **Good modeling** seems to be **the** key to successful application of Mathematical Programming techniques.

♣ A model is good, when

- it reflects reasonably well the most important dependencies, design specifications and tradeoffs of the situation we intend to model.

  To achieve this goal, you should understand well the application area in question.

- it allows for subsequent efficient numerical processing of the resulting optimization model.

  To achieve this goal, you should know what can be expected from existing optimization techniques as applied to optimization problems of various types.

Note: The outlined requirements somehow contradict each other – usually, the more adequate is a model, the more difficult it is for numerical processing. Finding reasonable tradeoff here requires a lot of knowledge (both in the relevant subject area *and* in Optimization) and some luck...

Both half a laptop and half a truck are nonexisting entities. However,

— it would be counter-productive to model planning laptop production as an optimization problem with integrality constraints on the outcome;

— it would be equally counter-productive to ignore integrality constraints when modelling vehicle routing problem for a small delivery firm with a fleet of 5 – 10 trucks...

♣ It seems that one of the major problems with applications of Mathematical Programming comes from the fact that

> *More often than not, potential clients are completely unaware of what Optimization can do well and what is problematic, and as a result arrive with "dirty" models badly suited for numerical processing* (if they arrive at all – in many cases they simply do not know that Optimization exists and/or can be of use for them).

Responsibility for this is partly on optimizers who do not care enough to educate potential clients...

A man searches for a lost wallet at the place where the wallet was lost.
A wise man searches at a place with enough light...

♣ Where should we search for a wallet? Where is "enough light" – what Optimization can do well?

The most straightforward answer is: we can solve well *convex* optimization problems.

The very existence of what is called Mathematical Programming stemmed from discovery of Linear Programming (George Dantzig, late 1940's) – a modelling methodology accompanied by extremely powerful in practice (although "theoretically bad") computational tool – Simplex Method. Linear Programming still underlies the majority of real life applications of Optimization, especially large-scale ones.

♣ Around mid-1970's, it was shown that

• Linear and, more generally, Convex Programming problems are *efficiently solvable* – under mild computability and boundedness assumptions, generic Convex Programming problems admit *polynomial time* solution algorithms.
As applied to an instance of a generic problem, a polynomial time algorithm solves it to a whatever high accuracy ϵ in the number of steps which is polynomial in the *size* of the instance (the number of data entries specifying the instance) and the number $\ln(1/\epsilon)$ of required accuracy digits.
$\Rightarrow$ Theoretical (and to some extent – also practical) possibility to solve convex programs of reasonable size to high accuracy in reasonable time

• No polynomial time algorithms for general-type nonconvex problems are known, and there are strong reasons to believe that no such methods exist.
$\Rightarrow$ Solving general nonconvex problems of not too small sizes is usually a highly unpredictable process: with luck, we can improve somehow the solution we start with, but we usually do not know how far from global optimality we terminate.

Polynomial-Time Solvability of Convex Programming

♣ From purely academical viewpoint, polynomial time solvability of Convex Programming is a straightforward consequence of the following statement:

Theorem [circa 1976] *Consider a convex problem*

$$\mathrm{Opt} = \min_{x\in\mathbb{R}^n}\left\{f(x): \begin{array}{l} g_i(x)\le 0,\ 1\le i\le m \\ |x_j|\le 1,\ 1\le j\le n \end{array}\right\}$$

normalized by the restriction

$$|f(x)|\le 1, |g_j(x)|\le 1\ \forall x\in B=\{|x_j|\le 1\,\forall j\}.$$

For every $\epsilon\in(0,1)$*, one can find an* ϵ*-solution*

$$x_\epsilon\in B: f(x_\epsilon)-\mathrm{Opt}\le\epsilon, g_i(x_\epsilon)\le\epsilon$$

or to conclude correctly that the problem is infeasible at the cost of at most

$$3n^2\ln\left(\frac{2n}{\epsilon}\right)$$

computations of the objective and the constraints, along with their (sub)gradients, at subsequently generated points of $\mathrm{int}\,B$*, with* $n(n+m)$ *additional arithmetic operations per every such computation.*

♣ The outlined Theorem is sufficient to establish theoretical solvability of generic Convex Programming problems. In particular, it underlies the famous result (Leo Khachiyan, 1979) on polynomial time solvability of LP – the first ever mathematical result which made the C2 page of *New York Times* (Nov 27, 1979).

♣ From practical perspective, however, polynomial type algorithms suggested by Theorem are too slow: the arithmetic cost of an accuracy digit is at least

$$O(n^2 n(m+n)) \geq O(n^4),$$

which, even with modern computers, allows to solve in reasonable time problems with hardly more than 100 – 200 design variables.

♣ The low (although polynomial time) performance of the algorithms in question stems from the *black box oriented* nature of the algorithms – they do not adjust themselves to the structure of the problem and use a priori knowledge of this structure solely to mimic First Order oracle.

Note: A convex program *always* has a lot of structure – otherwise how could we know that the problem is convex?
A good algorithm should utilize a priori knowledge of problem's structure in order to accelerate the solution process.

> **Example:** The LP Simplex Method is fully adjusted to the particular structure of an LP problem. Although not a polynomial time one, this algorithm in reality is capable to solve LP's with tens and hundreds of thousands of variables and constraints – a task which is by far out of reach of the theoretically efficient "universal" black box oriented algorithms underlying the Theorem.

♣ Since mid-1970's, Convex Programming is the most rapidly developing area in Optimization, with intensive and successful research primarily focusing on

- discovery and investigation of novel well-structured generic Convex Programming problems ("Conic Programming," especially *Conic Quadratic* and *Semidefinite*)
- developing theoretically efficient and powerful in practice algorithms for solving well-structured convex programs, including large-scale nonlinear ones
- building Convex Programming models for a wide spectrum of problems arising in Engineering, Management, Medicine, etc.
- extending modeling methodologies in order to capture factors like data uncertainty typical for real world situations
- "on-line optimization," where our losses to be minimized can rapidly and unpredictably vary in time, and we are interested to make small the quantity

$$\frac{1}{T}\sum_{t=1}^{T} f_t(x_t) - \min_x \frac{1}{T}\sum_{t=1}^{T} f_t(x)$$

  - $f_t(\cdot)$: *unknown in advance* convex loss function at time $t = 1, ..., T$
  - x_t: our decision at time t which should be based solely on the past observations $f_\tau(x_\tau), f'_\tau(x_\tau)$, $\tau < t$
- "distributed optimization," where several agents are trying to minimize $f(x)$ by updating in parallel prescribed blocks in the decision vector x under restrictions on how the agents can exchange information
- software implementation of novel optimization techniques at academic and industry levels

"Structure-Revealing" Representation of Convex Problem: Conic Programming

♣ When passing from a Linear Programming problem

$$\min_x \left\{ c^T x : Ax - b \geq 0 \right\} \qquad (*)$$

to a nonlinear convex one, the traditional wisdom is to replace linear inequality constraints

$$a_i^T x - b_i \geq 0$$

with nonlinear convex ones:

$$g_i(x) \geq 0.$$

♠ There exists, however, another way to introduce nonlinearity, namely, to replace the coordinate-wise *vector* inequality

$$y \geq z \Leftrightarrow y - z \in \mathbb{R}^m_+ = \{u \in \mathbb{R}^m : u_i \geq 0 \, \forall i\} \qquad [y, z \in \mathbb{R}^m]$$

with another *vector* inequality

$$y \geq_{\mathbf{K}} z \Leftrightarrow y - z \in \mathbf{K} \qquad [y, z \in \mathbb{R}^m]$$

where $\mathbf{K}$ is a *closed, pointed and convex cone with a nonempty interior* in $\mathbb{R}^M$.

$$y \geq_{\mathbf{K}} z \Leftrightarrow y - z \in \mathbf{K} \qquad [y, z \in \mathbb{R}^m]$$

$\mathbf{K}$: closed, pointed and convex cone in $\mathbb{R}^m$ with a nonempty interior.

Requirements on $\mathbf{K}$ ensure that $\geq_{\mathbf{K}}$ obeys the usual rules for inequalities:

- $\geq_{\mathbf{K}}$ is a *partial order*:

$$\begin{array}{l} y \geq_{\mathbf{K}} y \, \forall y \\ (y \geq_{\mathbf{K}} z \ \& \ z \geq_{\mathbf{K}} y) \Rightarrow y = z \\ (x \geq_{\mathbf{K}} y, y \geq_{\mathbf{K}} z) \Rightarrow x \geq_{\mathbf{K}} z \end{array}$$

- $\geq_{\mathbf{K}}$ is compatible with linear operations: the validity of $\geq_{\mathbf{K}}$ inequality is preserved when we multiply both sides by the same nonnegative real and add to it another valid $\geq_{\mathbf{K}}$-inequality;

- in a sequence of $\geq_{\mathbf{K}}$-inequalities, one can pass to limits:

$$\begin{array}{c} a_i \geq_{\mathbf{K}} b_i, \, i = 1, 2, ... \ \& \ a_i \to a \ \& b_i \to b \\ \Downarrow \\ a \geq_{\mathbf{K}} b \end{array}$$

- one can define the strict version $>_{\mathbf{K}}$ of $\geq_{\mathbf{K}}$:

$$a >_{\mathbf{K}} b \Leftrightarrow a - b \in \operatorname{int} \mathbf{K}.$$

  Arithmetics of $>_{\mathbf{K}}$ and $\geq_{\mathbf{K}}$ inequalities is completely similar to the arithmetics of the usual coordinate-wise $\geq$ and $>$.

♣ **LP problem:**

$$\min_x \left\{c^T x : Ax - b \geq 0\right\} \Leftrightarrow \min_x \left\{c^T x : Ax - b \in \mathbb{R}^m_+\right\}$$

♣ **General Conic problem:**

$$\min_x \left\{c^T x : Ax - b \geq_{\mathbf{K}} 0\right\} \Leftrightarrow \min_x \left\{c^T x : Ax - b \in \mathbf{K}\right\}$$

- (A, b) – *data* of conic problem
- $\mathbf{K}$ - structure of conic problem

♠ **Note:** Every convex problem admits equivalent conic reformulation
♠ **Note:** With conic formulation, convexity is "built in"; with the standard MP formulation convexity should be kept in mind as an additional property.

♣ (??) A general convex cone has no more structure than a general convex function. Why conic reformulation is "structure-revealing"?

♣ (!!) As a matter of fact, just 3 types of cones allow to represent an extremely wide spectrum ("essentially all") convex problems!

$$\boxed{\min_x \left\{c^T x : Ax - b \geq_{\mathbf{K}}\right\} \Leftrightarrow \min_x \left\{c^T x : Ax - b \in \mathbf{K}\right\}}$$

♠ Three Magic Families of cones:

- Direct products $\mathbb{R}^m_+$ of nonnegative rays $\mathbb{R}_+ = \{s \in \mathbb{R} : s \geq 0\}$ (nonnegative orthants) giving rise to Linear Programming programs

$$\min_s \left\{c^T x : a_\ell^T x - b_\ell \geq 0, 1 \leq \ell \leq q\right\}.$$

- Direct products of Lorentz cones $\mathbf{L}^p_+ = \{u \in \mathbb{R}^p : u_p \geq \left(\sum_{i=1}^{p-1} u_i^2\right)^{1/2}\}$ giving rise to Conic Quadratic programs

$$\min_x \left\{c^T x : \|A_\ell x - b_\ell\|_2 \leq c_\ell^T x - d_\ell, 1 \leq \ell \leq q\right\}.$$

- Direct products of Semidefinite cones $\mathbf{S}^p_+ = \{M \in \mathbf{S}^p : M \succeq 0\}$ giving rise to Semidefinite programs

$$\min_x \left\{c^T x : \lambda_{\min}(\mathcal{A}^\ell(x)) \geq 0,\ 1 \leq \ell \leq q\right\}.$$

  where $\mathcal{A}_\ell(x)$ are symmetric matrices affine in x and $\lambda_{\min}(S)$ is the minimal eigenvalue of a symmetric matrix S.

♣ Conic Programming admits nice Duality Theory completely similar to LP Duality.
Primal problem:

$$\min_x \left\{c^T x : Ax - b \geq_{\mathbf{K}} 0\right\} \Leftrightarrow \min_\xi \left\{e^T \xi : \xi \in [\mathcal{L} - b] \cap \mathbf{K}\right\}$$
$$[\mathcal{L} = \mathrm{Im} A,\ A^T e = c,\ \mathrm{Ker} A = \{0\}]$$

Dual problem:

$$\max_\lambda \left\{b^T \lambda : \lambda \in \left[\mathcal{L}^\perp + e\right] \cap \mathbf{K}_*\right\} \Leftrightarrow \max_\lambda \left\{b^T \lambda : A^T \lambda = c, \lambda \geq_{\mathbf{K}_*} 0\right\}$$
$$[\mathbf{K}_* = \{\lambda : \lambda^T \xi \geq 0\ \forall \xi \in \mathbf{K}\}]$$

Note: $\mathbf{K}_*$ is a closed pointed convex cone with a nonempty interior (called the cone dual to $\mathbf{K}$), and $(\mathbf{K}_*)_* = \mathbf{K}$. Thus,

- the dual problem is conic along with the primal
- the duality is completely symmetric

Note: Cones from Magic Families are self-dual, so that the dual of a Linear/Conic Quadratic/Semidefinite program is of exactly the same type.

$$\min_x \left\{c^T x : Ax - b \geq_{\mathbf{K}} 0\right\} \Leftrightarrow \min_\xi \left\{e^T \xi : \xi \in [\mathcal{L} - b] \cap \mathbf{K}\right\} \qquad (P)$$

$$\max_\lambda \left\{b^T \lambda : \lambda \in [\mathcal{L}^\perp + e] \cap \mathbf{K}_*\right\} \Leftrightarrow \max_\lambda \left\{b^T \lambda : A^T \lambda = c, \lambda \geq_{\mathbf{K}_*} 0\right\} \qquad (D)$$

$$\left[\ \mathcal{L} = \mathrm{Im} A,\ A^T e = c, \mathbf{K}_* = \{\lambda : \lambda^T \xi \geq 0\ \forall \xi \in \mathbf{K}\}\ \right]$$

Conic Programming Duality Theorem:

- [Symmetry] Conic Duality is fully symmetric: the dual problem is conic, and its dual is (equivalent to) the primal
- [Weak Duality] $\mathrm{Opt}(D) \leq \mathrm{Opt}(P)$
- [Strong Duality] If one of the problems is strictly feasible (i.e., the corresponding affine plane intersects the interior of the underlying cone) and bounded, then the other problem is solvable, and $\mathrm{Opt}(D) = \mathrm{Opt}(P)$. In particular, if both problems are strictly feasible, both are solvable with equal optimal values.

$$\min_x \left\{c^T x : Ax - b \geq_{\mathbf{K}} 0\right\} \Leftrightarrow \min_\xi \left\{e^T \xi : \xi \in [\mathcal{L} - b] \cap \mathbf{K}\right\} \qquad (P)$$

$$\max_\lambda \left\{b^T \lambda : \lambda \in [\mathcal{L}^\perp + e] \cap \mathbf{K}_*\right\} \Leftrightarrow \max_\lambda \left\{b^T \lambda : A^T \lambda = c, \lambda \geq_{\mathbf{K}_*} 0\right\} \quad (D)$$

$$\left[\ \mathcal{L} = \mathrm{Im} A,\ A^T e = c, \mathbf{K}_* = \{\lambda : \lambda^T \xi \geq 0\ \forall \xi \in \mathbf{K}\}\ \right]$$

Conic Programming Optimality Conditions:

Let both (P) and (D) be strictly feasible. Then a pair (x, λ) of primal and dual *feasible* solutions is comprised of optimal solutions to the respective problems if and only if

- [Zero Duality Gap]

$$c^T x - b^T \lambda = 0,$$

and if and only if

- [Complementary Slackness]

$$[Ax - b]^T \lambda = 0.$$

$$\min_x \left\{c^T x : Ax - b \geq_{\mathbf{K}} 0\right\} \Leftrightarrow \min_\xi \left\{e^T \xi : \xi \in [\mathcal{L} - b] \cap \mathbf{K}\right\} \qquad (P)$$

$$\max_\lambda \left\{b^T \lambda : \lambda \in [\mathcal{L}^\perp + e] \cap \mathbf{K}_*\right\} \Leftrightarrow \max_\lambda \left\{b^T \lambda : A^T \lambda = c, \lambda \geq_{\mathbf{K}_*} 0\right\} \qquad (D)$$

♣ **Conic Duality**, same as the LP one, is

- *fully algorithmic:* to write down the dual, given the primal, is a purely mechanical process
- *fully symmetric:* the dual problem "remembers" the primal one

♡ Cf. Lagrange Duality:

$$\min_x \{f(x) : g_i(x) \le 0,\ i = 1, ..., m\} \qquad (P)$$

$$\Downarrow$$

$$\max_{\lambda \ge 0} \underline{\mathbf{L}}(\lambda) \qquad (D)$$

$$\left[\underline{\mathbf{L}}(\lambda) = \min_x \left\{ f(x) + \sum_i \lambda_i g_i(x) \right\}\right]$$

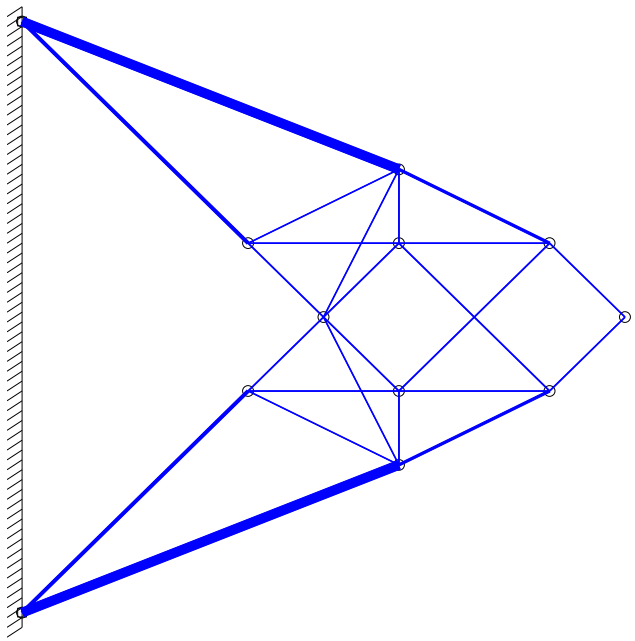
- Dual "exists in the nature," but is given implicitly; its objective, typically, is not available in a closed form
- Duality is asymmetric: given $\underline{\mathbf{L}}(\cdot)$, we, typically, cannot recover f and g_i...

♣ Conic Duality in the case of Magic cones:

- powerful tool to process problem, to some extent, "on paper," which in many cases provides extremely valuable insight and/or allows to end up with a reformulation much better suited for numerical processing
- is heavily exploited by efficient polynomial time algorithms for Magic conic problems

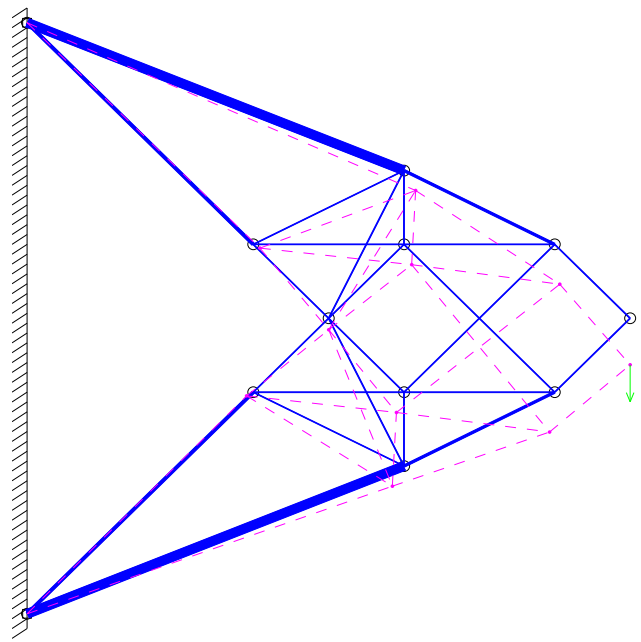
Example: Truss Topology Design

♣ A *truss* is a mechanical construction, like electric mast, railroad bridge, or Eiffel Tower, comprised of thin elastic *bars* linked with each other at *nodes:*



A console

♡ When a truss is subject to external *load* (collection of forces acting at the nodes), it deforms until the reaction forces caused by elongations/contractions of bars compensate the external force:



Loaded console

♡ At the equilibrium, the deformed truss capacitates certain potential energy – *compliance* of the truss w.r.t. the load.

♡ Compliance is a natural measure of the rigidity of the truss w.r.t. the load – the less is the compliance, the better.

♠ Mathematically:

- Displacements of a truss are identified with long vectors comprised of "physical" 2D/3D displacements of the nodes; these displacements form a linear space $V = \mathbb{R}^M$, where M is the total number of degrees of freedom of the nodes.

- An external load acting at a truss is identified with a long vector $f \in V$ comprised of "physical" 2D/3D forces acting at the nodes

- Assuming deformation small, the reaction forces caused by the deformation form the long vector

$$A(t)v$$

- v : displacement
- $A(t) = \sum_{i=1}^{N} t_i b_i b_i^T$: *stiffness matrix*

  t_i : volume of bar i

  b_i : readily given by geometry of nodal set

- Equilibrium displacement v solves

$$A(t)v = f$$

and the compliance is

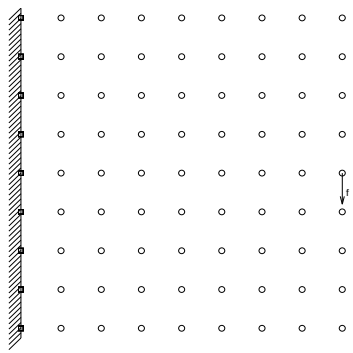
$$\begin{array}{rcl} \mathrm{Compl}_f(t) & = & \frac{1}{2} f^T v \\ & = & \frac{1}{2} v^T A(t) v \\ & = & \frac{1}{2} f^T A^{-1}(t) f \end{array}$$

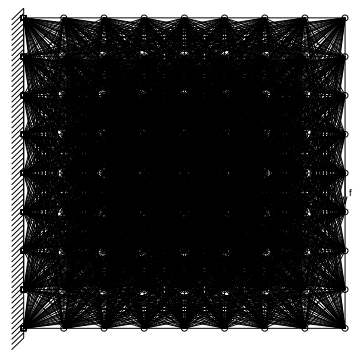
♣ In the simplest Truss topology Design problem one is given

- *Ground Structure:*
  — the 2D/3D set of tentative nodes along with boundary conditions fully or partially restricting displacements of some nodes
  — the set of tentative bars

- *load of interest* f

and seeks for the truss of a given total weight with minimum possible compliance w.r.t. f.



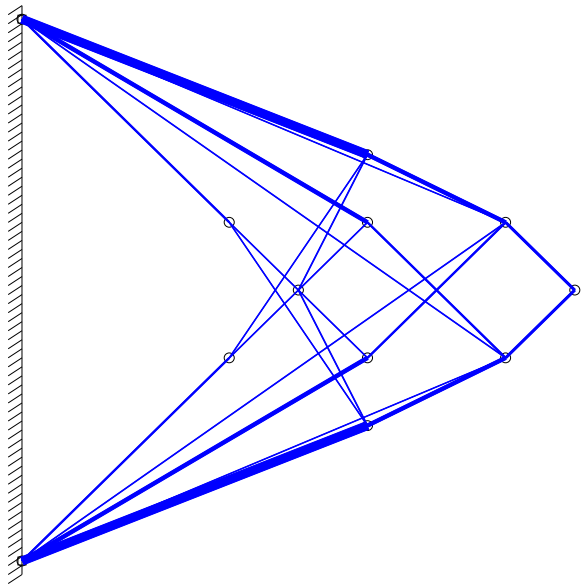


9×9 nodal grid and load $N = 2,039$ tentative bars

$$\Downarrow$$

$$\boxed{\min_{t\in\mathbb{R}^N,\tau}\left\{\tau : \begin{array}{c}\left[\begin{array}{c|c} 2\tau & f^T \\ \hline f & \sum_{i=1}^N t_i b_i b_i^T \end{array}\right] \succeq 0 \\ t \geq 0, \sum_i t_i \leq W\end{array}\right\}}$$

$$\Downarrow$$



Optimal console

♣ When solving a TTD problem, one starts with a dense nodal grid and allows for all pair connections of nodes by tentative bars.
At the optimal solution, most of these tentative bars get zero volume, and the design reveals *optimal topology*, not merely optimal sizing!

♠ **<u>However:</u>** To reveal optimal topology, one needs to work with dense nodal grids (M like few thousands, especially in 3D) and to allow for all tentative bars ($N \approx \frac{M^2}{8}$ in 2D and $M \approx \frac{N^2}{18}$ in 3D), which results in really huge semidefinite problems – millions of variables!

♠ **Remedy:** Conic Duality.

Applying Conic Duality to the semidefinite TTD program (this is a purely mechanical process!) one ends up with another semidefinite program. *This program admits analytical elimination of most of the variables* and is equivalent to the semidefinite program

$$\min_{v,\gamma}\left\{-2f^Tv+W\gamma : \begin{bmatrix} 1/2 & b_i^Tv \\ b_i^Tv & \gamma \end{bmatrix} \succeq 0,\ 1\le i\le N\right\}$$

♡ The dimension of this program is just $M+1$ – incomparably less than the dimension $N=O(M^2)$ of the primal TTD problem!
♡ In addition, the primal TTD has a single "big" LMI, while the dual one has N small 2×2 LMI's.
♣ When solving the primal TTD by the best known optimization methods, the price of accuracy digit is as large as $O(M^{1/2}N^3)=O(M^{6.5})$ operations, which for real life values of M is *by far* beyond our computational abilities.
♣ For the (transformed) dual problem, the price of accuracy digit is $O(N^{1/2}M^3)=O(M^4)$ operations, which is tolerable...

$$\boxed{\min_{t\in\mathbb{R}^N,\tau}\left\{\tau : \begin{array}{c}\left[\begin{array}{c|c} 2\tau & f^T \\ \hline f & \sum\limits_{i=1}^N t_i b_i b_i^T \end{array}\right] \succeq 0 \\ t \geq 0, \sum\limits_i t_i \leq W \end{array}\right\}} \quad \text{(TTD)}$$

$$\Downarrow$$

$$\boxed{\min_{v,\gamma}\left\{-2f^Tv + W\gamma : \left[\begin{array}{cc} 1/2 & b_i^Tv \\ b_i^Tv & \gamma \end{array}\right] \succeq 0,\ 1 \leq i \leq N\right\}} \quad \text{(D)}$$

♡ Semidefinite problem (D) is not exactly the dual of (TTD) – it is obtained from the dual by analytical partial optimization w.r.t. part of the variables.
If we were taking the problem dual to dual, we would recover (TTD). What happens when we pass from (D) to *its* dual?

<u>Answer:</u> We will get a highly nontrivial and instructive *equivalent reformulation* of (TTD):

$$\min_{q,t}\left\{\sum_i \frac{q_i^2}{t_i} : \begin{array}{l} \sum\limits_i q_i b_i = f \\ \sum\limits_i t_i \leq W, t \geq 0 \end{array}\right\}$$

$$\boxed{\min_{t\in\mathbb{R}^N,\tau}\left\{\tau : \begin{array}{c}\left[\begin{array}{c|c} 2\tau & f^T \\ \hline f & \sum\limits_{i=1}^N t_i b_i b_i^T \end{array}\right] \succeq 0 \\ t \geq 0, \sum\limits_i t_i \leq W \end{array}\right\}} \quad \text{(TTD)}$$

$$\Updownarrow$$

$$\boxed{\min_{q,t}\left\{\sum_i \frac{q_i^2}{t_i} : \begin{array}{c}\sum\limits_i q_i b_i = f \\ \sum\limits_i t_i \leq W, t \geq 0\end{array}\right\}} \quad \text{(TTD}^+\text{)}$$

♡ On a closest inspection, (TTD$^+$) is just a *Linear Programming* problem! (This miracle happens only in the simplest single-load TTD problem. It does not survive even nontrivial upper and lower bounds on bar volumes...)

♡ Up to the LP miracle, the above story can be repeated for pretty general Structural Design problems (Truss and Shape Design with several loading scenarios, bounds on variables, obstacles,...) In all these problems

- The problem of interest can be posed as SDP
- Applying Conic Duality, one can simplify the dual problem analytically to end up with a semidefinite problem much better suited for numerical processing than the original formulation
- Passing from the transformed dual to its dual, one gets a nontrivial and instructive equivalent reformulation of the original problem

$$\boxed{\min_{q,t}\left\{\sum_i \frac{q_i^2}{t_i} : \begin{array}{l}\sum_i q_i b_i = f \\ \sum_i t_i \leq W, t \geq 0\end{array}\right\}} \quad (\mathrm{TTD}^+)$$

♣ (TTD^+) has a transparent mechanical interpretation:
— q_i is can be thought of as the product of the tension caused by deformation of i-th bar and the cross-sectional area of the bar;
— constraint $\sum_i q_i b_i = f$ says exactly that reaction forces coming from the tensions should compensate the external forces.

♣ **<u>However:</u>** you *cannot* just write down (TTD^+) from purely mechanical considerations: in reality, N tensions of the bars come from $M << N$ displacements of the nodes, and (TTD^+) does *not* include such a constraint!

<u>Explanation:</u> *At the optimum*, q_i indeed come from M displacements (which, mathematically, are the Lagrange multipliers of the equality constraints $\sum_i q_i b_i = f$)!

♣ While *post factum* you can explain (TTD$^+$) from purely mechanical perspective (also in the multi-load case, with obstacles, etc.) nobody was smart enough to discover this formulation from scratch. It was discovered exactly as explained – via twice used Conic Duality!

Morality: Conic Formulation of a convex program and Conic Duality is much more than a tool for number-crunching!

Other known to me important results stemming from Conic Duality include

- tightness results for tractable approximations of various intractable problems
- stability analysis of uncertain linear dynamical systems
- synthesis of near-optimal linear controllers for disturbance-affected linear dynamical systems
- computationally efficient robust – immunized against data uncertainty – decision making
- computationally efficient statistically near-optimal recovery of signals x from their indirect noisy observations

$$y = Ax + \xi$$

[A: known sensing matrix, ξ: observation noise]

Polynomial Time Algorithms for Well-Structured Convex Programs

♣ The first polynomial time algorithm capable to utilize the structure of a convex problem (namely, a LP one) was discovered by Narendra Karmarkar (1984).
While Karmarkar's algorithm did not improve much the already known polynomial time LP complexity bounds, it was completely novel and turned out to be competitive with Simplex Method.

♣ A real shock caused by Karmarkar's algorithm opened what is now called "Interior Point Revolution" (mid-1980's – late 1990's). In course of this revolution effort of many tens of first-rate researchers led to

- improving theoretical complexity bounds for LP and developing new theoretically *and practically* efficient polynomial time algorithms for LP
- developing general theory of interior point polynomial methods capable to understand intrinsic nature of the IP LP algorithms and to extend them on the nonlinear well-structured convex problems, most notably the conic problems of Magic cones
- industry-level software implementation of IP algorithms for LP/CQP (CPLEX) and LP/CQP/SDP (latest version of MOSEK - MOSEK 7.0).

♣ As a result of Interior Point Revolution,

- essentially, the entire Convex Programming is within the reach of powerful IP polynomial time methods
- practical performance of Convex Optimization techniques was improved by factor about 10^6, with nearly equal contributions of progress in software and progress in algorithms

Challenge: extremely large-scale nonlinear convex programs, primarily SDP's.

♣ IPM's are Newton-type algorithms – at every step they solve $n \times n$ systems of linear equations, n being the design dimension of the problem.

Due to polynomial-time convergence, it takes a moderate number (10 – 40) Newton steps to solve the problem to high accuracy.

♠ **<u>However:</u>** To solve in a realistic time a system of linear equations with $n \sim 10^5$ or more variables is possible *only* when the system is highly sparse. This indeed happens with typical LP's (and to some extent - CQP's) of real life origin, *but almost never happens with SDP's.*

$\Rightarrow$ Really large-scale SDP's (and many other nonlinear convex problems) are beyond the grasp of IPM's – fast convergence does not help when the very first iteration lasts forever...

♣ With design dimension $n \sim 10^5$–10^6, the only realistic option is to use simple methods with (nearly) linear in n cost of an iteration. At the present level of our knowledge, the only methods meeting this requirement are simple gradient-type algorithms.

♠ Gradient-type algorithms are black-box oriented and in the large-scale case cannot exhibit linear convergence, only a sublinear one.
♣ **<u>However:</u>** For problems with favorable geometry, the rate of convergence of smart gradient-type algorithms is (nearly) dimension-independent, which makes these algorithms well-suited for finding medium-accuracy solutions of extremely large-scale convex problems. Could we further improve these algorithms by utilizing problem's structure? Yes! Such a possibility was discovered (Yuri Nesterov, 2003), and the resulting *fast gradient algorithms* form an extremely popular and rapidly developing research area with high (and partly already realized) applied potential.

Novel Applied Convex Optimization Models

♣ Dramatic methodological (discovery of Conic Optimization, especially CQP and SDP) and algorithmic (IPM's) progress in Convex Optimization has inspired (and was inspired by) huge activity in building of well-structured convex optimization models in various applications, including, but not reduced to,

- Control
- Communications
- Design of mechanical structures
- Design of circuits and chips
- Signal Processing, in particular, Medical Imaging
- Machine Learning and Data Mining
- .....

♣ Along with constantly extending applications outside of Optimization, Convex Programming, and primarily SDP, is extensively used within Optimization, most notably as the working horse for processing difficult combinatorial problems.

(Relatively) Novel Optimization Approaches and Methodologies

Let us present just two examples:

♣ Systematic search for *approximation algorithms* – polynomial time algorithms for building *sub*optimal solutions for difficult (e.g., combinatorial) problems.

Approximation algorithm for a generic difficult optimization problem must be

— efficient – a polynomial time one

— as applied to every instance of the problem, produce a *feasible* approximate solution x which is within an absolute constant factor of the optimal solution in terms of the objective:

$$\frac{\text{Objective at } x}{\text{True optimal value}} \leq O(1)$$

Algorithms of this type are known for many NP-hard optimization problems...

♣ Attention to data uncertainty – *Robust Optimization*.

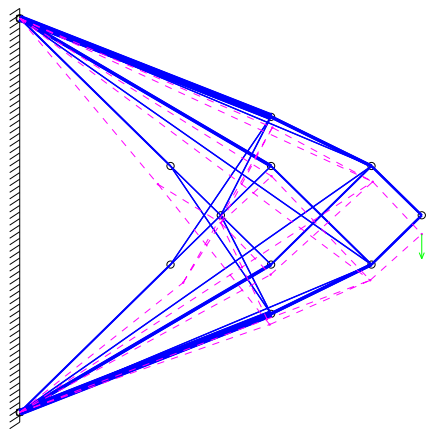
Example: TTD revisited. When designing the console, we took care about the only load – the one we are actually interested in. In reality, however, the console will be subject to other loads, perhaps small, but it still should be capable to carry them.

Equivalently: The data f in the TTD problem

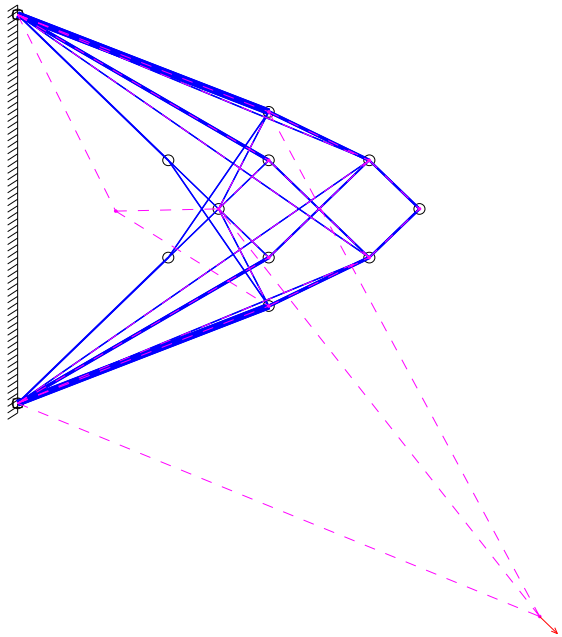
$$\boxed{\min_{t\in\mathbb{R}^N,\tau}\left\{\tau : \begin{array}{c}\left[\begin{array}{c|c} 2\tau & f^T \\ \hline f & \sum\limits_{i=1}^N t_i b_i b_i^T \end{array}\right] \succeq 0 \\ t\geq 0, \sum\limits_i t_i \leq W\end{array}\right\}}$$

is *uncertain* – running in a "massive set" $\mathcal{F}$ (containing *at least* the load of interest f_* and all small enough occasional loads), and a meaningful candidate solution should be *robust feasible* – it should remain feasible for all realizations of the data from $\mathcal{F}$.

How robust is the nominally optimal design?



Deformation under the load
of interest (10,000 kg)



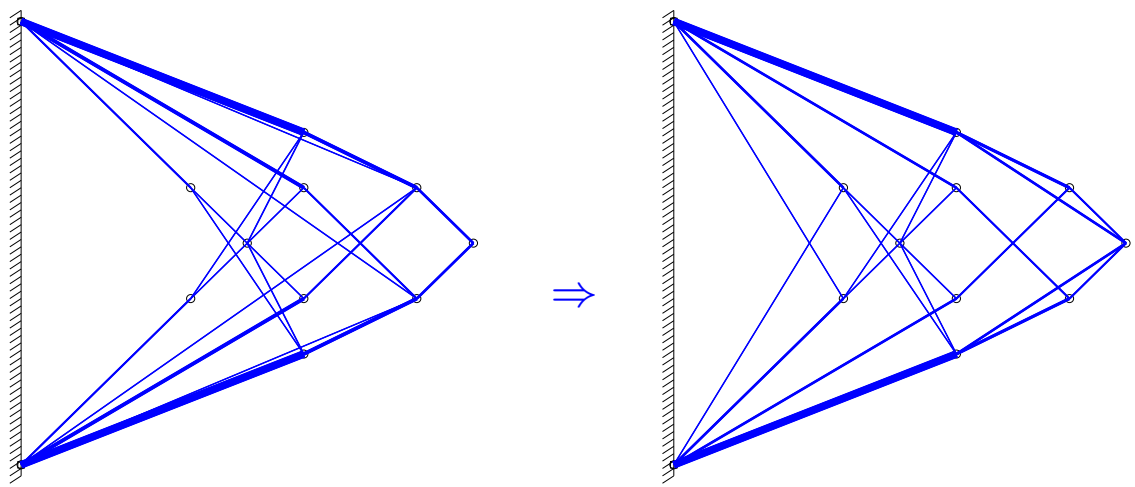
Deformation under "badly placed"
load 10^8 times less than
the load of interest (0.1 g)

♣ In Optimization, there exists a necessity to "immunize" solutions against data uncertainty.

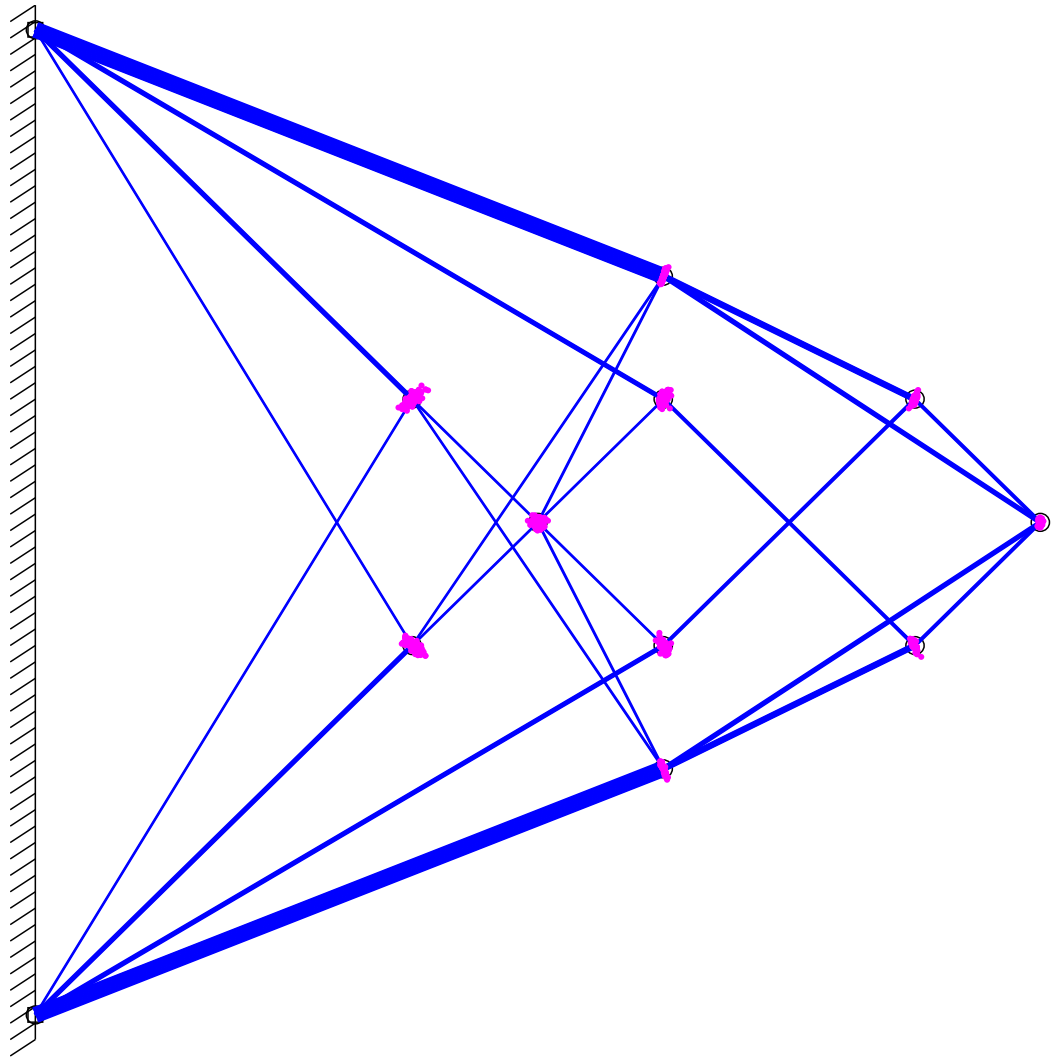
Robust Optimization is a relatively novel and rapidly developing methodology which takes data uncertainty into account from the very beginning and looks for solutions which are "immunized" against this uncertainty.

Development of RO poses highly challenging research questions and possesses huge practical potential.

Example (continued). Applying RO to the TTD problem, we end up with *robust design*



which carries the load of interest by just 2.5% worse than the nominal design, and is capable *equally well* withstand *all* occasional loads as large as 36% of the load of interest!



Nodal displacements of robust console,
sample of 100 occasional loads
10% of the load of interest

Lecture 15:
First Order Methods
for
Large-Scale Convex Minimization

For details, see Lecture 5 in
http://www.isye.gatech.edu/~nemirovs/LMCOLN2021WithSol.pdf

Simple methods for extremely large-scale problems

♣ The arithmetic complexity of a step in *all* known Convex Programming algorithms *capable to solve convex problems to high accuracy*, like Ellipsoid Algorithms or Path-Following Interior Point methods, grows up *nonlinearly* with the design dimension n of the problem – at least as $O(n^2)$, if not as $O(n^3)$ (the only exception are extremely sparse real-world LPs with favourable sparsity patterns).

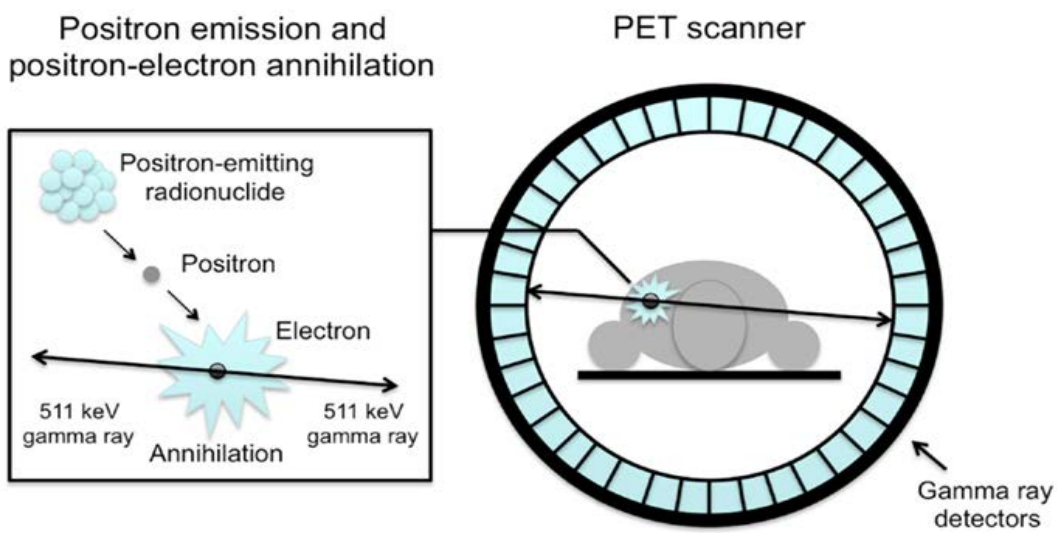
What to do when the design dimension is of order of tens and hundreds of thousands, and the problem is not a "very sparse LP"?

Nonlinear convex problems of huge design dimension do arise in numerous applications, e.g., in

- Structural Design (especially for 3D structures),
- Signal Processing, High-dimensional Statistics, Machine Learning
- *3D Medical imaging problems*

Example of Medical Imaging problem: PET Image Reconstruction

♣ **PET** (Positron Emission Tomography) is a powerful, non-invasive, medical diagnostic imaging technique for measuring the metabolic activity of cells in the human body. It has been in clinical use since the early 1990s. PET imaging is unique in that it shows the *chemical functioning* of organs and tissues, while other imaging techniques - such as X-ray, computerized tomography (CT) and magnetic resonance imaging (MRI) - show *anatomic structures*.



♣ **Physics of PET.** A PET scan uses *radioactive tracer* – a biologically active fluid with a radio-active component capable of emitting positrons. When administered to a patient, the tracer distributes within the body and, with properly chosen biologically active "carrier", concentrates in desired locations, e.g., in the areas of high metabolic activity where cancer tumors can be expected.

- The tracer disintegrates, emitting positrons.
- A positron immediately annihilates with a near-by electron, giving rise to two photons flying at the speed of light off the point of annihilation in nearly opposite directions. They are registered outside the patient by cylindrical PET scanner consisting of several rings of detectors.
- When two detectors "simultaneously" (within $\sim 10^{-8}$ sec time window) are hit by photons, this event is registered, indicating that somewhere on the line linking the detectors (LOR – "Line of Response") a disintegration act took place.

• The measured data is the collection of numbers of LOR's counted by different pairs of detectors ("bins"), and the problem is to recover from these measurements the 3D density of the tracer.

♣ Mathematically, the PET Image Reconstruction problem, after appropriate discretization, becomes the problem of recovering a vector $\lambda \geq 0$ from a noisy observation y of the vector $P\lambda$:

$$\lambda \mapsto y = P\lambda + \text{noise} \quad ? \mapsto ? \quad \text{estimate of } \lambda.$$

Specifically,

• entries of λ are indexed by *voxels* – small cubes into which we partition the field of view; λ_j is the average density of the tracer in voxel j;

• entries of y are indexed by bins (pairs of detectors); y_i is the number of LORs registered by bin i;

• $P = [p_{ij}]$ is a given matrix; p_{ij} is the probability for a LOR originating in voxel j to be registered by bin i.

Statistical model of PET states that the entries y_i in y are realizations of independent Poisson random variables with the expectations $(P\lambda)_i$.

♡ In the PET Reconstruction problem, we are interested, given observations y, to find the Maximum Likelihood estimate λ_* of tracer's density:

$$\lambda_* = \operatorname*{argmin}_{\lambda \geq 0} \left[\sum_{j=1}^{n} p_j \lambda_j - \sum_{i=1}^{m} y_i \ln(\sum_j p_{ij} \lambda_j) \right] \qquad [p_j = \sum_i p_{ij}] \qquad \text{(PET)}$$

(PET) is a nicely structured constrained convex program; the only difficulty – a true one! – is in huge sizes of (PET): for problems of actual interest,

- the design dimension n varies from 300,000 to 3,000,000
- the number m of log-terms in the objective varies from 6,000,000 to 25,000,000

♣ As far as nonlinear programs are concerned, design dimension $n \sim 10^4 - 10^5 - 10^6$ makes it necessary to use "cheap" algorithms – those with nearly linear in n arithmetic cost of a step (otherwise you never will finish the very first iteration). This requirement rules out all "advanced" polynomial time optimization techniques and leaves us with, essentially, just two options:

I. Traditional tools of *smooth unconstrained* minimization: gradient descent, conjugate gradients, quasi-Newton methods, etc.

II. Simple subgradient-type techniques for solving *convex nonsmooth constrained* optimization problems:

subgradient descent, restricted memory bundle methods, etc.

• We are interested in extremely large-scale constrained convex problems, and thus intend to focus on cheap subgradient-type techniques. The question of primary importance here is:
(?) *What are the limits of performance of cheap optimization techniques?*

• When answering (?), we shall restrict ourselves with the *black-box-represented* convex programs. As a matter of fact, this is exactly the "working environment" for cheap optimization algorithms.

Black-box-represented convex programs and Information-based complexity

♣ Let us fix a family $\mathcal{P}(X)$ of convex programs

$$\min_x \{f(x) : x \in X\}; \qquad \text{(CP)}$$

where $X \subset \mathbb{R}^n$ is a given *instance-independent* convex compact set, and $f : \mathbb{R}^n \to \mathbb{R}$ is convex.

• Formally, $\mathcal{P}(X)$ is some family of convex objectives $f : X \to \mathbb{R}$.

$$\min_x \{f(x) : x \in X\}; \qquad \text{(CP)}$$

♣ A *black-box-oriented* solution method $\mathcal{B}$ for $\mathcal{P}(X)$ is as follows:

• When starting to solve (CP), $\mathcal{B}$ is given an accuracy $\epsilon > 0$, knows what is X, and knows that f belongs to a given family $\mathcal{P}(X)$. However, $\mathcal{B}$ does *not* know in advance what is the particular f it deals with and must "learn" f to solve the problem.

• When solving the problem, $\mathcal{B}$ has access to the First Order oracle for f. Given on input $x \in \mathbb{R}^n$, the oracle returns $f(x)$ and a subgradient $f'(x)$ of f at x. $\mathcal{B}$ generates a sequence of *search points* $x_1, x_2, ...$ and calls the First Order oracle to get values and subgradients of f at these points. The rules for building x_t can be arbitrary, *except for the fact that they should be non-anticipative*: x_t can depend only on the information $f(x_1), f'(x_1), ..., f(x_{t-1}), f'(x_{t-1})$ on f accumulated by $\mathcal{B}$ at the first $t-1$ steps.

• After a number $T = T_{\mathcal{B}}(f, \epsilon)$ of calls to the oracle, $\mathcal{B}$ terminates and outputs a result $z_{\mathcal{B}}(f, \epsilon)$ which *should depend solely on the information on f accumulated by $\mathcal{B}$ at the T search steps*, and *must be an ϵ-solution to* (CP):

$$z_{\mathcal{B}}(f, \epsilon) \in X \ \& \ f(z_{\mathcal{B}}(f, \epsilon)) - \min_X f \leq \epsilon.$$

♣ The *complexity* of $\mathcal{P}(X)$ w.r.t. a solution method $\mathcal{B}$ is

$$\mathrm{Compl}_{\mathcal{B}}(\epsilon) = \max_{f \in \mathcal{P}(X)} T_{\mathcal{B}}(f, \epsilon)$$

which is the minimal number of steps sufficient for $\mathcal{B}$ to solve within accuracy ϵ every instance of $\mathcal{P}(X)$.

♣ The *Information-based complexity* of a family $\mathcal{P}(X)$ of problems is

$$\mathrm{Compl}(\epsilon) = \min_{\mathcal{B}} \mathrm{Compl}_{\mathcal{B}}(\epsilon),$$

the minimum being taken over all solution methods. Relation

$$\mathrm{Compl}(\epsilon) = N$$

means that

- there exists a solution method $\mathcal{B}$ capable to solve within accuracy ϵ every instance of $\mathcal{P}(X)$ in no more than N calls to the First Order oracle;
- for every solution method $\mathcal{B}$, there exists an instance of $\mathcal{P}(X)$ such that $\mathcal{B}$ solves the instance within the accuracy ϵ in at least N steps.

♣ The information-based complexity $\mathrm{Compl}(\epsilon)$ of a family $\mathcal{P}(X)$ is a *lower* bound on "actual" computational effort, whatever it means, sufficient to find ϵ-solution to every instance of the family.

Main results on Information-based complexity of Convex Programming

♣ Let

$$\begin{array}{rcl} X & \subset & \mathbb{R}^n - \text{a convex compact set, } \operatorname{int} X \neq \emptyset \\ \mathcal{P}(X) & = & \left\{ \left\{ \min\limits_{x \in X} f(x) \right\} : f \text{ is convex on } \mathbb{R}^n \text{ and is normalized by } \max\limits_X f - \min\limits_X f \leq 1. \right\} \end{array}$$

For the family $\mathcal{P}(X)$,
I. Complexity of finding high-accuracy solutions in fixed dimension is independent of the geometry of X. Specifically,

$$\begin{array}{ll} \forall(\epsilon \leq \epsilon(X)): & O(1) n \ln\left(2 + \frac{1}{\epsilon}\right) \leq \mathrm{Compl}(\epsilon); \\ \forall(\epsilon > 0): & \mathrm{Compl}(\epsilon) \leq O(1) n \ln\left(2 + \frac{1}{\epsilon}\right), \end{array}$$

where
$O(1)$ are appropriately chosen positive absolute constants,
$\epsilon(X)$ depends on the geometry of X, but never is less than $\frac{1}{n^2}$.

$$\begin{aligned} X &\subset \mathbb{R}^n - \text{a convex compact set, } \operatorname{int} X \neq \emptyset \\ \mathcal{P}(X) &= \left\{ \{\min_{x\in X} f(x)\} : f \text{ is convex on } \mathbb{R}^n \text{ and normalized by } \max_X f - \min_X f \leq 1. \right\} \end{aligned}$$

II. Complexity of finding solutions of fixed accuracy in high dimensions does depend on the geometry of X. Here are 3 typical results:

Let $X = \{x \in \mathbb{R}^n : \|x\|_\infty \leq 1\}$. Then

$$\epsilon \leq \tfrac{1}{2} \Rightarrow O(1) n \ln(\tfrac{1}{\epsilon}) \leq \text{Compl}(\epsilon) \leq O(1) n \ln(\tfrac{1}{\epsilon}). \qquad (\|\cdot\|_\infty\text{-Ball})$$

Let $X = \{x \in \mathbb{R}^n : \|x\|_2 \leq 1\}$. Then

$$n \geq \frac{1}{\epsilon^2} \Rightarrow \frac{O(1)}{\epsilon^2} \leq \text{Compl}(\epsilon) \leq \frac{O(1)}{\epsilon^2}. \qquad (\|\cdot\|_2\text{-Ball})$$

Let $X = \{x \in \mathbb{R}^n : \|x\|_1 \leq 1\}$. Then

$$n \geq \frac{1}{\epsilon^2} \Rightarrow \frac{O(1)}{\epsilon^2} \leq \text{Compl}(\epsilon) \leq \frac{O(\ln n)}{\epsilon^2}. \qquad (\|\cdot\|_1\text{-Ball})$$

($O(1)$ in the lower bound can be replaced with $O(\ln n)$, provided that $n \gg \frac{1}{\epsilon^2}$).

$$\boxed{\mathrm{Compl}(\epsilon) \geq O(1) n \ln(2 + 1/\epsilon) \ \forall(\epsilon \leq \epsilon(X))} \tag{I}$$

$$\boxed{X = \{x \in \mathbb{R}^n : \|x\|_2 \leq 1\} \Rightarrow \mathrm{Compl}(\epsilon) \leq \frac{O(1)}{\epsilon^2} \forall(\epsilon > 0) :} \tag{II}$$

♣ Consequences for large-scale convex minimization:

Bad news: I says that we have no hope to guarantee high-accuracy solutions (like $\epsilon = 10^{-6}$) when solving large-scale problems with black-box-oriented methods: it would require at least $O(n)$ calls to the first order oracle with at least $O(n)$ a.o. per call, i.e., totally at least $O(n^2)$ a.o. (with known methods – even $O(n^4)$ a.o.), which is too much for large n...

Good news: II says that there exist cases when medium accuracy solutions can be found in (nearly) dimension-independent number of oracle calls...

♣ **Good news:** There exist cases when medium accuracy solutions of convex programs

$$\min_{x\in X} f(x), \quad \max_X f - \min_X f \le 1 \qquad (*)$$

can be found in (nearly) dimension-independent number of oracle calls, e.g., the cases of

$$X = B_n^2 \equiv \{x \in \mathbb{R}^n : \|x\|_2 \le 1\} \qquad (\|\cdot\|_2\text{-Ball})$$

or

$$X = B_n^1 \equiv \{x \in \mathbb{R}^n : \|x\|_1 \le 1\} \qquad (\|\cdot\|_1\text{-Ball})$$

(but, unfortunately, *not* the case when X is a box).

$$\min_{x\in X} f(x), \quad \max_X f - \min_X f \le 1 \qquad (*)$$

♣ Problems of minimizing over a $\|\cdot\|_p$-ball, $p = 1, 2$, are not that typical. Fortunately, the corresponding (nearly) dimension-independent complexity bounds remain valid when X in $(*)$ is a subset of a "good" set B_n^p, $p = 1, 2$, *and* the normalization condition on f in $(*)$ is strengthened to

$$|f(x) - f(y)| \le \|x - y\|_p \quad \forall x, y \in X.$$

In particular, $O(\frac{\ln n}{\epsilon^2})$ oracle calls are sufficient to minimize, within accuracy ϵ, a convex function f over the *standard simplex*

$$\Delta_n = \{x \in \mathbb{R}^n : x \ge 0, \sum_i x_i = 1\},$$

provided that f is Lipschitz continuous, with constant 1, w.r.t. $\|\cdot\|_1$ (i.e., that the magnitudes of all first order partial derivatives of f are ≤ 1).

♣ **More good news:** The nearly dimension independent complexity bounds for minimization over ball and simplex are given by cheap minimization methods!

Convention: From now on, speaking about optimization problem

$$\min_{x\in X} f(x), \qquad (*)$$

we assume *by default* that

- X is nonempty closed and bounded convex subset of Euclidean space E (by default, $E = \mathbb{R}^n$)
- $f(x) : X \to \mathbb{R}$ is convex and Lipschitz continuous:

$$\forall (x, y \in X) : |f(x) - f(y)| \leq L\|x - y\| \qquad [L < \infty]$$

Note: The property of f to be Lipschitz continuous is independent of the choice of norm $\|\cdot\|$ on E; in contrast, the allowed values of the *Lipschitz constant* L do depend on $\|\cdot\|$. In the sequel,

$$L_{\|\cdot\|}(f) = \sup_{x\neq y, x,y\in X} \frac{|f(x) - f(y)|}{\|x - y\|}$$

stands for the best – the smallest – of the Lipschitz constants, taken w.r.t. $\|\cdot\|$, of a Lipschitz continuous function $f : X \to \mathbb{R}$.

$$\min_{x\in X} f(x), \qquad (*)$$

♠ Recall that a *subgradient* $f'(x)$ of a convex function $f : X \to \mathbb{R}$ at a point $x \in X$ is the slope of a linear function which underestimates f everywhere on X and coincides with f at x:

$$f(y) \geq f(x) + \langle y - x, f'(x)\rangle \ \forall y \in X.$$

For Lipschitz continuous convex f, a norm $\|\cdot\|$ on E, and *every* $x \in X$ there exists a subgradient $f'(x)$ of f at x satisfying the norm bound

$$\|f'(x)\|_* \leq L_{\|\cdot\|}(f) \qquad (!)$$
$$\left[\|z\|_* = \max_{u:\|u\|\leq 1}\langle z, u\rangle\right]$$

When $x \in \operatorname{int} X$, the above relation holds true for *every* subgradient of f at x.

Convention: *In the sequel, when speaking about First Order oracles for Lipschitz continuous convex functions f, we always assume that the subgradients $f'(x)$ reported by the oracles satisfy* (!).

The simplest of the cheapest – Subgradient Descent (N. Shor, 1967)

♣ The *Subgradient Descent* method (SD) for solving a convex program

$$\min_{x \in X} f(x) \qquad (P)$$

- X – convex compact set in $\mathbb{R}^n$
- f – Lipschitz continuous on X convex function

is the recurrence

$$x_{t+1} = \Pi_X(x_t - \gamma_t f'(x_t)) \quad [x_1 \in X] \qquad \text{(SD)}$$

where

- $\gamma_t > 0$ are *stepsizes*
- $\Pi_X(x) = \text{argmin}_{y \in X} \|x - y\|_2^2$ is the standard *projector* on X,
- $f'(x)$ is a *subgradient* of f at x:

$$f(y) \geq f(x) + (y - x)^T f'(x) \quad \forall y \in X.$$

When, why and how SD converges?

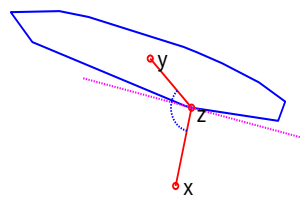
$$x_{t+1} = \Pi_X(x_t - \gamma_t f'(x_t)) \qquad \text{(SD)}$$

♣ We start with a simple geometric fact:

(!) *Let $X \subset \mathbb{R}^n$ be a closed convex set, $x \in \mathbb{R}^n$, and $z = \Pi_X(x)$. Then the vector $e = x - z$ forms an obtuse angle with every vector of the form $y - z$, $y \in X$:*

$$(x - z)^T(y - z) \leq 0 \quad \forall y \in X.$$

In particular, $y \in X \Rightarrow \|y - \Pi_X(x)\|_2^2 \leq \|y - x\|_2^2 - \|x - \Pi_X(x)\|_2^2$



In words: *When projecting a point x onto a closed convex set X, the squared $\|\cdot\|_2$ distance to any point from X is decreased by at least the squared $\|\cdot\|_2$-distance from the point x to its projection onto X.*

Indeed, when $y \in X$ and $0 \leq t \leq 1$, one has

$$\phi(t) = \|\underbrace{[\Pi_X(x) + t(y - \Pi_X(x))]}_{y_t \in X} - x\|_2^2 \geq \|\Pi_X(x) - x\|_2^2 = \phi(0),$$

whence $0 \leq \phi'(0) = 2(\Pi_X(x) - x)^T(y - \Pi_X(x))$. Consequently,

$$\|y - x\|_2^2 = \|y - \Pi_X(x)\|_2^2 + \|\Pi_X(x) - x\|_2^2 + 2(y - \Pi_X(x))^T(\Pi_X(x) - x) \geq \|y - \Pi_X(x)\|_2^2 + \|\Pi_X(x) - x\|_2^2.$$

$$x_{t+1} = \Pi_X(x_t - \gamma_t f'(x_t)) \qquad \text{(SD)}$$

♠ By Simple Geometric Fact, for every $u \in X$ one has

$$\begin{array}{rcl} \|x_{t+1} - u\|_2^2 & = & \|\Pi_X(x_t - \gamma_t f'(x_t)) - u\|_2^2 \\ & \leq & \|x_t - \gamma_t f'(x_t) - u\|_2^2 = \|x_t - u\|_2^2 - 2\gamma_t (x_t - u)^T f'(x_t) + \gamma_t^2 \|f'(x_t)\|_2^2 \end{array}$$

and we arrive at

Corollary: *For every $u \in X$ one has*

$$\gamma_t (x_t - u)^T f'(x_t) \leq \underbrace{\frac{1}{2}\|x_t - u\|_2^2}_{d_t} - \underbrace{\frac{1}{2}\|x_{t+1} - u\|_2^2}_{d_{t+1}} + \tfrac{1}{2}\gamma_t^2 \|f'(x_t)\|_2^2$$

Note: Since f is convex, one has $(x_t - u)^T f'(x_t) \geq f(x_t) - f(u)$, which combines with Corollary to yield

$$\boxed{\gamma_t [f(x_t) - f(u)] \leq \underbrace{\frac{1}{2}\|x_t - u\|_2^2}_{d_t} - \underbrace{\frac{1}{2}\|x_{t+1} - u\|_2^2}_{d_{t+1}} + \tfrac{1}{2}\gamma_t^2 \|f'(x_t)\|_2^2}$$

$f_* = \min_{x\in X} f(x)$	(1)
$x_{t+1} = \Pi_X(x_t - \gamma_t f'(x_t))$	(2)
$\gamma_t[f(x_t) - f(u)] \leq \underbrace{\frac{1}{2}\|x_t - u\|_2^2}_{d_t} - \underbrace{\frac{1}{2}\|x_{t+1} - u\|_2^2}_{d_{t+1}} + \frac{1}{2}\gamma_t^2\|f'(x_t)\|_2^2 \ \forall u \in X$	(3)

Summing up inequalities (3) over $t = T_0, T_0 + 1, ..., T$, we get

$$\sum_{t=T_0}^{T} \gamma_t(f(x_t) - f(u)) \leq \underbrace{d_{T_0} - d_{T+1}}_{\leq\Theta} + \sum_{t=T_0}^{T} \frac{1}{2}\gamma_t^2\|f'(x_t)\|_2^2$$

$$\left[\Theta = \max_{x,y\in X} \frac{1}{2}\|x - y\|_2^2\right]$$

Setting $u = x_* \equiv \mathrm{argmin}_X f$, we arrive at the bound

$$\boxed{\forall(T, T_0, T \geq T_0 \geq 1) : \epsilon_T \equiv \min_{t\leq T} f(x_t) - f_* \leq \frac{\Theta + \frac{1}{2}\sum_{t=T_0}^{T} \gamma_t^2\|f'(x_t)\|_2^2}{\sum_{t=T_0}^{T} \gamma_t}}$$

$$\forall(T, T_0, T \geq T_0 \geq 1) : \epsilon_T \equiv \min_{t \leq T} f(x_t) - f_* \leq \frac{\Theta + \frac{1}{2}\sum_{t=T_0}^{T} \gamma_t^2 \|f'(x_t)\|_2^2}{\sum_{t=T_0}^{T} \gamma_t}$$

♣ The resulting relation leads to various convergence results.

Example 1: "Divergent Series". *Let $\gamma_t \to 0$ as $t \to \infty$, while $\sum_t \gamma_t = \infty$. Then*

$$\lim_{T \to \infty} \epsilon_T = 0.$$

Proof. Set $T_0 = 1$ and note that

$$\frac{\sum_{t=1}^{T} \gamma_t^2 \|f'(x_t)\|_2^2}{\sum_{t=1}^{T} \gamma_t} \leq L^2_{\|\cdot\|_2}(f) \frac{\sum_{t=1}^{T} \gamma_t^2}{\sum_{t=1}^{T} \gamma_t} \to 0, \; T \to \infty.$$

$$\boxed{f_* = \min_{x\in X} f(x)}$$

$$\Downarrow$$

$$\boxed{\begin{array}{c}\forall (T, T_0, T \geq T_0 \geq 1) : \epsilon_T \equiv \min_{t\leq T} f(x_t) - f_* \leq \frac{\Theta + \frac{1}{2}\sum_{t=T_0}^{T} \gamma_t^2 \|f'(x_t)\|_2^2}{\sum_{t=T_0}^{T} \gamma_t} \\ \left[\Theta = \frac{1}{2}\max_{x,y\in X} \|x-y\|_2^2\right]\end{array}}$$

Example 2: "Optimal stepsizes":

$$\gamma_t = \frac{\sqrt{2\Theta}}{\|f'(x_t)\|_2\sqrt{t}} \Rightarrow \boxed{\epsilon_T \equiv \min_{t\leq T} f(x_t) - f_* \leq O(1)\frac{L_{\|\cdot\|_2}(f)\sqrt{\Theta}}{\sqrt{T}},\ T \geq 1}$$

Proof. Setting $T_0 = \lfloor T/2 \rfloor$, we get

$$\begin{array}{rcl}\epsilon_T & \leq & \left[\Theta + \Theta\sum_{t=T_0}^{T} t^{-1}\right]\left[\sum_{t=T_0}^{T} \frac{\sqrt{2\Theta}}{\sqrt{t}\|f'(x_t)\|_2}\right]^{-1} \leq \left[\Theta + \Theta\sum_{t=T_0}^{T} t^{-1}\right]\left[\sum_{t=T_0}^{T} \frac{\sqrt{2\Theta}}{\sqrt{t}L_{\|\cdot\|_2}(f)}\right]^{-1} \\ & \leq & L_{\|\cdot\|_2}(f)\sqrt{\Theta}\frac{1+O(1)}{O(1)\sqrt{T}} = O(1)\frac{L_{\|\cdot\|_2}(f)\sqrt{\Theta}}{\sqrt{T}}\end{array}$$

[note that with $T_0 = \lfloor T/2 \rfloor$ we have $\sum_{T_0}^{T} t^{-1} = O(1)$ and $\sum_{T_0}^{T} \frac{1}{\sqrt{t}} = O(1)\sqrt{T}$].

$$\boxed{\begin{array}{c} f_* = \min_{x\in X} f(x) \\ \Rightarrow x_{t+1} = \Pi_X(x_t - \gamma_t f'(x(t))), \quad \gamma_t = \frac{\max_{x,y\in X}\|x-y\|_2}{\sqrt{t}\|f'(x_t)\|_2} \\ \Rightarrow \epsilon_T \equiv \min_{1\le t\le T} f(x_t) - f_* \le O(1)\overbrace{L_{\|\cdot\|_2}(f)\max\limits_{x,y\in X}\|x-y\|_2}^{\mathrm{Var}_{\|\cdot\|_2,X}(f)}/\sqrt{T} \end{array}}$$

Good news: We have arrived at efficiency estimate which is *dimension-independent*, provided that the "$\|\cdot\|_2$-variation" of the objective on the feasible domain

$$\mathrm{Var}_{\|\cdot\|_2,X}(f) = L_{\|\cdot\|_2}(f)\max_{x,y\in X}\|x-y\|_2$$

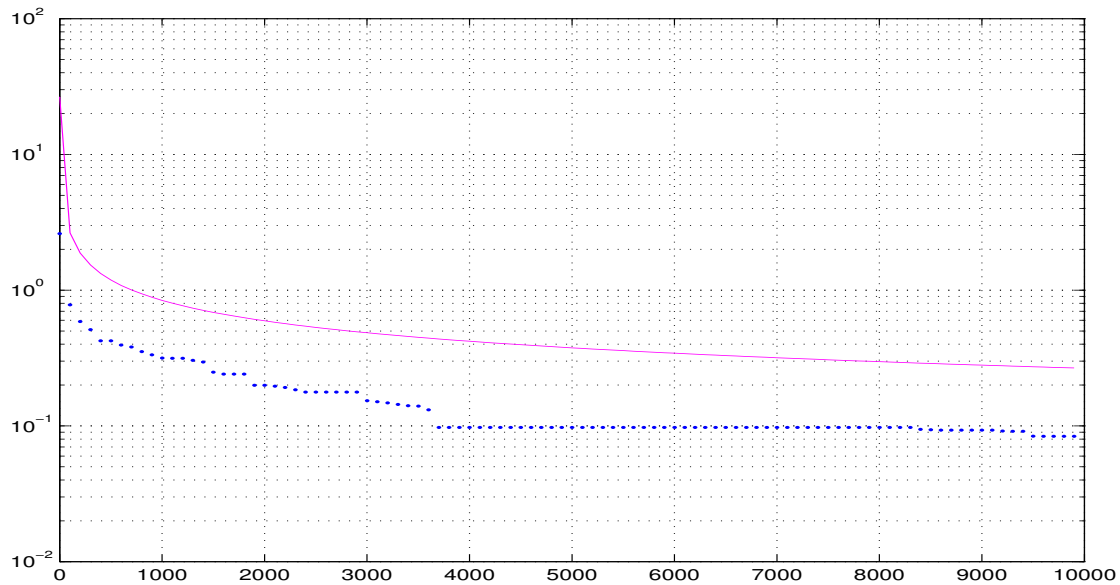
is fixed. Moreover, when X is a Euclidean ball in $\mathbb{R}^n$, this efficiency estimate "is as good as an efficiency estimate of a black-box-oriented method can be", provided that the dimension is large:

$$n \ge \left(\mathrm{Var}_{\|\cdot\|_2,X}(f)/\epsilon\right)^2$$

$$\epsilon_T \equiv \min_{1\le t\le T} f(x_t) - f_* \le O(1)\mathrm{Var}_{\|\cdot\|_2,X}(f)/\sqrt{T}$$
$$\left[\mathrm{Var}_{\|\cdot\|_2,X}(f) = L_{\|\cdot\|_2}(f)\max_{x,y\in X}\|x-y\|_2\right]$$

Bad news: Our "dimension-independent" efficiency estimate

- is pretty slow
- is indeed dimension-independent only for problems with "Euclidean geometry" – those with moderate $\|\cdot\|_2$-variation. As a matter of fact, in some (but not all!) important applications problems of this type are pretty rare.



SD as applied to $\min_{\|x\|_2\le 1}\|Ax-b\|_1$, $A: 50\times 50$
[red: efficiency estimate; blue: actual error]

$$x_{t+1} = \Pi_X(x_t - \gamma_t f'(x(t)))$$

♣ An evident drawback of SD is that all information on the objective accumulated so far is "summarized" in the current iterate, and this "summary" is very incomplete. With better usage of past information, one arrives at *bundle methods* which outperform SD significantly in practice, while preserving the most attractive theoretical property of SD – dimension-independent and optimal, in favourable circumstances, rate of convergence.

Bundle-Level method for solving $f_* = \min_{x\in X} f(x)$

♣ At the beginning of step t of BL, we have at our disposal
— the first-order information $\{f(x_\tau), f'(x_\tau)\}_{1\le\tau<t}$ on f along the previous search points $x_\tau \in X$, $\tau < t$;
— current iterate $x_t \in X$.
♣ At step t we
— compute $f(x_t), f'(x_t)$; this information, along with the past first-order information on f, provides is with the current *model of the objective*

$$f_t(x) = \max_{\tau\le t}[f(x_\tau) + (x - x_\tau)^T f'(x_\tau)]$$

This model underestimates the objective and is exact at the points $x_1, ..., x_t$;
— define the *best found so far value* $f^t = \min_{\tau\le t} f(x_\tau)$ of f
— define the current *lower bound* f_t on f_* by solving the auxiliary problem

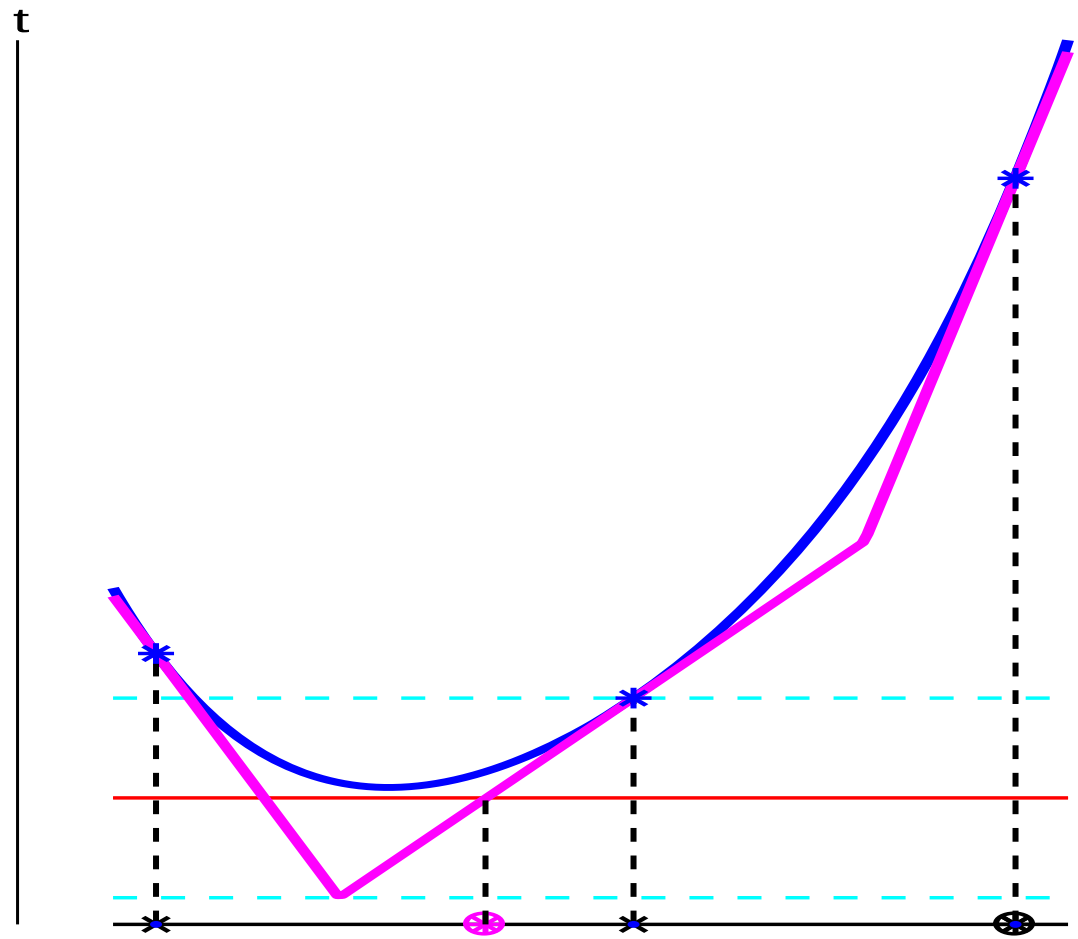
$$f_t = \min_{x\in X} f_t(x) \qquad (\mathrm{LP}_t)$$

Note: current *gap* $\Delta_t = f^t - f_t$ upper-bounds the inaccuracy of the best found so far solution;

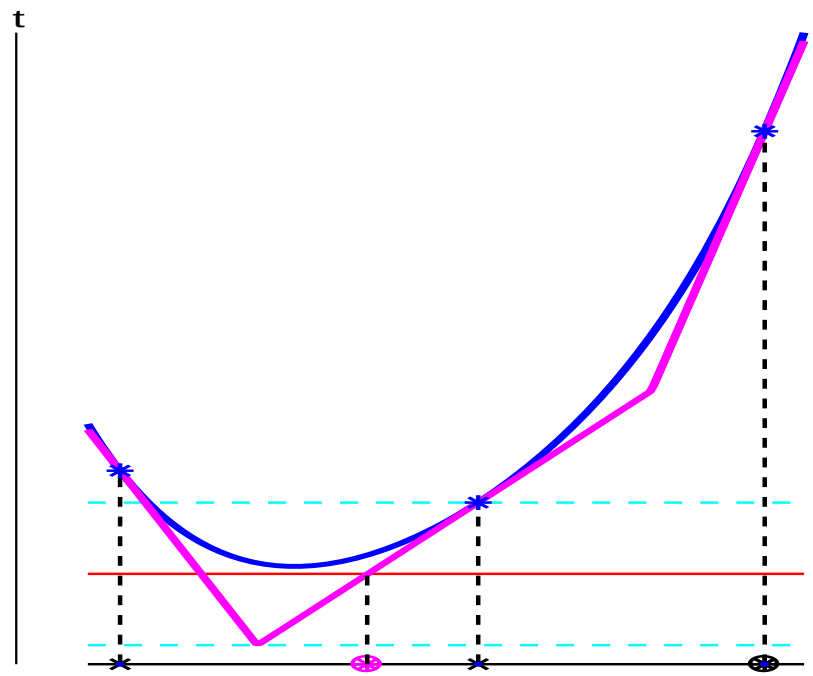
- compute the current *level* $\ell_t = f_t + \lambda\Delta_t$ ($\lambda \in (0,1)$ is a parameter)
- build a new search point by solving the auxiliary problem

$$x_{t+1} = \operatorname*{argmin}_x\{\|x - x_t\|_2^2 : x \in X, f_t(x) \le \ell_t\} \qquad (\mathrm{QP}_t)$$

and loop to step $t + 1$.



- blue: the objective f
- $*$: x_1, x_2, x_3
- magenta: current piecewise linear model $f_3(\cdot)$ of f
- cyan horizontal lines: $t = \min_{i \le 3} f(x_i)$ and $t = \min_x f_3(x)$
- red horizontal line: $t = \ell_3$
- red circle: new iterate x_4



Note: It seems to be more intuitive to "fully trust" in model and take, as the next iterate, the minimizer of the model or, which is the same, to set the level ℓ_t equal to f_t rather than to

$$\ell_t = f_t + \lambda \Delta_t \quad \Delta_t = \min_{\tau \le t} f(x_\tau) - f_t. \qquad [\lambda \in (0,1), \text{ usually } \lambda = 0.5]$$

Unfortunately, the resulting *Kelley method* has disastrously bad theoretical complexity (and from time to time exhibits disastrously bad actual performance).

How BL converges?

Claim: *For every* ϵ, $0 < \epsilon < \Delta_1$, *the number* N *of steps before a gap* $\leq \epsilon$ *is obtained* (i.e., before an ϵ-solution is found) *does not exceed the bound*

$$N(\epsilon) = \frac{\mathrm{Var}^2_{\|\cdot\|_2, X}(f)}{\lambda(1-\lambda)^2(2-\lambda)\epsilon^2},$$

$\Rightarrow$ *Inaccuracy after* $T = 1, 2, \ldots$ *steps is upper-bounded by*

$$C(\lambda)\frac{\mathrm{Var}_{\|\cdot\|_2, X}(f)}{\sqrt{T}}$$

— the same efficiency estimate as for SD with optimal stepsizes.

♣ We have seen that Bundle-Level shares the dimension-independent (and optimal in the "favourable" large-scale case) theoretical complexity bound

For every $\epsilon > 0$, the number of steps before an ϵ-solution to convex program $\min_{x\in X} f(x)$ is found, does not exceed

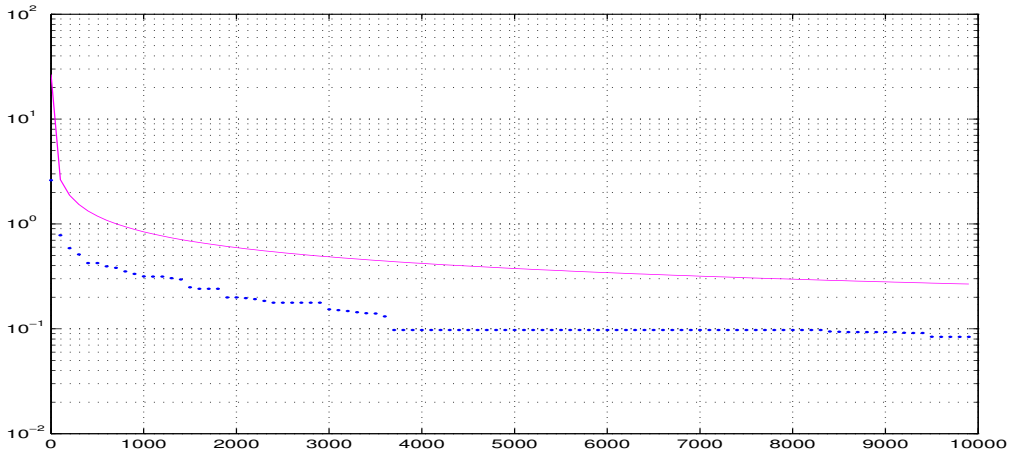
$$O(1)\left(\frac{\mathrm{Var}_{\|\cdot\|_2,X}(f)}{\epsilon}\right)^2.$$

♣ There exists quite convincing *experimental* evidence that Bundle-Level obeys the optimal in fixed dimension "polynomial time" complexity bound:

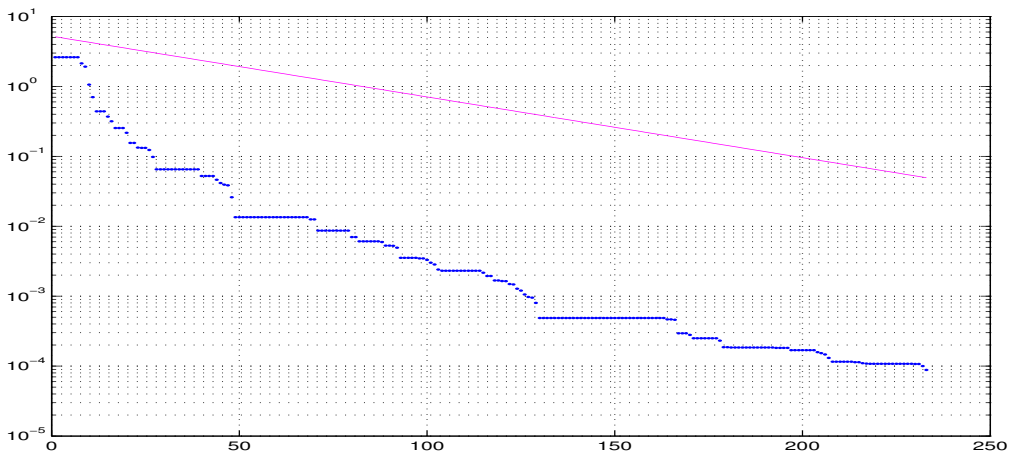
For every $\epsilon \in (0, \mathrm{Var}_X(f) \equiv \max_X f - \min_X f)$, the number of steps before an ϵ-solution to convex program $\min_{x\in X} f(x)$ with $X \subset \mathbb{R}^n$ is found, does not exceed $n \ln\left(\frac{\mathrm{Var}_X(f)}{\epsilon}\right) + 1$.

♠ **Experimental rule:** *When solving convex program with n variables by BL, every n steps add new accuracy digit.*

Illustration: $\min_{x:\|x\|_2\le 1} f(x) \equiv \|Ax-b\|_1$, $\dim x = 50$ $(f(0) = 2.61,\ f_* = 0)$



SD, accuracy vs. iteration count. blue: errors; red: efficiency estimate $3\frac{\mathrm{Var}_{\|\cdot\|_2,X}(f)}{\sqrt{t}}$; $\epsilon_{10000} = 0.084$



BL, accuracy vs. iteration count. blue: errors; red: efficiency estimate $e^{-t/n}\mathrm{Var}_X(f)$; $\epsilon_{233} < 1.e-4$

♣ In BL, the number of linear constraints in the auxiliary problems

$$f_t = \min_{x\in X} f_t(x) \qquad (\mathrm{LP}_t)$$
$$x_{t+1} = \mathrm{argmin}_x \left\{\|x_t - x\|_2^2 : x \in X, f_t(x) \le \ell_t\right\} \qquad (\mathrm{QP}_t)$$

is equal to the size t of the current *bundle* – the collection of affine forms $g_\tau(x) = f(x_\tau) + (x - x_\tau)^T f'(x_\tau)$ participating in the model $f_t(\cdot)$. Thus, the complexity of an iteration in BL grows with the iteration number. In order to suppress this phenomenon, one needs a mechanism for *shrinking* the bundle (and thus – simplifying the models of f).

♠ The simplest way of shrinking the bundle is to initialize d as Δ_1 and to run plain BL until an iteration t with $\Delta_t \le d/2$ is met. At such an iteration, we

— shrink the current bundle, keeping in it the minimum number of the forms g_τ sufficient to ensure that

$$f_t \equiv \min_{x\in X} \max_{1\le\tau\le t} g_\tau(x) = \min_{x\in X} \max_{\text{selected } \tau} g_\tau(x)$$

(this number is at most n),

— reset d as Δ_t,

and proceed with plain BL until the gap is again reduced by factor 2, etc.

♣ Computational experience demonstrates that the outlined approach does not slow BL down, while keeping the size of the bundle below the level of about $2n$.

Stochastic Subgradient Descent (Stochastic Approximation)

♣ Consider the case when solving a convex program

$$f_* = \min_{x \in X} f(x)$$

[• $X \subset \mathbb{R}^n$: convex compact • $f : X \to \mathbb{R}$ convex and Lipschitz]

no precise first order information is available. Specifically, we have at our disposal

• *Stochastic Oracle* (SO) for f as follows: at t-th call to the oracle, x_t being the input, the oracle returns

$$g(x_t, \xi_t) \in \mathbb{R},\ G(x_t, \xi_t) \in \mathbb{R}^n$$

as random estimates of $f(x_t)$ and $f'(x_t)$, where $\xi_1, \xi_2, ...$ is a sequence of independent realizations of a *random variable* ξ ("oracle's noise").

♠ We assume that the SO is *unbiased*:

$$\mathbf{E}\{g(x, \xi)\} = f(x), \quad f'(x) := \mathbf{E}\{G(x, \xi)\} \in \partial f(x).$$

In addition, we assume that

$$\mathbf{E}\{\|G(x, \xi)\|_2^2\} \leq L^2 < \infty \ \forall x \in X$$

Example: Our f is given as expectation:

$$f(x) = \int_{\Xi} F(x, \xi) dP(\xi),$$

where F is convex in x and efficiently computable.
When we cannot compute the expectation in a closed analytic form, but can instead sample from the distribution P, we, under mild regularity assumptions on F, have at our disposal unbiased Stochastic Oracle

$$g(x, \xi) = F(x, \xi), \quad G(x, \xi) = F'_x(x, \xi)$$

$f_* = \min\limits_{x\in X} f(x)$ $\mathbf{E}\{g(x,\xi)\} = f(x), f'(x) := \mathbf{E}\{G(x,\xi)\} \in \partial f(x), \mathbf{E}\{\|G(x,\xi)\|_2^2\} \leq L^2 < \infty \ \forall x \in X$
$\Pi_X(z) = \mathrm{argmin}_{u\in X} \|z-u\|_2^2$ $\left[\forall u \in X, x \in \mathbb{R}^n : \|\Pi_X(x) - u\|_2^2 \leq \|x-u\|_2^2 - \|x - \Pi_X(x)\|_2^2\right]$

♣ We can solve the problem with *Stochastic Subgradient Descent* (a.k.a. *Stochastic Approximation*) which is completely similar to deterministic Subgradient Descent:

$$x_1 \in X; x_{t+1} = \Pi_X\left(x_t - \gamma_t G(x_t, \xi_t)\right), 1 \leq t \leq T;$$
$$x_{T_0}^T = \frac{1}{\sum_{t=T_0}^T \gamma_t} \sum_{t=T_0}^T \gamma_t x_t.$$

Here $\gamma_t > 0$ are deterministic stepsizes, and (deterministic) total number of steps T and T_0 are such that $1 \leq T_0 \leq T$.

$$x_1 \in X; x_{t+1} = \Pi_X(x_t - \gamma_t G(x_t, \xi_t)),\ 1 \le t \le N;$$
$$x_{T_0}^T = \frac{1}{\sum_{t=T_0}^T \gamma_t} \sum_{t=T_0}^T \gamma_t x_t.$$

Fact: *For Stochastic Subgradient Descent one has*

$$\mathbf{E}\{f(x_{T_0}^T) - f(x_*)\} \le \frac{\Theta + \frac{1}{2}\sum_{t=T_0}^T \gamma_t^2 L^2}{\sum_{t=T_0}^T \gamma_t},$$
$$\Theta = \max_{x,y \in X} \frac{1}{2}\|x - y\|_2^2$$

that is, we get exactly the same efficiency estimate as in the case of precise First Order oracle, but now – for the expected inaccuracy of the approximate solutions $x_{T_0}^T$ – the weighted sums of the search points we have generated in course of $T = 1, 2, ...$ steps.

$$x_1 \in X; x_{t+1} = \Pi_X(x_t - \gamma_t G(x_t, \xi_t)),\ 1 \le t \le T; x_{T_0}^T = \frac{1}{\sum_{t=T_0}^T \gamma_t} \sum_{t=T_0}^T \gamma_t x_t.$$

Convergence Analysis of Stochastic Subgradient Descent

♠ Let us carry out convergence analysis of the algorithm. Denoting by x_* a minimizer of f over X, we, as always, have

$$\begin{array}{rl} & \gamma_t \langle G(x_t, \xi_t), x_t - x_* \rangle \le \frac{1}{2}\|x_t - x_*\|_2^2 - \frac{1}{2}\|x_{t+1} - x_*\|_2^2 + \frac{1}{2}\gamma_t^2 \|G(x_t, \xi_t)\|_2^2 \\ \Rightarrow & \sum_{t=T_0}^T \gamma_t \langle G(x_t, \xi_t), x_t - x_* \rangle \le \Theta + \frac{1}{2}\sum_{t=T_0}^T \gamma_t^2 \|G(x_t, \xi_t)\|_2^2 \qquad (*) \end{array}$$

Taking expectations of both sides in $(*)$ and taking into account that x_t is a deterministic function of $\xi_1, ..., \xi_{t-1}$ and the conditional, $\xi_1, ..., \xi_{t-1}$ given, expectation of $G(x_t, \xi_t)$ is $f'(x_t)$ (since $\xi_1, \xi_2, ...$ are i.i.d.), we get

$$\sum_{t=T_0}^T \gamma_t \mathbf{E}\{\langle f'(x_t), x_t - x_* \rangle\} \le \Theta + \frac{1}{2}\sum_{t=T_0}^T \gamma_t^2 L^2,$$

whence also

$$\mathbf{E}\{\sum_{t=T_0}^T \gamma_t [f(x_t) - f(x_*)]\} \le \Theta + \frac{1}{2}\sum_{t=T_0}^T \gamma_t^2 L^2$$

Setting $\lambda_t = \gamma_t / \sum_{s=T_0}^T \gamma_s$, $T_0 \le t \le T$, we get

$$\mathbf{E}\left\{\sum_{t=T_0}^T \lambda_t f(x_t)\right\} - f(x_*) = \mathbf{E}\left\{\sum_{t=T_0}^T \lambda_t [f(x_t) - f(x_*)]\right\} \le \frac{\Theta + \frac{1}{2}\sum_{t=T_0}^T \gamma_t^2 L^2}{\sum_{t=T_0}^T \gamma_t}$$

$$\mathbf{E}\left\{\sum_{t=T_0}^T \lambda_t f(x_t)\right\} - f(x_*) = \mathbf{E}\left\{\sum_{t=T_0}^T \lambda_t [f(x_t) - f(x_*)]\right\} \leq \frac{\Theta + \frac{1}{2}\sum_{t=T_0}^T \gamma_t^2 L^2}{\sum_{t=T_0}^T \gamma_t}$$

$$x_{T_0}^T = \frac{1}{\sum_{t=T_0}^T \gamma_t} \sum_{t=T_0}^T \gamma_t x_t = \sum_{t=T^0}^T \lambda_t x_t$$

$$\left[\lambda_t = \gamma_t / \sum_{s=T_0}^T \gamma_s\right]$$

By convexity, $f(x_{T_0}^T) \leq \sum_{t=T_0}^T \lambda_t f(x_t)$, whence

$$\mathbf{E}\{f(x_{T_0}^T) - f(x_*)\} = \mathbf{E}\{f(x_{T_0}^T)\} - f(x_*) \quad \leq \quad \mathbf{E}\left\{\sum_{t=T_0}^T \lambda_t f(x_t)\right\} - f(x_*) \leq \frac{\Theta + \frac{1}{2}\sum_{t=T_0}^T \gamma_t^2 L^2}{\sum_{t=T_0}^T \gamma_t},$$

as claimed. $\square$

Stochastic Subgradient Descent (Stochastic Approximation)

♣ Consider the case when solving a convex program

$$f_* = \min_{x \in X} f(x)$$

[• $X \subset \mathbb{R}^n$: convex compact • $f : X \to \mathbb{R}$ convex and Lipschitz]

no precise first order information is available. Specifically, we have at our disposal

• *Stochastic Oracle* (SO) for f as follows: at t-th call to the oracle, x_t being the input, the oracle returns

$$g(x_t, \xi_t) \in \mathbb{R},\ G(x_t, \xi_t) \in \mathbb{R}^n$$

as random estimates of $f(x_t)$ and $f'(x_t)$, where $\xi_1, \xi_2, ...$ is a sequence of independent realizations of a *random variable* ξ ("oracle's noise").

♠ We assume that the SO is *unbiased*:

$$\mathbf{E}\{g(x, \xi)\} = f(x), \quad \mathbf{E}\{G(x, \xi)\} \in \partial f(x).$$

In addition, we assume that

$$\mathbf{E}\{\|G(x, \xi)\|_2^2\} \leq L^2 < \infty \ \forall x \in X$$

Example: Our f is given as expectation:

$$f(x) = \int_{\Xi} F(x, \xi) dP(\xi),$$

where F is convex in x and efficiently computable.
When we cannot compute the expectation in a closed analytic form, but can instead sample from the distribution P, we, under mild regularity assumptions on F, have at our disposal unbiased Stochastic Oracle

$$g(x, \xi) = F(x, \xi), \quad G(x, \xi) = F'_x(x, \xi)$$

$f_* = \min\limits_{x\in X} f(x)$ $\mathbf{E}\{g(x,\xi)\} = f(x), \mathbf{E}\{G(x,\xi)\} \in \partial f(x), \mathbf{E}\{\|G(x,\xi)\|_2^2\} \leq L^2 < \infty \ \forall x \in X$
$\Pi_x(\xi) = \mathrm{argmin}_{u\in X} \|\xi - u\|_2^2$ $\left[\forall u \in X, x \in \mathbb{R}^n : \|\Pi_X(x) - u\|_2^2 \leq \|x - u\|_2^2 - \|x - \Pi_X(x)\|_2^2\right]$

♣ We can solve the problem with *Stochastic Subgradient Descent* (a.k.a. *Stochastic Approximation*) which is completely similar to deterministic Subgradient Descent:

$$x_1 \in X; x_{t+1} = \Pi_X\left(x_t - \gamma_t G(x_t, \xi_t)\right), 1 \leq t \leq T;$$
$$x_{T_0}^T = \frac{1}{\sum_{t=T_0}^T \gamma_t} \sum_{t=T_0}^T \gamma_t x_t.$$

Here $\gamma_t > 0$ are deterministic stepsizes, and (deterministic) total number of steps T and T_0 are such that $1 \leq T_0 \leq T$.

$$x_1 \in X; x_{t+1} = \Pi_X(x_t - \gamma_t G(x_t, \xi_t)),\ 1 \le t \le N;$$
$$x_{T_0}^T = \frac{1}{\sum_{t=T_0}^T \gamma_t} \sum_{t=T_0}^T \gamma_t x_t.$$

Fact: *For Stochastic Subgradient Descent one has*

$$\mathbf{E}\{f(x_{T_0}^T) - f(x_*)\} \le [\sum_{t=T_0}^T \gamma_t]^{-1} \mathbf{E}\{\sum_{t=T_0}^T \gamma_t [f(x_t) - f_*]\} \le \frac{\Theta + \frac{1}{2}\sum_{t=T_0}^T \gamma_t^2 L^2}{\sum_{t=T_0}^T \gamma_t},$$
$$\Theta = \max_{x,y \in X} \frac{1}{2}\|x - y\|_2^2$$

that is, we get exactly the same efficiency estimate as in the case of precise First Order oracle, but now – for the expected inaccuracy of the approximate solutions $x_{T_0}^T$ – the weighted sums of the search points we have generated in course of $T = 1, 2, ...$ steps.

$$x_1 \in X; x_{t+1} = \Pi_X(x_t - \gamma_t G(x_t, \xi_t)),\ 1 \le t \le T; x_{T_0}^T = \frac{1}{\sum_{t=T_0}^T \gamma_t} \sum_{t=T_0}^T \gamma_t x_t.$$

Convergence Analysis of Stochastic Subgradient Descent

♠ Let us carry out convergence analysis of the algorithm. Denoting by x_* a minimizer of f over X, we, as always, have

$$\begin{array}{rl} & \gamma_t \langle G(x_t, \xi_t), x_t - x_* \rangle \le \frac{1}{2}\|x_t - x_*\|_2^2 - \frac{1}{2}\|x_{t+1} - x_*\|_2^2 + \frac{1}{2}\gamma_t^2 \|G(x_t, \xi_t)\|_2^2 \\ \Rightarrow & \sum_{t=T_0}^T \gamma_t \langle G(x_t, \xi_t), x_t - x_* \rangle \le \Theta + \frac{1}{2}\sum_{t=T_0}^T \gamma_t^2 \|G(x_t, \xi_t)\|_2^2 \qquad (*) \end{array}$$

Taking expectations of both sides in $(*)$ and taking into account that x_t is a deterministic function of $\xi_1, ..., \xi_{t-1}$, while $\xi_1, ..., \xi_T$ are independent and the conditional, $\xi_1, ..., \xi_{t-1}$ given, expectation of $G(x_t, \xi_t)$ is $f'(x_t)$, we get

$$\sum_{t=T_0}^T \gamma_t \mathbf{E}\{\langle f'(x_t), x_t - x_* \rangle\} \le \Theta + \frac{1}{2}\sum_{t=T_0}^T \gamma_t^2 L^2,$$

whence also

$$\mathbf{E}\{\sum_{t=T_0}^T \gamma_t [f(x_t) - f(x_*)]\} \le \Theta + \frac{1}{2}\sum_{t=T_0}^T \gamma_t^2 L^2$$

Setting $\lambda_t = \gamma_t / \sum_{s=T_0}^T \gamma_s$, $T_0 \le t \le T$, we get

$$\mathbf{E}\left\{\sum_{t=T_0}^T \lambda_t f(x_t)\right\} - f(x_*) = \mathbf{E}\left\{\sum_{t=T_0}^T \lambda_t [f(x_t) - f(x_*)]\right\} \le \frac{\Theta + \frac{1}{2}\sum_{t=t_0}^T \gamma_t^2 L^2}{\sum_{t=T_0}^T \gamma_t}$$

$$\mathbf{E}\left\{\sum_{t=T_0}^T \lambda_t f(x_t)\right\} - f(x_*) = \mathbf{E}\left\{\sum_{t=T_0}^T \lambda_t [f(x_t) - f(x_*)]\right\} \leq \frac{\Theta + \frac{1}{2}\sum_{t=t_0}^T \gamma_t^2 L^2}{\sum_{t=T_0}^T \gamma_t}$$

$$x_{T_0}^T = \frac{1}{\sum_{t=T_0}^T \gamma_t} \sum_{t=T_0}^T \gamma_t x_t = \sum_{t=T^0}^T \lambda_t x_t$$

$$\left[\lambda_t = \gamma_t / \sum_{s=T_0}^T \gamma_s\right]$$

By convexity, $f(x_{T_0}^T) \leq \sum_{t=T_0}^T \lambda_t f(x_t)$, whence

$$\mathbf{E}\{f(x_{T_0}^T) - f(x_*)\} = \mathbf{E}\{f(x_{T_0}^T)\} - f(x_*) \quad \leq \quad \mathbf{E}\left\{\sum_{t=T_0}^T \lambda_t f(x_t)\right\} - f(x_*) \leq \frac{\Theta + \frac{1}{2}\sum_{t=T_0}^T \gamma_t^2 L^2}{\sum_{t=T_0}^T \gamma_t},$$

as claimed. $\square$

$$f_* = \min_{x \in X} f(x) \qquad (*)$$

From Gradient to Mirror Descent

♣ Subgradient Descent method and its bundle versions are "intrinsically adjusted" to problems with Euclidean geometry; this is where the role of the $\|\cdot\|_2$-variation of the objective

$$\mathrm{Var}_{\|\cdot\|_2, X}(f) = L_{\|\cdot\|_2}(f) \max_{x, x' \in X} \|x - x'\|_2$$

in the efficiency estimate

$$\min_{t \le T} f(x_t) - f_* \le O(1) \frac{\mathrm{Var}_{\|\cdot\|_2, X}(f)}{\sqrt{T}}$$

comes from.

♣ An extension of SD and its bundle versions onto problems with "nice non-Euclidean geometry" is offered by the *Mirror Descent* scheme.

Mirror Descent – Building Blocks

♣ **Building block #1: Distance-Generating Function.**

♠ A SD step

$$x \mapsto x_+ = \Pi_X(x - \gamma f'(x)) \tag{1}$$

can be viewed as follows: given an iterate $x \in X$, we

1) Compute $f'(x)$

2) Perform the *prox-step* $x \mapsto x_+ = \mathrm{Prox}_x(\gamma f'(x))$

$$\underbrace{\mathrm{Prox}_x(\xi) := \operatorname*{argmin}_{u \in X} [\langle \xi, u \rangle + V_x(u)]}_{\xi \mapsto \mathrm{Prox}_x(\xi)\text{: prox-mapping with prox-center } x}$$

$$V_x(u) = \omega(u) - \omega(x) - \langle u - x, \nabla\omega(x) \rangle$$

where

$$\omega(u) = \frac{1}{2}\|u\|_2^2 \tag{2}$$

is a specific "distance-generating function."

Indeed, with the above $\omega(\cdot)$, we have

$$V_x(u) := \tfrac{1}{2}u^T u - x^T(u - x) - \tfrac{1}{2}x^T x = \tfrac{1}{2}\|u - x\|_2^2$$

$$\Downarrow$$

$$\mathrm{Prox}_x(\xi) = \operatorname*{argmin}_{u \in X} \left[\xi^T u + \tfrac{1}{2}(u - x)^T(u - x)\right] = \operatorname*{argmin}_{u \in X} \tfrac{1}{2}\|u - (x - \xi)\|_2^2 = \Pi_x(x - \xi)$$

$$\boxed{\begin{array}{rcl}\mathrm{Prox}_x(\xi) & = & \mathrm{argmin}_{u\in X}\left[\langle\xi,u\rangle + V_x(u)\right]\\ V_x(u) & = & \omega(u)-\omega(x)-\langle\nabla\omega(x),u-x\rangle\end{array}}$$

♠ The "Main Inequality"

$$x_+ = \Pi_X(x-\gamma f'(x)) \Rightarrow \forall u\in X: \gamma\langle f'(x),x-u\rangle \le \tfrac{1}{2}\|x-u\|_2^2 - \tfrac{1}{2}\|x_+-u\|_2^2 + \tfrac{1}{2}\gamma^2\|f'(x)\|_2^2$$

underlying all our convergence and rate-of-convergence results is an immediate corollary of the following "Magic Inequality:"

(!) *With convex and continuously differentiable $\omega(\cdot): X\to\mathbb{R}$ for all $x\in X$, $\xi\in\mathbb{R}^n$ one has:*

$$x_+ = \mathrm{Prox}_x(\xi) \Rightarrow \forall u\in X: \langle\xi,x_+-u\rangle \le V_x(u) - V_{x_+}(u) - V_x(x_+)$$

where $V_x(u) = \omega(u) - [\omega(x) + \langle u-x,\nabla\omega(x)\rangle]$ is the generated by $\omega(\cdot)$ Bregman distance from u to x, $u,x\in X$.

as applied to $\omega(u)\equiv\frac{1}{2}u^Tu$.

• **Justifying Magic Inequality**:

$$x_+ = \text{argmin}_{u\in X}\,[\langle\xi,u\rangle + V_x(u)] \Rightarrow \forall u\in X : \langle \xi - \nabla\omega(x) + \nabla\omega(x_+), u - x_+\rangle \geq 0$$

[optimality conditions]

$$\begin{array}{rcl} \Leftrightarrow \forall u\in X : \langle \xi, x_+ - u\rangle & \leq & \langle \nabla\omega(x_+) - \nabla\omega(x), u - x_+\rangle \\ & = & [\omega(u) - \omega(x) - \langle\nabla\omega(x), u - x\rangle] \\ & & -[\omega(u) - \omega(x_+) - \langle\nabla\omega(x_+), u - x_+\rangle] \\ & & -[\omega(x_+) - \omega(x) - \langle\nabla\omega(x), x_+ - x] \\ & = & V_x(u) - V_{x_+}(u) - V_x(x_+) \end{array}$$

• **Magic Inequality $\Rightarrow$ Main Inequality:** As we know, with $\omega(u) = \frac{1}{2}\|u\|_2^2$ we have $\Pi_X(x-\xi) = \text{Prox}_x(\xi)$. Thus,

$$\begin{array}{l} x_+ = \Pi_X(x - \gamma f'(x)) \Rightarrow x_+ = \text{Prox}_x(\gamma f'(x)) \\ \Rightarrow \forall u\in X : \langle \gamma f'(x), x_+ - u\rangle \leq V_x(u) - V_{x_+}(u) - V_x(x_+) \\ \Rightarrow \forall u\in X : \langle \gamma f'(x), x - u\rangle \quad \leq \quad V_x(u) - V_{x_+}(u) + \underbrace{[\langle\gamma f'(x), x - x_+\rangle - V_x(x_+)]}_{\delta} \end{array}$$

With our $\omega(\cdot)$, $V_x(x_+) = \frac{1}{2}\|x - x_+\|_2^2$, whence

$$\delta = \langle \gamma f'(x), x - x_+\rangle - \frac{1}{2}\|x - x_+\|_2^2 \leq \frac{1}{2}\|\gamma f'(x)\|_2^2,$$

and we arrive at the Main Inequality.

Distance-Generating Functions

♣ Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. A function $\omega(\cdot) : X \to \mathbb{R}$ is called *Distance-Generating Function (DGF) for X compatible with $\|\cdot\|$*, if
— $\omega(\cdot) : X \to \mathbb{R}$ is convex and continuously differentiable
— $\omega(\cdot)$ is strongly convex, modulus 1, w.r.t. $\|\cdot\|$, that is,

$$\forall x, y \in X : \langle \nabla\omega(x) - \nabla\omega(y), x - y\rangle \geq \|y - x\|^2$$

or, equivalently,

$$\forall (x \in X, u \in X) : V_x(u) := \omega(u) - \omega(x) - \langle u - x, \nabla\omega(x), \rangle \geq \tfrac{1}{2}\|u - x\|^2.$$

Note: For every convex compact set $X \subset \mathbb{R}^n$, the function $\omega(u) = \frac{1}{2}\|u\|_2^2$ restricted to X is a DGF compatible with $\|\cdot\| = \|\cdot\|_2$. For this DGF, $V_x(y) = \frac{1}{2}\|y - x\|_2^2$.

$$\forall (x \in X, u \in X) : V_x(u) := \omega(u) - \omega(x) - \langle \nabla\omega(x), u - x\rangle \geq \tfrac{1}{2}\|u - x\|^2.$$

Fact: *Whenever $\omega(\cdot)$ is a DGF for X compatible with $\|\cdot\|$, for $x \in X$, $\xi \in \mathbb{R}^n$, the prox-mapping*

$$x_+ = \mathrm{Prox}_x(\xi) \quad := \quad \mathrm{argmin}_{u\in X}\left[\langle \xi, u\rangle + V_x(u)\right]$$

is well-defined, takes values in X, and ensures that

$$\forall (u \in X) : \langle \xi, x_+ - u\rangle \leq V_x(u) - V_{x_+}(u) - V_x(x_+), \tag{1}$$

whence also

$$\forall (u \in X) : \langle \xi, x - u\rangle \leq V_x(u) - V_{x_+}(u) + \frac{1}{2}\|\xi\|_*^2, \tag{2}$$

where $\|\cdot\|_*$ is the norm conjugate to $\|\cdot\|$:

$$\|\xi\|_* = \max_x \left\{\langle \xi, x\rangle : \|x\| \leq 1\right\}.$$

$$V_x(u) = \omega(u) - \omega(x) - \langle u - x, \nabla\omega(x)\rangle \geq \tfrac{1}{2}\|u - x\|^2$$
$$x_+ = \mathrm{Prox}_x(\xi) := \mathrm{argmin}_{u\in X}\left[\langle \xi, u\rangle + V_x(u)\right]$$

Claims:

$$\forall(u \in X) : \langle \xi, x_+ - u\rangle \leq V_x(u) - V_{x_+}(u) - V_x(x_+) \quad (1)$$
$$\forall(u \in X) : \langle \xi, x - u\rangle \leq V_x(u) - V_{x_+}(u) + \tfrac{1}{2}\|\xi\|_*^2 \quad (2)$$

Indeed, as we have seen, (1) follows from optimality conditions as applied to the problem defining x_+. To derive (2) from (1), we need to show that

$$\langle \xi, x - x_+\rangle - V_x(x_+) \leq \tfrac{1}{2}\|\xi\|_*^2,$$

which is immediate due to

$$\langle \xi, x - x_+\rangle \leq \|\xi\|_*\|x - x_+\| \quad \& \quad V_x(x_+) \geq \frac{1}{2}\|x - x_+\|^2.$$

♣ **Conclusion:** *Subgradient Descent step*

$$x \mapsto x_+ = \Pi_X(x - \gamma f'(x)) \tag{1}$$

is nothing but the prox-step

$$\begin{aligned} x \mapsto x_+ &= \operatorname*{argmin}_{y \in X} [\langle \gamma f'(x), y \rangle + V_x(y)] \\ V_x(y) &= \omega(y) - [\omega(x) + \langle y - x, \nabla \omega(x) \rangle] \end{aligned} \tag{$*$}$$

associated with the specific distance-generating function

$$\omega(u) = \frac{1}{2} u^T u \tag{2}$$

$$X \ni x \mapsto x_+ = \mathrm{argmin}_{y\in X}\left[\langle \xi, y\rangle + V_x(y)\right] \qquad (*)$$
$$\Rightarrow \forall (u \in X): \langle \xi, x-u\rangle \le V_x(u) - V_{x_+}(u) + \tfrac{1}{2}\|\xi\|_*^2 \qquad (2)$$
$$\left[\begin{array}{c} V_x(u) = \omega(u) - [\langle u-x, \nabla\omega(x)\rangle + \omega(x)] \\ \omega(z): X \to \mathbb{R}: \text{ continuously differentiable \& } \langle \nabla\omega(x) - \nabla\omega(y), x-y\rangle \ge \|x-y\|^2 \end{array}\right]$$

♣ **Building block #2: the potential.** Convergence analysis of SD was based on the ensured by SD step inequality

$$\forall u \in X: \gamma\langle f'(x), x-u\rangle \le \underbrace{\frac{1}{2}\|x-u\|_2^2 - \frac{1}{2}\|x_+-u\|_2^2}_{\substack{= [\frac{1}{2}x^Tx - x^Tu] - [\frac{1}{2}x_+^Tx_+ - x_+^Tu] \\ = V_x(u) - V_{x_+}(u)}} + \tfrac{1}{2}\|\gamma f'(x)\|_2^2 \qquad (3)$$

where V_x stems from $\omega(\cdot) = \frac{1}{2}\|\cdot\|_2^2$. This inequality states that when $\omega(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, a SD iteration $x \mapsto x_+$ reduces the "potential" – the Bregman distance

$$V_x(u) = \omega(u) - [\omega(x) + \langle u - x\, \nabla\omega(x)\rangle] = \tfrac{1}{2}(u-x)^T(u-x)$$

from $u \in X$ to the iterate by at least $\gamma\langle f'(x), x-u\rangle - O(\gamma^2)$.

♠ (2) says that *when $\omega(\cdot)$ is continuously differentiable and strongly convex, modulus 1 w.r.t. $\|\cdot\|$, on X:*

$$\langle \nabla\omega(u) - \nabla\omega(v), u-v\rangle \ge \|u-v\|^2 \ \forall u, v \in X$$

prox-step $x \mapsto x_+ = \mathrm{argmin}_{y\in X}\left[\langle \gamma f'(x), y\rangle + V_x(y)\right]$ ensures inequality similar to (3):

$$\forall u \in X: \gamma\langle f'(x), x-u\rangle \le V_x(u) - V_{x_+}(u) + \tfrac{1}{2}\gamma^2\|f'(x)\|_*^2 \qquad (!)$$
$$[\|\xi\|_* = \mathrm{max}_u\{\langle \xi, u\rangle : \|u\| \le 1\}]$$

Non-Euclidean SD – Mirror Descent

$$\min_{x\in X} f(x) \qquad (P)$$

- X: convex compact set in Euclidean space E
- f: Lipschitz continuous convex function on X

♣ **Setup for MD ("Proximal Setup")** is given by

— norm $\|\cdot\|$ on E

— DGF (Distance-Generating Function) $\omega(\cdot) : X \to \mathbb{R}$ which should be continuously differentiable and strongly convex, modulus 1 w.r.t. $\|\cdot\|$, function on X:

$$\langle \nabla\omega(u) - \nabla\omega(v), u - v\rangle \geq \|u - v\|^2 \, \forall u, v \in X$$

♠ $\omega(\cdot)$ and $\|\cdot\|$ define the important parameter — *ω-capacity of X*

$$\Theta = \max_{u,v\in X} [V_v(u) := \omega(u) - \omega(v) - \langle \nabla\omega(v), u - v\rangle]$$

Note: With "Ball setup" $\omega(u) = \frac{1}{2}\langle u, u\rangle$, $\|u\| \equiv \|u\|_2 = \sqrt{\langle u, u\rangle}$ one has

$$\Theta = \frac{1}{2} \max_{u,v\in X} \|u - v\|_2^2 \quad .$$

♣ As applied to (P), MD generates search points x_t according to

$$\begin{gathered} x_1 \in X, \; x_{t+1} = \mathrm{Prox}_{x_t}(\gamma_t f'(x_t)) := \operatorname*{argmin}_{y\in X} [\langle \gamma_t f'(x_t), y\rangle + V_{x_t}(y)] \, , \\ V_x(y) = \omega(y) - [\omega(x) + \langle y - x, \nabla\omega(x)\rangle] \end{gathered} \qquad (\mathrm{MD})$$

where $\gamma_t > 0$ are stepsizes.

$$x_{t+1} = \mathrm{Prox}_{x_t}(\gamma_t f'(x_t)) := \operatorname*{argmin}_{y \in X} \left[\langle \gamma_t f'(x_t), y\rangle + V_{x_t}(y)\right] \qquad \text{(MD)}$$

Note:

- With Ball setup, (MD) becomes exactly the SD recurrence

$$x_{t+1} = \Pi_X(x_t - \gamma_t f'(x_t))$$

- In order for (MD) to be practical, a step should be easy to implement. Thus, X and $\omega(\cdot)$ should fit each other, meaning that auxiliary problems

$$\min_{y \in X} [\langle \zeta, y\rangle + \omega(y)]$$

should be easy to solve.

Why and how MD converges?

$$\boxed{\begin{array}{c}\{\min_{x\in X} f(x), \omega(\cdot)\} \Rightarrow x_{t+1} = \mathrm{argmin}_{y\in X}\left[\langle \gamma_t f'(x_t), y\rangle + V_{x_t}(y)\right] \\ V_x(y) = \omega(y) - [\omega(x) + \langle y - x, \nabla\omega(x)\rangle]\end{array}}$$

We have seen that MD step ensures inequality

$$\forall u \in X : \gamma_t \langle f'(x_t), x_t - u\rangle \leq V_{x_t}(u) - V_{x_{t+1}}(u) + \tfrac{1}{2}\gamma_t^2 \|f'(x_t)\|_*^2$$

It follows that for positive integers $T_0 \leq T$ one has

$$\sum_{t=T_0}^{T} \gamma_t \underbrace{\langle f'(x_t), x_t - u\rangle}_{\geq f(x_t) - f(u)} \leq V_{x_{T_0}}(u) - V_{x_{T+1}}(u) + \tfrac{1}{2}\sum_{t=T_0}^{T} \gamma_t^2 \|f'(x_t)\|_*^2 \leq \Theta + \tfrac{1}{2}\sum_{t=T_0}^{T} \gamma_t^2 \|f'(x_t)\|_*^2 \qquad (!)$$

$$\left[\Theta = \max_{u,v\in X} V_u(v)\right]$$

For MD, relation (!) plays the same crucial role that the inequality

$$\sum\nolimits_{t=T_0}^{T} \gamma_t \langle f'(x_t), x_t - u\rangle \leq \frac{1}{2}\max_{x,y\in X} \|x - y\|_2^2 + \frac{1}{2}\sum\nolimits_{t=T_0}^{T} \gamma_t^2 \|f'(x_t)\|_2^2$$

played for SD.

$$f_* = \min_{x\in X} f(x)$$
$$\Downarrow$$
$$x_{t+1} = \mathrm{argmin}_{y\in X}\left[\langle \gamma_t f'(x_t) - \nabla\omega(x_t), y\rangle + \omega(y)\right]$$
$$\Downarrow$$

$$\boxed{\sum_{t=T_0}^{T} \gamma_t \langle f'(x_t), x_t - u\rangle \leq \Theta + \tfrac{1}{2}\sum_{t=T_0}^{T} \gamma_t^2 \|f'(x_t)\|_*^2} \qquad (!)$$

For MD, relation (!) plays the same crucial role as the inequality

$$\sum_{t=T_0}^{T} \gamma_t \langle f'(x_t), x_t - u\rangle \leq \frac{1}{2}\max_{x,y\in X}\|x-y\|_2^2 + \frac{1}{2}\sum_{t=T_0}^{T}\gamma_t^2\|f'(x_t)\|_2^2$$

played for SD. Specifically, (!) implies that

$$\boxed{\epsilon_T \equiv \min_{t\leq T} f(x_t) - f_* \leq \frac{\Theta + \frac{1}{2}\sum_{t=T_0}^{T}\gamma_t^2\|f'(x_t)\|_*^2}{\sum_{t=T_0}^{T}\gamma_t}}$$

$$\boxed{\epsilon_T \equiv \min_{t\le T} f(x_t) - f_* \le \frac{\Theta + \frac{1}{2}\sum_{t=T_0}^{T} \gamma_t^2 \|f'(x_t)\|_*^2}{\sum_{t=T_0}^{T} \gamma_t}}$$

As a result,

♣ [Convergence with "divergent series" stepsizes] *Whenever* $0 < \gamma_t \to 0$ *as* $t \to \infty$ *in such a way that* $\sum_t \gamma_t = \infty$, *one has* $\epsilon_T \to 0$ *as* $T \to \infty$

♣ [Optimal stepsize policy] *With stepsizes* $\gamma_t = \frac{\sqrt{2\Theta}}{\|f'(x_t)\|_* \sqrt{t}}$, *one has*

$$\epsilon_T \equiv \min_{t\le T} f(x_t) - f_* \le O(1)\frac{\sqrt{\Theta} L_{\|\cdot\|}(f)}{\sqrt{T}}$$

where $L_{\|\cdot\|}(f)$ *is the Lipschitz constant of* f *w.r.t. the norm* $\|\cdot\|$.

$$\{f_* = \min_{x\in X} f(x), \omega(\cdot) : X \to \mathbb{R}, \Theta = \max_{u,v\in X} [\omega(u) - \omega(v) - \langle \nabla\omega(v), u - v\rangle]\}$$
$$\Rightarrow x_{t+1} = \operatorname{argmin}_{y\in X} [\langle \gamma_t f'(x_t), y\rangle + V_{x_t}(y)], \ \gamma_t = \frac{\sqrt{\Theta}}{\|f'(x_t)\|_* \sqrt{t}}$$
$$\Rightarrow \min_{t\le T} f(x_t) - f_* \le O(1)\frac{\sqrt{\Theta} L_{\|\cdot\|}(f)}{\sqrt{T}}$$

♠ To get the usual SD, one uses

♣ **Ball setup** $\omega(u) = \frac{1}{2}\|u\|_2^2$, $\|\cdot\| = \|\cdot\|_2$ $[X \subset \{x : \|x\|_2 \le R\} \Rightarrow \Theta \le \frac{1}{2}R^2]$

♠ There are several other important setups:

♣ **Simplex setup:** $\|\cdot\| = \|\cdot\|_1$, $X \subset \Delta_n = \{x \in \mathbb{R}^n : x \ge 0, \sum_i x_i \le 1\}$

$$\omega(x) = (1+\delta)\sum_i (x_i + \delta/n)\ln(x_i + \delta/n), \ \delta = 10^{-16}$$

resulting in

$$\Theta \le O(1)\ln(n+1)$$

♣ ℓ_1/ℓ_2 **setup:** $X \subset \mathbb{R}^{k_1} \times \mathbb{R}^{k_2} \times ... \times \mathbb{R}^{k_n}$,

$$\omega([x^1;...;x^n]) = O(1)\left[\sum_{i=1}^n \|x^i\|_2^{\pi_n}\right]^{2/\pi_n}, \ \pi_n = 1 + \frac{1}{n}$$
$$\|[x^1;...;x^n]\| = \sum_i \|x^i\|_2$$

resulting in

$$X \subset \{x : \|x\| \leq R\} \Rightarrow \Theta \leq O(1)\ln(n+1)R^2$$

Note:

• When $k_i = 1$ for all i, $\|\cdot\|$ becomes $\|\cdot\|_1$ and $\omega(x)$ becomes strongly convex with modulus 1, w.r.t. $\|\cdot\|_1$, on the entire $\mathbb{R}^n$.

• When $n = 1$, $\|\cdot\|$ becomes $\|\cdot\|_2$, and $\omega(u)$ becomes $\frac{1}{2}\|u\|_2^2$

♣ **Nuclear norm setup:** $X \subset \mathbb{R}^{p\times q}$,

$$\omega(x) = O(1)\left[\sum_{i=1}^n \sigma_i^{\pi_n}(x)\right]^{2/\pi_n}$$
$$\left[n = \min[p,q], \pi_n = 1 + \frac{1}{n}, \sigma_i(x) : \text{ singular values of } x\right]$$
$$\|x\| = \|x\|_{\text{nuc}} := \sum_i \sigma_i(x)$$

resulting in

$$X \subset \{x : \|x\| \leq R\} \Rightarrow \Theta \leq O(1)\ln(n+1)R^2$$

$$f_* = \min_{x \in X} f(x) \qquad (P)$$

♣ Let us compare the convergence properties of MD with Simplex setup and SD (i.e., MD with Ball setup).

• Observe that in order to apply MD with Simplex setup, X should be a subset of the standard simplex. We can ensure this requirement by scaling and translating the original feasible domain. As a result, MD with Simplex setup becomes applicable to an *arbitrary* convex problem (P) with compact feasible domain X, and the efficiency estimate for the method becomes

$$\epsilon_T[\ \begin{matrix}\text{Simplex}\\ \text{setup}\end{matrix}\ ] = \min_{t \le T} f(x_t) - f_* \le E_{\text{simplex}}(T) := O(1) \ln^{1/2}(n) \overbrace{\max_{x,y \in X} \|x-y\|_1 L_{\|\cdot\|_1}(f)}^{\text{Var}_{\|\cdot\|_1,X}(f)} / \sqrt{T} \qquad (S)$$

while for SD the efficiency estimate is

$$\epsilon_T[\ \begin{matrix}\text{Ball}\\ \text{setup}\end{matrix}\ ] = \min_{t \le T} f(x_t) - f_* \le E_{\text{ball}}(T) := O(1) \overbrace{\max_{x,y \in X} \|x-y\|_2 L_{\|\cdot\|_2}(f)}^{\text{Var}_{\|\cdot\|_2,X}(f)} / \sqrt{T} \qquad (B)$$

The ratio of the right hand side bounds in the estimates is

$$\frac{E_{\text{simplex}}(T)}{E_{\text{ball}}(T)} = O(\sqrt{\ln n}) \cdot \underbrace{\left[\frac{\max_{x,y \in X} \|x-y\|_1}{\max_{x,y \in X} \|x-y\|_2}\right]}_{A} \cdot \underbrace{\left[\frac{L_{\|\cdot\|_1}(f)}{L_{\|\cdot\|_2}(f)}\right]}_{B}$$

$$\frac{E_{\text{simplex}}(T)}{E_{\text{ball}}(T)} = O(\sqrt{\ln n}) \cdot \underbrace{\left[\frac{\max_{x,y\in X}\|x-y\|_1}{\max_{x,y\in X}\|x-y\|_2}\right]}_{A} \cdot \underbrace{\left[\frac{L_{\|\cdot\|_1}(f)}{L_{\|\cdot\|_2}(f)}\right]}_{B}$$

• Small (large) ratio $\frac{E_{\text{simplex}}(T)}{E_{\text{ball}}(T)}$ means that *as far as theoretical accuracy guarantees are concerned, Simplex setup is much better (worse) than Ball setup.*

• The factor $O(\sqrt{\ln n})$ is "against" Simplex setup; however, in practice this factor is just a moderate absolute constant.

• Note that $\frac{\|u\|_1}{\|u\|_2}$ is always ≥ 1 and, depending on x, can be as large as $\sqrt{n}$. Therefore

— factor A is always ≥ 1 (i.e., is "against" Simplex setup). Depending on the geometry of X, it can be as small as 1 and as large as $\sqrt{n}$

— factor B is always ≤ 1 (i.e., is "in favour" of Simplex setup) and can be as small as $\frac{1}{\sqrt{n}}$. The actual value of B is

$$\frac{L_{\|\cdot\|_1}(f)}{L_{\|\cdot\|_2}(f)} = \frac{\max_{x\in X}\|f'(x)\|_\infty}{\max_{x\in X}\|f'(x)\|_2}$$

and depends on the "geometry" of f. For example,

— when all first order partial derivatives of f in X are of the same order ("f is nearly equally sensitive to all variables"), we have

$$B = O\left(\frac{\|(a,\ldots,a)^T\|_\infty}{\|(a,\ldots,a)^T\|_2}\right) = O(n^{-1/2})$$

— when just $O(1)$ first order derivatives of f on X are of the same order, and the remaining derivatives are negligible small ("f is sensitive to just $O(1)$ variables"), we have

$$B = O\left(\frac{\|(a,0,\ldots,0)^T\|_\infty}{\|(a,0,\ldots,0)^T\|_2}\right) = O(1)$$

♣ **Conclusion:** The performance ratio χ depends on the geometry of X and f.

$$\chi = \frac{E_{\text{simplex}}(T)}{E_{\text{ball}}(T)} = O(\sqrt{\ln n}) \cdot \underbrace{\left[\frac{\max_{x,y\in X} \|x-y\|_1}{\max_{x,y\in X} \|x-y\|_2}\right]}_{\substack{A \\ 1 \le A \le \sqrt{n}}} \cdot \underbrace{\left[\frac{L_{\|\cdot\|_1}(f)}{L_{\|\cdot\|_2}(f)}\right]}_{\substack{B \\ 1 \ge B \ge 1/\sqrt{n}}}$$

Extreme example I: X is a ball. In this case, $A = \sqrt{n}$, and since $B \ge \frac{1}{\sqrt{n}}$, $\chi \ge 1$ – method with Ball setup (i.e., the classical SD) outperforms the method with Simplex setup by factor which varies from $O(\sqrt{\ln n})$ (f is nearly equally sensitive to all variables) to $O(\sqrt{n \ln n})$ (f is sensitive to just $O(1)$ variables).

Extreme example II: X is the unit simplex Δ_n. In this case, $A = O(1)$, and since $B \le 1$ and $O(\sqrt{\ln n})$ in practice a moderate absolute constant, $\chi \le O(1)$ – method with Simplex setup outperforms the classical SD by factor which varies from $O\left(\sqrt{\frac{n}{\ln n}}\right)$ (f is nearly equally sensitive to all variables) to $O\left(\sqrt{\frac{1}{\ln n}}\right)$ (f is sensitive to just $O(1)$ variables).

Conclusion: Flexibility in setup allows to adjust MD, to some extent, to the geometry of the problem to be solved. Let all flowers blossom!

Application example:
Positron Emission Tomography Image Reconstruction

♣ The Maximum Likelihood estimate of tracer's density in PET is

$$\lambda_* = \mathrm{argmin}_{\lambda\geq 0}\left\{\sum_{j=1}^n p_j\lambda_j - \sum_{i=1}^m y_i \ln(\sum_{j=1}^n p_{ij}\lambda_j)\right\}$$
$$\left[y_i \geq 0 \text{ are observations}, p_{ij}\geq 0, p_j = \sum_i p_{ij}\right]$$

The KKT optimality conditions read

$$\lambda_j\left(p_j - \sum_i y_i \frac{p_{ij}}{\sum_\ell p_{i\ell}\lambda_\ell}\right) = 0 \quad \forall j,$$

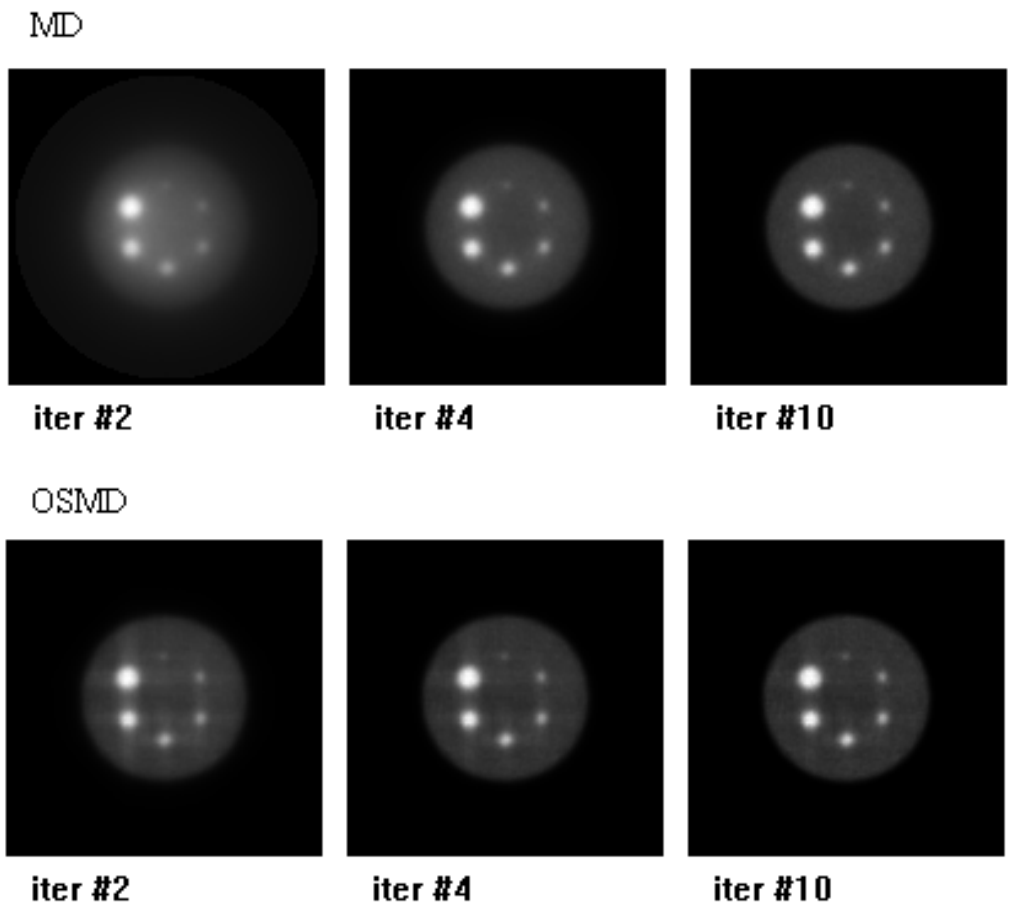
whence, taking sum over j,

$$\sum_j p_j\lambda_j = B \equiv \sum_i y_i.$$

Thus, in fact (PET) is the problem of minimizing over a simplex. Passing to the variables $x_j = p_j B^{-1}\lambda_j$, we end up with the problem

$$\min_x\left\{f(x) = -\sum_i y_i \ln(\sum_j q_{ij}x_j) : x\in\Delta_n\right\} \qquad \text{(PET)}$$
$$\left[q_{ij} = Bp_{ij}p_j^{-1}\right]$$

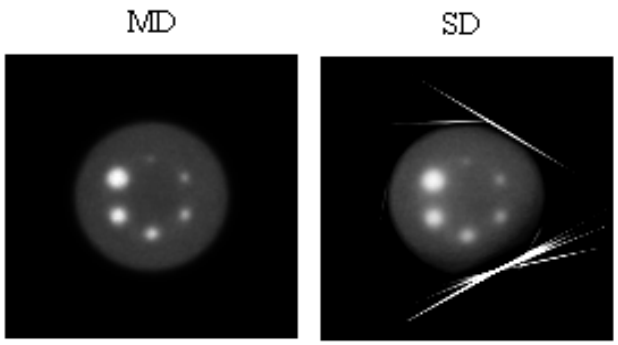
♣ Illustration: "Hot Spheres" phantom ($n = 515,871$)



Itr	1	2	3	4	5	6	7	8	9	10
$f(x_t)$	-4.295	-4.767	-5.079	-5.189	-5.168	-5.230	-5.181	-5.227	-5.189	-5.225

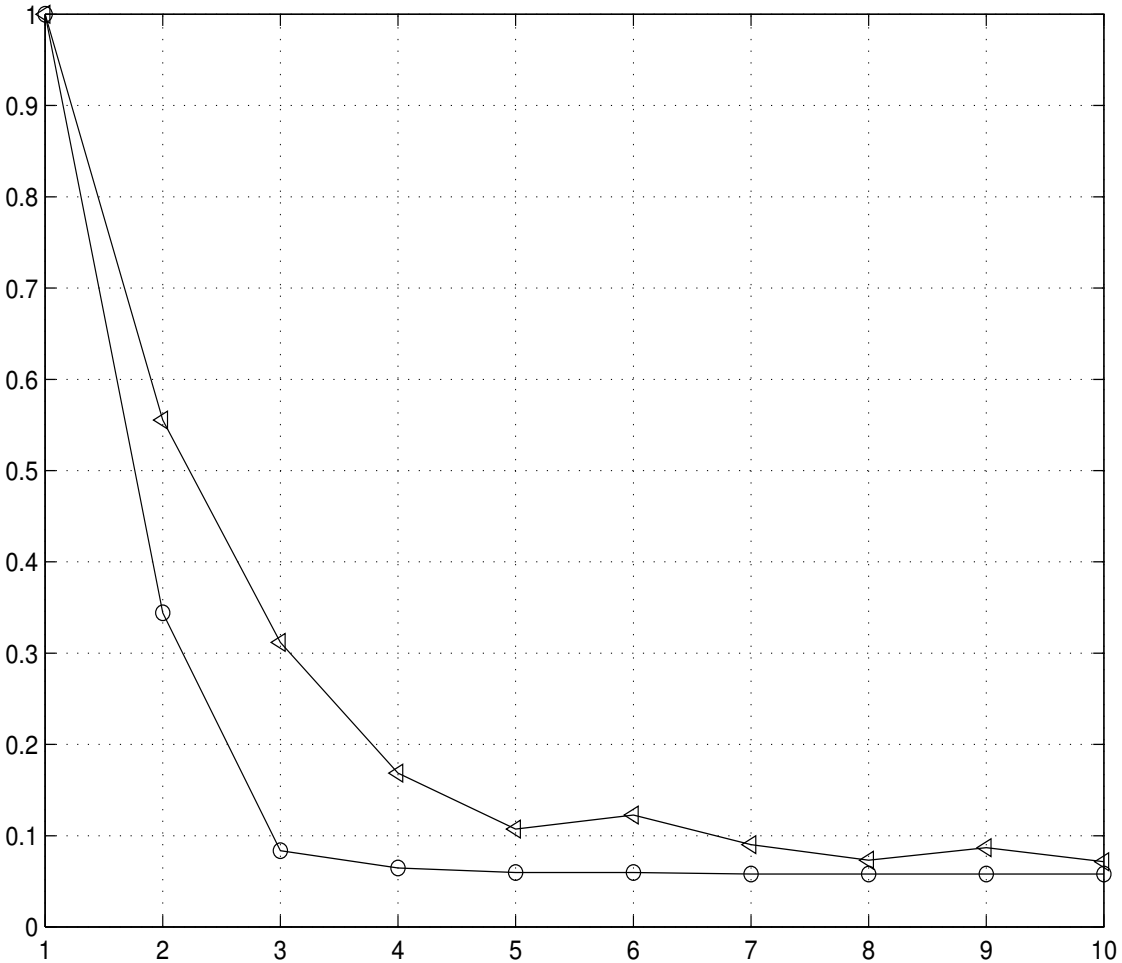
$[f_* \geq -5.283]$

Simplex setup. Progress in accuracy in 10 iterations by factor 21.4



Simplex setup (left) vs. Ball setup (right) progress in accuracy 21.4 vs. 5.26

♣ Illustration: Brain clinical data ($n = 2,763,635$)



Itr	1	2	3	4	5	6	7	8	9	10
$f(x_t)$	-1.463	-1.848	-2.001	-2.012	-2.015	-2.015	-2.016	-2.016	-2.016	-2.016

$[f_* \geq -2.050]$

Simplex setup. Progress in accuracy in 10 iterations by factor 17.5

Mirror-Level Algorithm

♣ Same as SD, the general Mirror Descent admits a version with memory – Mirror Level (ML) algorithm. The setup for ML is similar to the one of MD and is given by a norm $\|\cdot\|$ on E and a continuously differentiable and strongly convex, modulus 1 w.r.t. $\|\cdot\|$, DGF $\omega(\cdot) : X \to \mathbb{R}$.

♣ At step t of ML, we

— compute $f(x_t), f'(x_t)$ and build the current model of f

$$f_t(x) = \max_{\tau \le t}[f(x_\tau) + \langle f'(x_\tau), x - x_\tau\rangle]$$

which underestimates the objective and is exact at the points $x_1, ..., x_t$;

— define the *best found so far value of the objective* $f^t = \min_{\tau \le t} f(x_\tau)$

— define the current *lower bound* f_t on f_* by solving the auxiliary problem

$$f_t = \min_{x \in X} f_t(x)$$

The current *gap* $\Delta_t = f^t - f_t$ is an upper bound on the inaccuracy of the best found so far approximate solution;

— compute the current *level* $\ell_t = f_t + \lambda\Delta_t$ ($\lambda \in (0,1)$ is a parameter)

— finally, we set

$$L_t = \{x \in X : f_t(x) \le \ell_t\},$$

$$x_{t+1} = \mathrm{Prox}_{x_t}^{L_t}(0) := \operatorname*{argmin}_{x \in L_t} \left[\langle -\nabla\omega(x_t), x\rangle + \omega(x)\right]$$

and loop to step $t+1$.

♠ With Ball setup,

$$\mathrm{Prox}_{x_t}^{L_t}(0) = \operatorname*{argmin}_{x \in L_t} \left[-x_t^T x + \frac{1}{2} x^T x \right] = \operatorname*{argmin}_{x \in L_t} \frac{1}{2} \|x - x_t\|_2^2.$$

i.e., the method becomes exactly the BL algorithm.

Efficiency Estimate for ML

Fact: *For every ϵ, $0 < \epsilon < \Delta_1$, the number N of steps of ML before a gap $\leq \epsilon$ is obtained* (i.e., before an ϵ-solution is found) *does not exceed the bound*

$$N(\epsilon) = \frac{4\Theta L^2_{\|\cdot\|}(f)}{\lambda(1-\lambda)^2(2-\lambda)\epsilon^2}.$$

$$\left[\Theta = \max_{x,y\in X}\{V_x(y) := \omega(y) - \omega(x) - \langle y - x, \nabla\omega(x)\rangle\}\right]$$

In particular, for ℓ_1/ℓ_2 and Nuclear Norm setups one has

$$N(\epsilon) = O(\ln n)\frac{\left(\max_{x,y\in X}\|x-y\| L_{\|\cdot\|}(f)\right)^2}{\lambda(1-\lambda)^2(2-\lambda)\epsilon^2}.$$

with $\|\cdot\|$ and n defined in the descriptions of the setups.

Mirror Descent Stochastic Approximation

♣ Consider the case when solving a convex program

$$f_* = \min_{x \in X} f(x)$$

[• $X \subset \mathbb{R}^n$: convex compact • $f : X \to \mathbb{R}$ convex and Lipschitz]

no precise first order information is available. Specifically, we have at our disposal

• *Proximal setup for* X – norm $\|\cdot\|$ and DGF $\omega(\cdot)$

• *Stochastic Oracle* (SO) for f as follows: at t-th call to the oracle, x_t being the input, the oracle returns

$$g(x_t, \xi_t) \in \mathbb{R},\ G(x_t, \xi_t) \in \mathbb{R}^n$$

as random estimates of $f(x_t)$ and $f'(x_t)$, where $\xi_1, \xi_2, ...$ is a sequence of independent realizations of a *random variable* ξ ("oracle's noise").

♠ We assume that the SO is *unbiased*:

$$\mathbf{E}\{g(x, \xi)\} = f(x), \quad \mathbf{E}\{G(x, \xi)\} \in \partial f(x).$$

In addition, we assume that

$$\mathbf{E}\{\|G(x, \xi)\|_*^2\} \leq L^2 < \infty \ \forall x \in X$$

Example: Our f is given as expectation:

$$f(x) = \int_{\Xi} F(x, \xi) dP(\xi),$$

where F is convex in x and efficiently computable.
When we cannot compute the expectation in a closed analytic form, but can instead sample from the distribution P, we, under mild regularity assumptions on F, have at our disposal unbiased Stochastic Oracle

$$g(x, \xi) = F(x, \xi), \quad G(x, \xi) = F'_x(x, \xi)$$

$$f_* = \min_{x \in X} f(x)$$
$$\mathbf{E}\{g(x,\xi)\} = f(x), \mathbf{E}\{G(x,\xi)\} \in \partial f(x), \mathbf{E}\{\|G(x,\xi)\|_*^2\} \leq L^2 < \infty \ \forall x \in X$$

$$\mathrm{Prox}_x(\xi) = \mathrm{argmin}_{u \in X} \left[\langle \xi, u \rangle + \underbrace{\omega(u) - \omega(x) - \langle u - x, \nabla \omega(x) \rangle}_{V_x(u)} \right]$$

♣ We can solve the problem with *Mirror Descent Stochastic Approximation* which is completely similar to MD:

$$x_1 \in X; x_{t+1} = \mathrm{Prox}_{x_t}(\gamma_t G(x_t, \xi_t)), 1 \leq t \leq N;$$
$$x^N = \frac{1}{\gamma_1 + \ldots + \gamma_N} \sum_{t=1}^N \gamma_t x_t.$$

Here $\gamma_t > 0$ are deterministic stepsizes, and $\|\cdot\|$ and the function ω underlying the prox-mapping are given by Proximal setup.

$$x_1 \in X; x_{t+1} = \mathrm{Prox}_{x_t}(\gamma_t G(x_t, \xi_t)), 1 \le t \le N;$$
$$x^N = \frac{1}{\gamma_1 + \ldots + \gamma_N} \sum_{t=1}^N \gamma_t x_t.$$

Fact: *For the MD Stochastic Approximation one has*

$$\mathbf{E}\{f(x^N) - f(x_*)\} \le [\sum_{t=1}^N \gamma_t]^{-1} \mathbf{E}\{\sum_{t=1}^N \gamma_t [f(x_t) - f_*]\} \le \frac{\Theta + \frac{1}{2}\sum_{t=1}^N \gamma_t^2 L^2}{\sum_{t=1}^N \gamma_t},$$
$$\Theta = \max_{x,y \in X} V_x(y)$$

that is, we get exactly the same efficiency estimate as in the case of precise First Order oracle, but now – for the expected inaccuracy of the approximate solution x^N – the weighted sum of the search points we have generated in course of $N = 1, 2, \ldots$ steps.

- **Remark:** Euclidean version

$$x_{t+1} = \mathrm{argmin}_{u \in X} \|[x_t - \gamma_t G(x_t, \xi_t)] - u\|_2^2$$

of Mirror Descent Stochastic Approximation is called *Stochastic Subgradient Descent* and is extremely popular in today Machine Learning.

$$x_1 \in X; x_{t+1} = \mathrm{Prox}_{x_t}(\gamma_t G(x_t, \xi_t)), 1 \le t \le N; x^N = \frac{1}{\gamma_1 + ... + \gamma_N} \sum_{t=1}^N \gamma_t x_t.$$

Convergence Analysis of Mirror Descent Stochastic Approximation

♠ Let us carry out convergence analysis of the algorithm. Denoting by x_* a minimizer of f over X, we, as always, have

$$\sum_{t=1}^N \gamma_t \langle G(x_t, \xi_t), x_t - x_* \rangle \le \Theta + \frac{1}{2} \sum_{t=1}^N \gamma_t^2 \|G(x_t, \xi_t)\|_*^2$$

Taking expectations of both sides and taking into account that x_t is a deterministic function of $\xi_1, ..., \xi_{t-1}$, while $\xi_1, ..., \xi_N$ are independent, we get

$$\sum_{t=1}^N \gamma_t \mathbf{E}\{\langle f'(x_t), x_t - x_* \rangle\} \le \Theta + \frac{1}{2} \sum_{t=1}^N \gamma_t^2 L^2,$$

whence also

$$\mathbf{E}\{\sum_{t=1}^N \gamma_t [f(x_t) - f(x_*)]\} \le \Theta + \frac{1}{2} \sum_{t=1}^N \gamma_t^2 L^2$$

$$\sum_{t=1}^N \gamma_t \mathbf{E}\{f(x_t) - f(x_*)\} \leq \Theta + \tfrac{1}{2}\sum_{t=1}^N \gamma_t^2 L^2 \ \& \ x^N = \tfrac{1}{\gamma_1 + \ldots + \gamma_N} \sum_{t=1}^T \gamma_t x_t$$

By convexity,

$$\mathbf{E}\{f(x^N) - f(x_*)\} \quad \leq \quad [\sum_{t=1}^N \gamma_t]^{-1} \mathbf{E}\{\sum_{t=1}^N \gamma_t [f(x_t) - f(x_*)]\} \leq \frac{\Theta + \frac{1}{2}\sum_{t=1}^N \gamma_t^2 L^2}{\sum_{t=1}^N \gamma_t},$$

as claimed. $\square$

Online Optimization

• **Problem:** Assume on time horizon $1, 2, ..., T$ you and nature (or adversary) play game as follows:
— at time t you are at a point $x_t \in X$, where $X \subset \mathbb{R}^n$ is a once for ever fixed convex compact set.
— at time t the nature/adversary selects a Lipschitz continuous convex function $f_t(x) : X \to \mathbb{R}$ and enforces you to pay the random amount

$$\phi_t(x_t, \xi_t)$$

where ξ_t is random variable, and ϕ_t, ξ_t are such that

$$\mathbf{E}_{\xi_t}\{\phi_t(x, \xi_t)\} = f_t(x),\ x \in X.$$

Besides this, the nature reports stochastic subgradient $G_t(x_t, \xi_t)$ of f_t at x_t:

$$g_t(x) := \mathbf{E}_{\xi_t}\{G_t(x, \xi_t)\} \in \partial f_t(x_t).$$

— you are allowed to use all accumulated so far information to select the next point $x_{t+1} \in X$, and then the process continues.
Important: The random variables $\xi_1, \xi_2, ..., \xi_T$ are mutually independent.

• **Goal:** The performance of your policy for selecting $x_1, ..., x_T$ is the expectation

$$\mathbf{E}_{\xi_1,...,\xi_T}\{\phi_1(x_1,\xi_1)+\phi_2(x_2,\xi_2)+...+\phi_T(x_T,\xi_T)\}$$

of your total payment. In Online Optimization, this performance is compared with the one of "ideal player" who knows the future – the sequence $f_1, ..., f_T$, but not the realization of noices! – in advance, but *cannot move* - must ensure that $x_1 = x_2 = ... = x_T$. Denoting the common value of x_t by x, the ideal player will select x by solving the problem

$$\min_{x\in X}\mathbf{E}_{\xi_1,...,\xi_T}\left\{\sum_{t=1}^{T}\phi_t(x,\xi_t)\right\}=\min_{x\in X}\left\{\sum_{t=1}^{T}f_t(x)\right\}.$$

The difference

$$\text{Regret}=\mathbf{E}_{\xi_1,...,\xi_T}\left\{\sum_{t=1}^{T}\phi_t(x_t,\xi_t)\right\}-\min_{x\in X}\left\{\sum_{t=1}^{T}f_t(x)\right\}$$

is called *regret*; the goal of Online Minimization is to select the policy for updating x_t which makes the regret as small as possible.

Note: The paradigm of Online Minimization is *different* from the one of usual optimization even when $f_t \equiv f$ is independent of t. With the usual approach, an algorithm is an offline process; it does not matter how nonoptimal are the search points — the only thing which matters is how nonoptimal is the resulting approximate solution. In contrast, in Online Optimization with fixed f, we "pay on the fly," and what matters is how good at average, in terms of the objective, are the search points.

♣ **Mirror Descent Regret Minimization.** Let us fix Proximal setup for X — a norm $\|\cdot\|$ on the embedding X linear space E, and a DGF $\omega(x) : X \to \mathbb{R}$ which is continuously differentiable and strongly convex, modulus 1, w.r.t. $\|\cdot\|$. As always, we set

$$\Theta = \max_{u,v\in X}[V_v(u) := \omega(u) - \omega(v) - \langle\nabla\omega(v), u - v\rangle]$$

Assumption: $\mathbf{E}_{\xi_t}\left\{\|G_t(x,\xi_t)\|_*^2\right\} \leq L^2 \ \forall(x \in X, t \leq T)$.

♠ Consider the recurrence

$$x_{t+1} = \mathrm{Prox}_{x_t}[\gamma G_t(x_t,\xi_t)] := \operatorname*{argmin}_{u\in X}\left[\langle\gamma G_t(x_t,\xi_t), y\rangle + V_{x_t}(y)\right], \ t = 1, ..., T.$$

with *fixed* stepsize $\gamma > 0$.

Let $x_* \in \mathrm{Argmin}_{x\in X}\sum_{t=1}^T f_t(x)$. By our standard argument, we have

$$\sum_{t=1}^T \gamma\langle G_t(x_t,\xi_t), x_t - x_*\rangle \leq \Theta + \frac{1}{2}\gamma^2\sum_{t=1}^T \|G_t(x_t,\xi_t)\|_*^2.$$

Taking expectations and recalling that x_t is a deterministic function of $\xi_1, ..., \xi_{t-1}$ and therefore

$$\mathbf{E}_{\xi_t}\{\langle G_t(x_t,\xi_t), x_t - x_*\rangle\} = \langle f_t'(x_t), x_t - x_*\rangle,$$

we get $\sum_{t=1}^T \mathbf{E}\left\{\langle f_t'(x_t), x_t - x_*\rangle\right\} \leq \frac{\Theta}{\gamma} + \gamma T L^2$

$$\begin{aligned}\Rightarrow \text{Regret} &= \mathbf{E}\left\{\sum_{t=1}^T[\phi_t(x_t,\xi_t) - f_t(x_*)]\right\} = \mathbf{E}\left\{\sum_{t=1}^T[f_t(x_t) - f_t(x_*)]\right\} \\ &\leq \mathbf{E}\left\{\sum_{t=1}^T\langle f_t'(x_t), x_t - x_*\rangle\right\} \leq \frac{\Theta}{\gamma} + \frac{\gamma}{2}TL^2\end{aligned}$$

$$\mathbf{E}\left\{\sum_{t=1}^{T}[f_t(x_t) - f_t(x_*)]\right\} \leq \frac{\Theta}{\gamma} + \frac{\gamma}{2}TL^2$$

Setting $\gamma = \frac{\sqrt{2\Theta}}{L\sqrt{T}}$, we get for the policy in question

$$\frac{1}{T}\text{Regret} \leq \frac{\sqrt{2\Theta}L}{\sqrt{T}}$$

Thus, with the MD policy *the average regret per step* $\frac{\text{Regret}}{T}$ *for large* T *can be made as small as* $O(1/\sqrt{T})$.

Note: In the above construction, the stepsize γ is the same for all $t \leq T$ and is "tuned" to the time horizon T we are interested in. With appropriate modification, the stepsize can be made varying in time in such a way that the average, per unit time, regrets on time horizons $T = 1, 2, ...$ will go to zero as $T \to \infty$ at the rate $O(1/\sqrt{T})$.

Application Example: Prediction for Deterministic Boolean Sequence

[for in-depth treatment, see A. Rakhlin, K. Sridharan, *Statistical Learning and Sequential Prediction*, `http://www.mit.edu/~rakhlin/courses/stat928/stat928_notes.pdf`]

Situation: We observe a deterministic Boolean sequence $\xi^N = (\xi_1, ..., \xi_N)$, $\xi_t \in \{0, 1\}$ on time horizon $1, ..., N$.

Goal: To build *predictions* $\widehat{\xi}_t$ which, given $\xi^{t-1} = (\xi_1, ...\xi_{t-1})$, predict (perhaps in randomized fashion) ξ_t, $t = 1, ..., N$.

Performance of a collection $\Xi = \{\widehat{\xi}_t, t \leq N\}$, of predictions is quantified by *average over time expected prediction error*

$$\mathrm{Err}[\Xi] = \mathbf{E}\left\{\frac{1}{N}\sum_{t=1}^{N}\chi(\widehat{\xi}_t, \xi_t)\right\} \qquad \left[\chi(\xi, \xi') = \begin{cases} 0, & \xi = \xi' \\ 1, & \xi \neq \xi' \end{cases}\right]$$

the expectation being taken over the random "driving factors," if any, influencing $\widehat{\xi}_t$ (these factors are present when the predictions indeed are randomized).

Note: *We make no assumptions* on the nature of Boolean sequence ξ^N !!

Basic Predictor: We allow for $\widehat{\xi}_t$ to be randomized: the conditional, given what happened on time horizon $1, ..., t-1$, probability for $\widehat{\xi}_t$ to take value 1 is $x_t \in [0,1]$. Note that

$$\mathbf{E}_{|t-1}\left\{\chi(\widehat{\xi}_t, \xi_t)\right\} = x_t[1-\xi_t] + (1-x_t)\xi_t = f_t(x_t) := |x_t - \xi_t| \qquad (!)$$

where $\mathbf{E}_{|s}$ is the conditional, given realization of driving factors influencing $\widehat{\xi}_1, ..., \widehat{\xi}_s$, expectation.

- To update x_t, we use "online subgradient descent," – the recurrence

$$x_{t+1} = \Pi_\Delta[x_t - \gamma_t f_t'(x_t)], \quad f_t'(x) = \begin{cases} -1, & x < \xi_t \\ 0, & x = \xi_t \\ 1, & x > \xi_t \end{cases}$$

where $\Pi_\Delta(s) = \begin{cases} 0, & s < 0 \\ s, & 0 \le s \le 1 \\ 1, & s > 1 \end{cases}$ is the metric projection on $\Delta = [0,1]$, $x_1 \in [0,1]$ is once for ever fixed, and γ_t are deterministic positive stepsizes satisfying $\gamma_1 \ge \gamma_2 \ge ... \ge \gamma_N$.

Note: The resulting sequence $x_1, ..., x_N$ is deterministic! $\Rightarrow \mathrm{Err}[\Xi] = \frac{1}{N}\sum_{t=1}^N f_t(x_t)$ *by* (!).

Performance Analysis: Let us fix $\bar{x} \in [0,1]$ and set $d_t = \frac{1}{2}(x_t - \bar{x})^2$. By the standard argument, noting that $f_t(x)$ is convex, we have

$$
\begin{aligned}
& \gamma_t f_t'(x_t)(x_t - \bar{x}) \le d_t - d_{t+1} + \tfrac{1}{2}\gamma_t^2 \\
\Rightarrow\ & f_t(x_t) - f_t(\bar{x}) \le f_t'(x_t)(x_t - \bar{x}) \le \tfrac{d_t - d_{t+1}}{\gamma_t} + \tfrac{1}{2}\gamma_t \\
\Rightarrow\ & \textstyle\sum_{t=1}^N [f_t(x_t) - f_t(\bar{x})] \le \sum_{t=1}^N \frac{d_t - d_{t+1}}{\gamma_t} + \frac{1}{2}\sum_{t=1}^N \gamma_t \\
=\ & \tfrac{1}{2}\textstyle\sum_{t=1}^N \gamma_t + \frac{d_1}{\gamma_1} + d_2 \underbrace{\left[\frac{1}{\gamma_2} - \frac{1}{\gamma_1}\right]}_{\ge 0} + d_3 \underbrace{\left[\frac{1}{\gamma_3} - \frac{1}{\gamma_2}\right]}_{\ge 0} + \ldots + d_N \underbrace{\left[\frac{1}{\gamma_N} - \frac{1}{\gamma_{N-1}}\right]}_{\ge 0} - \frac{1}{\gamma_N} d_{N+1} \\
\le\ & \tfrac{1}{2}\textstyle\sum_{t=1}^N \gamma_t + \frac{1}{2}\left[\frac{1}{\gamma_1} + \left[\frac{1}{\gamma_2} - \frac{1}{\gamma_1}\right] + \left[\frac{1}{\gamma_3} - \frac{1}{\gamma_2}\right] + \ldots + \left[\frac{1}{\gamma_N} - \frac{1}{\gamma_{N-1}}\right]\right] \text{ [since } 0 \le d_t \le 1/2] \\
=\ & \tfrac{1}{2}\textstyle\sum_{t=1}^N \gamma_t + \frac{1}{2}\frac{1}{\gamma_N} \\
\Rightarrow\ & \mathrm{Err}[\Xi] = \tfrac{1}{N}\textstyle\sum_{t=1}^N f_t(x_t) \le \frac{1}{N}\sum_{t=1}^n f_t(\bar{x}) + \frac{1}{2N}\left[\sum_{t=1}^N \gamma_t + \frac{1}{\gamma_N}\right]
\end{aligned}
$$

- Let us set $\gamma_t = \frac{\alpha}{\sqrt{t}}$ with some $\alpha > 0$. Then $\sum_{t=1}^N \gamma_t \le \alpha \int_0^N s^{-1/2} ds = 2\alpha N^{1/2}$ and $\frac{1}{\gamma_N} = \alpha^{-1}\sqrt{N}$, and we get $\mathrm{Err}[\Xi] \le \frac{1}{N}\sum_{t=1}^n f_t(\bar{x}) + \frac{1}{2}\left[2\alpha + \frac{1}{\alpha}\right] N^{-1/2}$, which with $\alpha = 1/\sqrt{2}$ yields

$$\mathrm{Err}[\Xi] \le \frac{1}{N}\sum_{t=1}^N |\bar{x} - \xi_t| + \sqrt{2/N}. \qquad (\#)$$

$$\mathrm{Err}[\Xi] \leq \underbrace{\frac{1}{N}\sum_{t=1}^{n}|\bar{x}-\xi_t|}_{E(\bar{x})} + \sqrt{2/N}. \qquad (\#)$$

♠ Now let λ be the fraction of ones in ξ^N. Note that $E(1) = 1-\lambda$ and $E(0) = \lambda$, so that (#) (which holds true for every $\bar{x} \in [0,1]$) implies that

$$\mathrm{Err}[\Xi] \leq \min[\lambda, 1-\lambda] + \sqrt{2/N}. \qquad (!)$$

Conclusions: • When N is large, upper bound (!) on average, over time horizon $1, ..., N$, expected prediction error is close to $\min[\lambda, 1-\lambda]$. The latter quantity *always is* $\leq 1/2$.
• Bound $1/2$ is not interesting: we can arrive at $\mathrm{Err}[\Xi] = 1/2$ when "predicting" by flipping a perfect coin, not using observations at all.
• **However:** In the "asymmetric case" $\min[\lambda, 1-\lambda] < 1/2$, we get a *nontrivial* upper bound on the average expected prediction error – *and this is with no assumptions on* ξ^N *except for asymmetry!*
♠ **Fact:** *When all we know about* ξ^N *is that the fraction of ones in the sequence is a given* λ*, then, for every* $\epsilon > 0$*, no prediction can* <u>*guarantee*</u> *average expected error* $\leq \min[\lambda, 1-\lambda] - \epsilon$*, provided that* N *is large enough!*

Mirror Descent
for
Convex-Concave Saddle Point Problems

♣ **Convex-Concave Saddle Point problem** is

$$\mathrm{SV} = \min_{x\in X} \max_{y\in Y} \phi(x,y) \tag{SP}$$

where:

• $X \subset E_x$, $Y \subset E_y$ are nonempty closed and bounded convex sets in Euclidean spaces E_x, E_y

• $\phi(x,y) : Z := X \times Y \to \mathbb{R}$ is the *cost function* which is Lipschitz continuous, convex in $x \in X$ and concave in $y \in Y$.

♣ *Solutions* to (SP) are, by definition, *saddle points* of ϕ on $X \times Y$, that is, points $(x_*, y_*) \in X \times Y$ where ϕ achieves its minimum in $x \in X$ and its maximum in $y \in Y$:

$$\forall (x \in X, y \in Y) : \phi(x, y_*) \geq \phi(x_*, y_*) \geq \phi(x_*, y).$$

$$\mathrm{SV} = \min_{x\in X} \max_{y\in Y} \phi(x,y) \tag{SP}$$

♠ **Fact:** (SP) gives rise to two optimization problems:

$$\begin{array}{rrcl} (P): & \mathrm{Opt}(P) & = & \min_{x\in X}\left[\overline{\phi}(x) := \max_{y\in Y}\phi(x,y)\right] \\ & & = & \min_{x\in X}\max_{y\in Y}\phi(x,y) \\ (D): & \mathrm{Opt}(D) & = & \max_{y\in Y}\left[\underline{\phi}(y) := \min_{x\in X}\phi(x,y)\right] \\ & & = & \max_{y\in Y}\min_{x\in X}\phi(x,y) \end{array}$$

- We always have $\mathrm{Opt}(P) \geq \mathrm{Opt}(D)$ ["weak duality"]
- ϕ has saddle points on $X \times Y$ *iff* both (P) and (D) are solvable *with equal optimal values:* $\mathrm{Opt}(P) = \mathrm{Opt}(D)$, that is,

$$\min_{x\in X} \max_{y\in Y} \phi(x,y) = \max_{y\in Y} \min_{x\in X} \phi(x,y)$$

["strong duality"]. In this case the saddle points are exactly the pairs ($x \in \mathrm{Argmin}_X\, \overline{\phi}, y \in \mathrm{Argmax}_Y\, \underline{\phi}$).

$$
\begin{array}{rrcl}
(P): & \mathrm{Opt}(P) & = & \min_{x\in X}\left[\overline{\phi}(x) := \max_{y\in Y}\phi(x,y)\right] \\
 & & = & \min_{x\in X}\max_{y\in Y}\phi(x,y) \\
(D): & \mathrm{Opt}(D) & = & \max_{y\in Y}\left[\underline{\phi}(y) := \min_{x\in X}\phi(x,y)\right] \\
 & & = & \max_{y\in Y}\min_{x\in X}\phi(x,y)
\end{array}
$$

• Under our standing assumption (X, Y are nonempty convex compacts, ϕ is Lipschitz continuous convex-concave), *both (P) and (D) are solvable with equal optimal values, that is, saddle points do exist.*

♠ It is natural to quantify the (in)accuracy of an approximate saddle point $(x, y) \in Z := X \times Y$ by its *saddle point residual*

$$\epsilon_{\mathrm{Sad}}(x,y) = \overline{\phi}(x) - \underline{\phi}(y) = [\overline{\phi}(x) - \mathrm{Opt}(P)] + [\mathrm{Opt}(D) - \underline{\phi}(y)]$$

This residual always is nonnegative and is zero iff (x, y) is a saddle point of ϕ.

♣ **Vector field associated with a saddle point problem.** Under our standing assumptions, we can associate with a convex-concave saddle point problem

$$\min_{x\in X}\max_{y\in Y}\phi(x,y)$$

vector field

$$F(z=[x;y])=[F_x(x,y);F_y(x,y)]:Z:=X\times Y\to E_z:=E_x\times E_y$$

with

$$F_x(x,y)\in\partial_x\phi(x,y),\ F_y(x,y)\in\partial_y[-\phi(x,y)]$$

♠ **Assumption:** *From now on, we assume that the vector field $F:Z\to E_z$ is bounded.*

$$F(z=[x;y]) = [F_x(x,y); F_y(x,y)] : Z := X \times Y \to E_z := E_x \times E_y$$
$$F_x(x,y) \in \partial_x \phi(x,y),\ \ F_y(x,y) \in \partial_y[-\phi(x,y)]$$

♠ **Facts:**

- *F is monotone:*

$$\forall (z, z' \in Z := X \times Y) : \langle F(z) - F(z'), z - z' \rangle \geq 0$$

Indeed, setting $z = (x, y)$, $z' = (x', y')$, we have

$$\begin{aligned}&\langle F(z) - F(z'), z - z' \rangle = \langle F_x(x,y) - F_x(x',y'), x - x' \rangle + \langle F_y(x,y) - F_y(x',y'), y - y' \rangle \\ &\geq [\phi(x,y) - \phi(x',y)] + [\phi(x',y') - \phi(x,y')] + [(-\phi)(x,y) - (-\phi)(x,y')] + [(-\phi)(x',y') - (-\phi)(x',y)] \\ &= 0\end{aligned}$$

- *Saddle points of ϕ on $Z = X \times Y$ are exactly the points $z_* \in Z$ such that*

$$\langle F(z), z - z_* \rangle \geq 0 \, \forall z \in Z.$$

♠ **Note:** When Y is a singleton, convex-concave saddle point problem $\min_{x \in X} \max_{y \in Y} \phi(x,y)$ becomes the problem of minimizing a convex function over X. "Convex minimization" versions of the above facts read: *For a Lipschitz continuous convex function $f(x) : X \to \mathbb{R}$*

- *The field $f'(\cdot)$ of subgradients of f is monotone: $\langle f'(x) - f'(y), x - y \rangle \geq 0, x, y \in X$*
- *Minimizers of f on X are exactly the points $x_* \in X$ such that $\langle f'(x), x - x_* \rangle \geq 0 \, \forall x \in X$.*

$$\mathrm{SV} = \min_{x\in X}\max_{y\in Y}\phi(x,y) \tag{SP}$$

• $X \subset E_x$, $Y \subset E_y$ are nonempty closed and bounded convex sets in Euclidean spaces E_x, E_y

• $\phi(x,y) : Z := X \times Y \to \mathbb{R}$ is the *cost function* which is Lipschitz continuous, convex in $x \in X$ and concave in $y \in Y$.

♣ Problems (SP) arise in a wide spectrum of applications. Our major interest in these problems stems from the fact that *numerous "complex" and nonsmooth convex functions $f(x)$ admit saddle point representation:*

$$f(x) = \max_{y\in Y}\phi(x,y)$$

with convex-concave and smooth functions ϕ, which allows to reduce a *nonsmooth* minimization problem

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

to a *smooth* convex-concave saddle point problem

$$\min_{x\in X}\max_{y\in Y}\phi(x,y)$$

and this "gain in smoothness" possesses dramatic potential as far as computationally cheap First Order methods are concerned.

Examples of saddle point reformulations:

- **Maximum of smooth convex functions:**

$$f(x) := \max_{1\le i\le m} f_i(x) = \max_{y\in Y}[\phi(x,y) := \sum_i y_i f_i(x)]$$
$$[Y = \{y \ge 0, \sum_i y_i = 1\}]$$

When f_i are smooth, so is ϕ; when f_i are linear, ϕ is just bilinear.

- **Norm-type functions:**

$$\|Ax - b\| = \max_{y:\|y\|_*\le 1}[\phi(x,y) = \langle y, Ax - b\rangle]$$

- **Maximal eigenvalue of a symmetric matrix:**

$$\lambda_{\max}(x) = \max_{y\in Y}[\phi(x,y) = \mathrm{Tr}(xy)],\ Y = \{y \succeq 0 : \mathrm{Tr}(y) = 1\}$$

Note: Smooth/bilinear saddle point representations admit fully algorithmic calculus. For example,

General case:

$$\begin{array}{rcl} f_i(x) & = & \max_{y_i\in Y_i}\phi_i(x,y_i),\ \lambda_i \ge 0 \\ \Rightarrow \sum_i \lambda_i f_i(x) & = & \max\limits_{y=[y_1;\dots;y_k]\in Y_1\times\dots\times Y_k} \underbrace{\left[\sum_i \lambda_i \phi_i(x,y_i)\right]}_{\phi(x,[y_1;\dots;y_k])} \end{array}$$

Bilinear case:

$$\begin{array}{rcl} f_i(x) & = & \max_{y_i\in Y_i}[\langle a_i, x\rangle + \langle b_i, y_i\rangle + \langle x, A_i y_i\rangle],\ \lambda_i \ge 0 \\ \Rightarrow \sum_i \lambda_i f_i(x) & = & \max\limits_{y=[y_1;\dots;y_k]\in Y_1\times\dots\times Y_k} \left[\sum_i \langle \lambda_i a_i, x\rangle + \langle \lambda_i b_i, y_i\rangle + \langle x, \lambda_i A_i y_i\rangle\right] \\ & = & \max\limits_{y=[y_1;\dots;y_k]\in Y_1\times\dots\times Y_k} \left[\langle \sum_i \lambda_i a_i, x\rangle + \langle [\lambda_1 b_1;\dots;\lambda_k b_k], y\rangle + \langle x, [\lambda_1 A_1,\dots,\lambda_k A_k] y\rangle\right] \end{array}$$

$$\mathrm{SV} = \min_{x\in X}\max_{y\in Y}\phi(x,y) \qquad\qquad (\mathrm{SP})$$
$$\Rightarrow F(z=[x;y]) = [F_x(x,y)\in\partial_x\phi(x,y); F_y(x,y)\in\partial_y[-\phi(x,y)]].$$

• $X\subset E_x$, $Y\subset E_y$ are nonempty closed and bounded convex sets in Euclidean spaces E_x, E_y

• $\phi(x,y): Z:=X\times Y\to\mathbb{R}$ is the *cost function* which is Lipschitz continuous, convex in $x\in X$ and concave in $y\in Y$.

♠ (SP) can be solved by MD. Indeed, let $\|\cdot\|$ be a norm on $E=E_x\times E_y$ and $\omega(\cdot)$ be a DGF for $Z=X\times Y$ which is compatible with $\|\cdot\|$. Consider the process

$$z_1\in Z; z_{t+1}=\mathrm{Prox}_{z_t}(\gamma_t F(z_t)); z^t=\left[\sum_{\tau=1}^t\gamma_\tau\right]^{-1}\sum_{\tau=1}^t\gamma_\tau z_\tau$$
$$[z_\tau=[x_\tau;y_\tau]]$$

♣ **Fact I:** *One has*

$$\epsilon_{\mathrm{Sad}}(x^t,y^t)\le\frac{\Theta+\frac{1}{2}\sum_{\tau=1}^T\gamma_\tau^2\|F(z_\tau)\|_*^2}{\sum_{\tau=1}^T\gamma_\tau},\qquad [\Theta=\max_{z,z'\in Z}V_z(z')]$$

with all consequences related to the rate of convergence, stepsize policies, etc.

Mirror-Prox Scheme

Saddle Point Mirror Descent for $\min_{x\in X}\max_{y\in Y}\phi(x,y)$:

$$z_1\in Z=X\times Y;\, z_{t+1}=\mathrm{Prox}_{z_t}(\gamma_t F(z_t));\, z^N=[\gamma_1+...+\gamma_N]^{-1}\sum_{t=1}^N\gamma_t z_t$$

♣ Consider the *extragradient* Saddle Point MD:

$$z_1\in Z=X\times Y;\, z_t\mapsto w_t=\mathrm{Prox}_{z_t}(\gamma_t F(z_t));\, w_t\mapsto z_{t+1}=\mathrm{Prox}_{z_t}(\gamma_t F(w_t));$$
$$z^t=\left[\sum_{\tau=1}^t\gamma_\tau\right]^{-1}\sum_{\tau=1}^t\gamma_\tau w_\tau$$

♣ **Fact II:** *Let F be Lipschitz:*

$$\|F(z)-F(z')\|_*\le L\|z-z'\|.$$

Then the constant stepsizes

$$\gamma_t\equiv\gamma=\tfrac{1}{L}$$

ensure that

$$\epsilon_{\mathrm{Sad}}(z^t)\le\frac{\Theta}{t\gamma}=\frac{\Theta L}{t},\, t=1,2,...$$

[$1/t$ rate!!!]

♣ **Conclusion:** *When the objective of a convex optimization problem*

$$\mathrm{Opt} = \min_{x \in X} f(x)$$

with convex compact X admits saddle point representation:

$$f(x) = \max_{y \in Y} \phi(x, y)$$

with convex-concave smooth (with Lipschitz continuous gradient) *ϕ and convex compact Y, we can solve the problem at the rate $O(1/t)$, provided we can equip X and Y with "computationally cheap" proximal setup* (i.e., with norms and DGF's resulting in easy-to-compute prox-mappings).

Stochastic Saddle Point Mirror Descent and Acceleration by Randomization

♠ Consider a convex-concave saddle point problem

$$\mathrm{SV} = \min_{x\in X}\max_{y\in Y}\phi(x,y) \qquad \text{(SP)}$$
$$\Rightarrow F(z=(x,y)) = [F_x(x,y)\in\partial_x\phi(x,y); F_y(x,y)\in\partial_y[-\phi(x,y)]]$$

- $X\subset E_x$, $Y\subset E_y$: nonempty closed and bounded convex sets in Euclidean spaces E_x, E_y
- $\phi: X\times Y\to\mathbb{R}$: Lipschitz continuous and convex-concave

♠ $Z = X\times Y$ is equipped with Proximal setup – a norm $\|\cdot\|$ on $E = E_x\times E_y$ and a compatible with this norm DGF $\omega: Z\to\mathbb{R}$.

♠ Assume that the field F is given by Stochastic Oracle:

When calling the oracle at step t, the query point being $z_t = (x_t, y_t)$, the oracle returns a random estimate $G(z_t,\xi_t)$ of $F(z_t)$ which is unbiased and "stochastically bounded":

$$\forall z\in Z = X\times Y: \mathbf{E}\{G(z,\xi)\} = F(z) \ \&\ \mathbf{E}\{\|G(z,\xi)\|_*^2\}\le L^2.$$

As always, $\xi_1,\xi_2,...$ are independent realizations of a random variable ξ.

$$\mathsf{SV} = \min_{x\in X}\max_{y\in Y}\phi(x,y) \qquad\qquad \text{(SP)}$$
$$F(x,y) = [F_x(x,y) \in \partial_x\phi(x,y); F_y(x,y) \in \partial_y[-\phi(x,y)]]$$
$$G(z,\xi) : \mathbf{E}_\xi\{G(z,\xi)\} = F(z) \ \&\ \mathbf{E}_\xi\{\|G(z,\xi)\|_*^2\} \leq L^2\ \forall z = [x;y] \in Z = X\times Y$$

♠ Stochastic Saddle Point Mirror Descent for (SP) is the recurrence

$$z_1 \in Z; z_{t+1} = \mathsf{Prox}_{z_t}(\gamma_t G(z_t,\xi_t)); z^t = \left[\sum\nolimits_{\tau=1}^t \gamma_\tau\right]^{-1}\sum\nolimits_{\tau=1}^T \gamma_\tau z_\tau. \qquad [\gamma_\tau > 0]$$

Theorem: [Lecture Notes, Theorem 5.3.6] *For the above recurrence one has*

$$\mathbf{E}\left\{\epsilon_{\mathsf{Sad}}(z^t)\right\} \leq \frac{7}{2}\cdot\frac{2\Theta + L^2\sum_{\tau=1}^t\gamma_\tau^2}{\sum_{\tau=1}^t\gamma_\tau}.$$
$$\left[\Theta = \max_{u,v\in Z}\left\{V_u(v) := \omega(v) - \omega(u) - \langle v-u, \nabla\omega(u)\rangle\right\}\right]$$

In particular, given a number N of iterations and setting

$$\gamma_t = \frac{\sqrt{2\Theta}}{L\sqrt{N}},\ 1 \leq t \leq N,$$

we ensure that

$$\mathbf{E}\{\epsilon_{\mathsf{Sad}}(z^N)\} \leq \frac{7\sqrt{2\Theta}L}{\sqrt{N}}.$$

Note: Similar results hold true for Mirror Prox.

♣ **Application: Matrix Game.** *Matrix Game* problem is as follows:

$$\mathsf{SV} = \min_{x\in\Delta_n} \max_{y\in\Delta_m} y^T A x \qquad \text{(MG)}$$

$$\left[\Delta_p = \{u \in \mathbb{R}^p : u \geq 0, \textstyle\sum_i u_i = 1\}\right]$$

Interpretation: Two players are playing an antagonistic game; the first selects a $j \in \{1, ..., n\}$, the second selects an $i \in \{1, ..., m\}$. The loss of the first player (i.e., the profit of the second player) is A_{ij}, where A is a given $m \times n$ matrix. Naturally, the first player wants to reduce his losses, and the second player wants to increase his profit.

• When players make their choices simultaneously, there is *no* natural definition of "equilibrium," unless the matrix has a "saddle point" – some entry A_{i_*,j_*} is minimal in its column and is maximal in its row.

• In the general case, the concept of a solution to the game, going back to von Neumann and Morgenstern, is to look what happens when the players repeat the matrix game many times, drawing their choices at random independently of each other and across the time. Denoting by $x \in \Delta_n$ the probability distribution from which the first player draws his choices, and by $y \in \Delta_m$ similar distribution for the second player, the expected loss of the first player (expected profit of the second player) will be

$$y^T A x$$

Thus, (MG) can be thought of as the problem of finding *the best randomized* policies of the players (called their *mixed strategies*); if both players are interested in their long run losses and profits, sticking to the mixed strategies given by a saddle point of the *bilinear* (and thus convex-concave) game (MG) will be optimal policies for every one of them.

$$\mathsf{SV} = \min_{x \in \Delta_n} \max_{y \in \Delta_m} y^T A x \qquad \text{(MG)}$$

$$\left[\Delta_p = \{u \in \mathbb{R}^p : u \geq 0, \textstyle\sum_i u_i = 1\}\right]$$

(MG) is just a primal-dual pair of LP programs:

$$\begin{array}{rcl} \mathsf{Opt}(P) & = & \min_{x \in \Delta_n} \max_i \mathsf{Row}_i^T[A]x \\ \mathsf{Opt}(D) & = & \max_{y \in \Delta_m} \min_j \mathsf{Col}_j^T[A]y \end{array}$$

where $\mathsf{Row}_i^T[A]$ is i-th row, and $\mathsf{Col}_j[A]$ is j-th column in A.
$\Rightarrow$ (MG) can be solved by interior point LP methods.

$$\mathsf{SV} = \min_{x\in\Delta_n}\max_{y\in\Delta_m} y^TAx \qquad \text{(MG)}$$
$$\left[\Delta_p = \{u\in\mathbb{R}^p : u\geq 0, \textstyle\sum_i u_i = 1\}\right]$$

♠ In the large-scale case, (MG) can be solved by Mirror Prox; with appropriate setup, MP yields the efficiency estimate

$$\epsilon_{\mathsf{Sad}}(x^N, y^N) \leq O(1)\sqrt{\ln(n)\ln(m)}\max_{i,j}|A_{ij}|/N$$

The complexity of a step is $O(m+n)$ plus the complexity of two matrix-vector multiplications:

$$\Delta_n \ni x \mapsto Ax,\ \ \Delta_m \ni y \mapsto A^Ty$$

needed to compute the associated with (MG) vector field

$$F(x,y) = \left[\begin{array}{c|c} & A^T \\ \hline -A & \end{array}\right]\left[\begin{array}{c} x \\ y \end{array}\right].$$

When A is a general-type dense matrix, the arithmetic complexity of finding an ϵ-solution to the problem is therefore

$$\mathcal{C}_{\mathsf{determ}}(\epsilon) = O(1)\sqrt{\ln(m)\ln(n)}mn\frac{\max_{i,j}|A_{ij}|}{\epsilon}\ \text{flop.}$$

Can we do better?

♣ **Observation:** Computing matrix-vector multiplication

$$\mathbb{R}^p \ni u \mapsto Bu \in \mathbb{R}^q$$

is easy to randomize:

— the vector $v = \mathrm{abs}[u]/\|u\|_1$ (abs acts coordinatewise) is a probabilistic vector (non-negative entries summing up to 1). Treating v as a probability distribution on $\{1, 2, ..., p\}$, we draw at random an index $\jmath$ from this distribution and return

$$\eta = \|u\|_1 \mathrm{sign}(u_\jmath) \mathrm{Col}_\jmath(B),$$

thus ensuring that $\mathbf{E}\{\eta\} = Bu$.

— generating a realization of η is cheap:

— drawing $\jmath$ costs $O(p)$ flop: in $O(p)$ flop one computes the "cumulative distribution"

$$U_j = \|u\|_1^{-1} \sum_{k<j} |u_k|, \ 1 \le j \le p,$$

of the probabilistic vector, generates $\zeta \sim \mathrm{Uniform}[0, 1]$ and needs $O(\ln(p))$ comparisons to find by Bisection $\jmath$ such that

$$U_{\jmath-1} < \zeta \le U_\jmath$$

— after $\jmath$ is generated, computing η takes just $O(q)$ flop

$$\Rightarrow \textit{arithmetic cost of computing } \eta \textit{ is } O(1)(p+q)$$

- Whatever be a norm $\|\cdot\|$, the noise of our oracle is under control:

$$\|\eta\| \le \|u\|_1 \max_j \|\mathrm{Col}_j[B]\|.$$

The situation is especially nice when $\|u\|_1$ can be bounded in advance.

$$\mathsf{SV} = \min_{x\in\Delta_n}\max_{y\in\Delta_m} y^TAx \qquad \text{(MG)}$$

$$\left[\Delta_p = \{u\in\mathbb{R}^p : u\geq 0, \textstyle\sum_i u_i = 1\}\right] \Rightarrow F(x,y) = \left[\begin{array}{c|c} & A^T \\ \hline -A & \end{array}\right]\left[\begin{array}{c} x \\ y \end{array}\right]$$

♠ Applying the above approach to (MG), we get a cheap randomized oracle for F; a call to this oracle costs just $O(m+n)$ flop, vs. the cost $O(mn)$ of the precise computation of F.

$\Rightarrow$ *Utilizing the cheap stochastic oracle in MD, we get an algorithm for solving* (MG) *which ensures*

$$\mathbf{E}\left\{\epsilon_{\mathsf{Sad}}(x^N,y^N)\right\} \leq O(1)\sqrt{\ln(m)\ln(n)}\left(\frac{\max_{i,j}|A_{ij}|}{\sqrt{N}}\right),$$

with $O(m+n)$ *flop per step.*

$\Rightarrow$ *For every* $\epsilon>0, \delta\in(0,1)$, *one can build in* $(1-\delta)$*-reliable fashion an* ϵ*-solution to* (MG) *at the cost of*

$$\mathcal{C}_{\mathsf{rand}}(\epsilon) = C(\delta)\ln(n)\ln(m)(m+n)/\chi^2 \text{ flop}$$
$$[\chi = \epsilon/\max_{i,j}|A_{ij}|\text{: relative accuracy}]$$

which for fixed δ, χ *and large* m,n *is by orders of magnitude better than the best known "deterministic cost"*

$$\mathcal{C}_{\mathsf{determ}}(\epsilon) = O(1)\cdot\sqrt{\ln(m)\ln(n)}mn/\chi \text{ flop.}$$

of ϵ*-solution to* (MG).

$$\mathcal{C}_{\mathsf{rand}}(\epsilon) = C(\delta)\ln(n)\ln(m)(m+n)/\chi^2 \text{ flop}$$
$$[\chi = \epsilon/\max_{i,j}|A_{ij}|\text{: relative accuracy}]$$

Note: Our algorithm exhibits *sublinear time behavior:* for fixed χ and large m, n, *reliable design of ϵ-solution requires inspection of a negligibly small, going to 0 as m, n grow, randomly selected fraction of the data*.

An "ad hoc" algorithm with this property (in retrospect, pretty similar to Stochastic MD Approximation) was discovered in 1995 by Grigoriadis and Khachiyan.

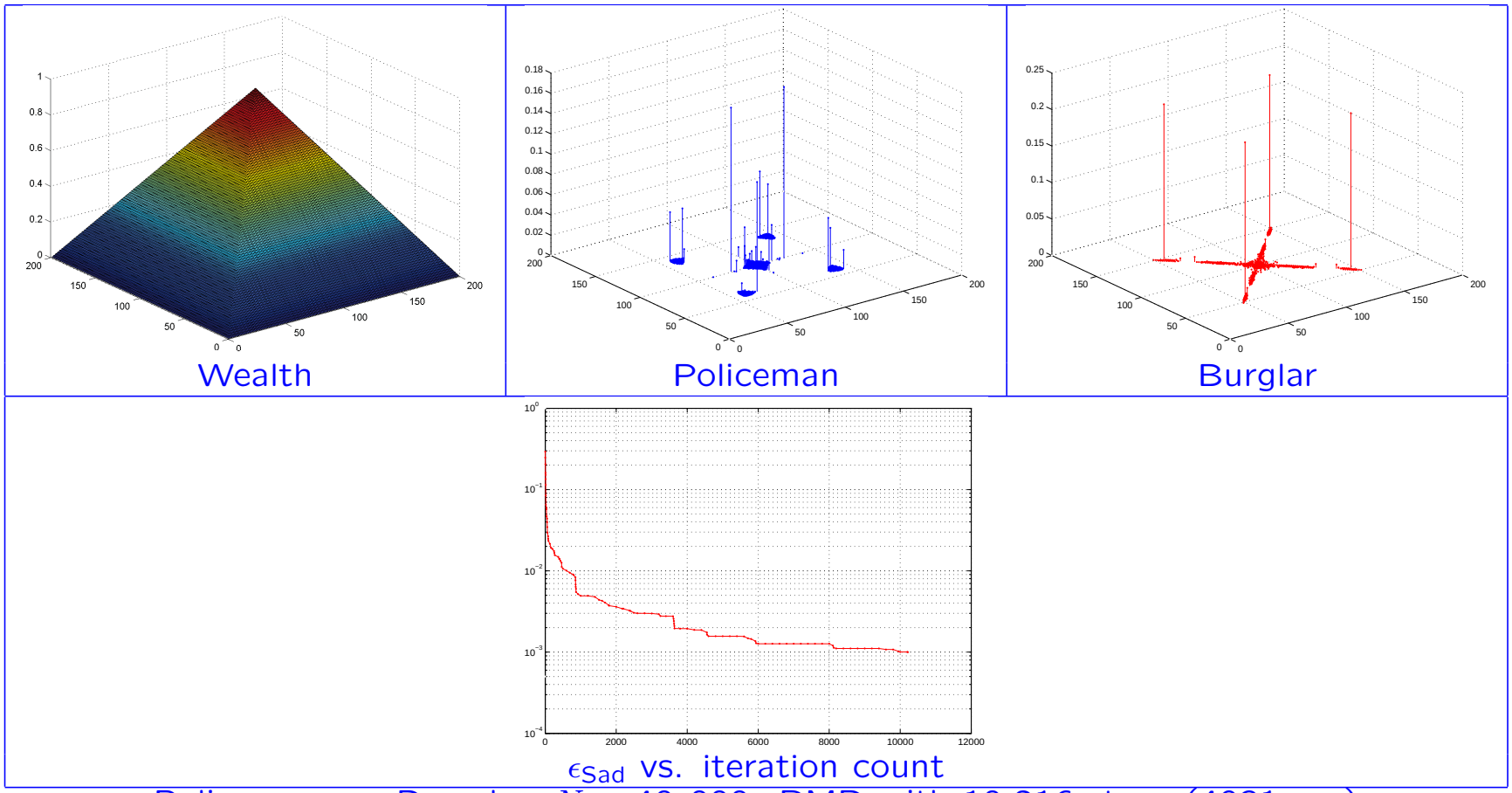
♣ **Illustration:** There are N houses in a city, i-th with wealth w_i. Every evening, Burglar selects a house i to be attacked, and Policeman selects his location at a house j. When the burglary starts, the probability for Policeman to react to alarm and to prevent the burglary is $\exp\{-\theta d(i,j)\}$, where $d(i,j)$ is the distance between locations i and j, so that the expected profit of Burglar is $A_{ij} = w_i[1 - \exp\{-\theta d(i,j)\}]$. Our goal is to solve in mixed strategies the resulting game

$$\max_{y\in\Delta_N} \min_{x\in\Delta_N} y^T A x.$$

♠ Assuming an $n\times n$ equidistant grid of houses with wealth decreasing from the downtown to outskirts, the resulting $(N := n^2) \times N$ matrix game was solved by the state-of-the-art commercial LP Interior Point Method (IPM) `mosekopt`, by the Deterministic Mirror Prox and by the randomized MD seeking $\epsilon_{\text{Sad}} < 0.001$, with CPU limit of 5,300 sec. Here are the results:

	IPM			DMP			RMD		
N	Steps	CPU	Gap	Steps	CPU	Gap	Steps	CPU	Gap
1600	21	120	6.0e-9	78	6	1.0e-3	10556	264	1.0e-3
6400	21	6930	1.1e-8	80	31	1.0e-3	10408	796	1.0e-3
14400	not tested			95	171	1.0e-3	9422	1584	1.0e-3
40000	out of memory			15	5533	0.022	10216	4931	1.0e-3

Policeman vs. Burglar, N houses



Policeman vs. Burgrlar, $N = 40,000$. RMD with 10,216 steps (4931 sec)

Smooth Convex Minimization:
Nesterov's Fast Gradient Method

♣ **Problem of interest: Composite minimization**

$$\mathrm{Opt} = \min_{x \in X} \{\phi(x) = \Psi(x) + f(x)\}$$

- X: closed convex nonempty subset in Euclidean space E
  (X, E) is equipped with proximal setup $(\omega(\cdot), \|\cdot\|)$
- $\Psi : X \to \mathbb{R}$: convex and continuous
- $f : X \to \mathbb{R}$: represented by FO oracle convex function
  with Lipschitz continuous gradient:
  $$\forall x, y \in X : \|\nabla f(x) - \nabla f(y)\|_* \leq L_f \|x - y\|$$

♠ **Main Assumption:** *We are able to compute composite prox-mappings*, i.e., solve auxiliary problems

$$\min_{x \in X} \{\omega(x) + \langle h, x \rangle + \alpha \Psi(x)\} \qquad [\alpha \geq 0]$$

♡ **Example:** LASSO problem

$$\min_{x\in X}\left\{\overbrace{\lambda\|x\|_E}^{\Psi(x)}+\overbrace{\frac{1}{2}\|A(x)-b\|_2^2}^{f(x)}\right\}$$

- $\|\cdot\|_E$: $\begin{cases}\text{(a)} & \text{block } \ell_1/\ell_2 \text{ norm } \sum_{j=1}^n \|x^j\|_2 \text{ on } E=\mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_n} \ (\ell_1 \textit{ case})\\ \text{(b)} & \text{nuclear norm on the space } E \text{ of block diagonal matrices of a given block diagonal structure } (\textit{nuclear norm case})\end{cases}$
- $A(\cdot):E\to\mathbb{R}^m$: linear mapping
- X: either the unit $\|\cdot\|_E$-ball, or the entire E

♡ For properly chosen proximal setup, Main Assumption is satisfied: *computing composite prox mapping*

$$\min_{x\in X}\{\omega(x)+\langle h,x\rangle+\alpha\Psi(x)\} \qquad [\alpha\geq 0]$$

takes $O(\dim E)$ a.o. in the case of (a) and reduces to computing singular value decomposition of a matrix from E in the case of (b).

Example: $\|\cdot\|_E$ is $\|\cdot\|_1$ norm on $\mathbb{R}^n$ ("sparse recovery").

• With Ball setup $\|\cdot\| = \|\cdot\|_2$, $\omega(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ computing composite prox-mapping reduces to solving the problem

$$\min_x \left\{ \sum_i [h_i x_i + \beta |x_i| + \frac{1}{2} x_i^2] : x \in X \right\} \qquad [\beta \geq 0]$$

The problem is trivial when $X = \mathbb{R}^n$ or X is a box $a \leq x \leq b$. When X is the unit $\|\cdot\|_p$-ball, $1 \leq p < \infty$, the problem still is easy – it reduces to *one-dimensional* Lagrange dual problem

$$\max_{\lambda \geq 0} \left[\underline{L}(\lambda) := \underbrace{\min_{x \in \mathbb{R}^n} \sum_i [h_i x_i + \beta |x_i| + \frac{1}{2} x_i^2 + \lambda |x_i|^p] - \lambda}_{\text{easy to compute}} \right]$$

• When $X = E$ or $X = \{x \in E : \|x\|_E \leq 1\}$, computing composite prox-mapping remains easy when the Ball proximal setup is replaced with ℓ_1/ℓ_2 one.

Nesterov's Fast Gradient algorithm for Composite Minimization

♣ **Problem:**

$$\mathrm{Opt} = \min_{x \in X \subset E} \{\phi(x) := \Psi(x) + f(x)\}$$

$$\bullet\ \Psi,\ f\text{: convex and} \qquad (CP)$$

$$\forall x, y \in X : \|\nabla f(x) - \nabla f(y)\|_* \leq L_f \|x - y\|$$

♠ **Assumptions:** *L_f is known and (CP) is solvable with an optimal solution x_*.*

♠ The algorithm is described in terms of proximal setup $(\omega(\cdot), \|\cdot\|)$ for X and *auxiliary sequence*

$$\{L_t \in (0, L_f]\}_{t=0}^{\infty}$$

which can be adjusted on-line.

Recall that DGF ω defines *Bregman distance*

$$V_x(y) = \omega(y) - \omega(x) - \langle \nabla\omega(x), y - x \rangle \ \ [x, y \in X]$$

$$\mathrm{Opt} = \min_{x\in X\subset E} \{\phi(x) := \Psi(x) + f(x)\}$$

♣ **Algorithm:**

♠ **Initialization:** Set

$$A_0 = 0,\ y_0 = x_\omega = \mathrm{argmin}_X\, \omega,\ \psi_0(x) = V_{x_\omega}(x)$$

and select $y_0^+ \in X$ such that $\phi(y_0^+) \le \phi(y_0)$.

♠ **Step** $t = 0, 1, 2, ...$**:** Given $\psi_t(\cdot) = \omega(\cdot) + \alpha\Psi(\cdot)+$ <affine form> $[\alpha \ge 0]$, $y_t^+ \in X$, $A_t \in \mathbb{R}_+$, and L_t, $0 < L_t \le L_f$,

- Compute $z_t = \mathrm{argmin}_{x\in X}\, \psi_t(x)$ (reduces to computing composite prox-mapping)
- Find the positive root a_{t+1} of the equation $L_t a_{t+1}^2 = A_t + a_{t+1}$ and set

$$A_{t+1} = A_t + a_{t+1},\ \ \tau_t = a_{t+1}/A_{t+1} \in (0, 1]$$

- Set $x_{t+1} = \tau_t z_t + (1 - \tau_t) y_t^+$ and compute $f(x_{t+1})$, $\nabla f(x_{t+1})$
- Compute $\widehat{x}_{t+1} = \mathrm{argmin}_{x\in X} \left\{ \langle \nabla f(x_{t+1}), x\rangle + \Psi(x) + \frac{1}{a_{t+1}} V_{z_t}(x) \right\}$ (reduces to computing composite prox-mapping)
- Set

$$\begin{array}{rcl} y_{t+1} & = & \tau_t \widehat{x}_{t+1} + (1 - \tau_t) y_t^+ \\ \psi_{t+1}(x) & = & \psi_t(x) + a_{t+1} \left[f(x_{t+1}) + \langle \nabla f(x_{t+1}), x - x_{t+1}\rangle + \Psi(x) \right] \end{array}$$

and select somehow $y_{t+1}^+ \in X$ such that $\phi(y_{t+1}^+) \le \phi(y_{t+1})$.

- Finally, select $L_{t+1} \in (0, L_f]$.

Step t is completed; go to step $t + 1$.

♣ **Theorem** [Yu. Nesterov '83, '07] *Assume that the sequence* $\{L_t \in (0, L_f]\}$ *is such that*

$$\frac{V_{z_t}(\widehat{x}_{t+1})}{A_{t+1}} + \langle \nabla f(x_{t+1}), y_{t+1} - x_{t+1} \rangle + f(x_{t+1}) \geq f(y_{t+1})$$

(this for sure is the case when $L_t \equiv L_f$*). Then*

$$\boxed{\phi(y_t^+) - \mathrm{Opt} \leq A_t^{-1} V_{x_\omega}(x_*) \leq \frac{4L_f}{t^2} V_{x_\omega}(x_*), \ t = 1, 2, ...}$$

♠ **Illustration:** *As applied to a solvable LASSO problem*

$$x_* = \operatorname*{argmin}_x \left\{ \phi(x) := \lambda \|x\|_E + \frac{1}{2}\|A(x) - b\|_2^2 \right\}$$

with $\|\cdot\|_E$ *either (a) block* ℓ_1/ℓ_2 *norm on* $E = \underbrace{\mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_n}}_{n \text{ factors}}$, *or (b) nuclear norm on* $E = \mathbb{R}^{p \times q}$ *with* $n = \min[p, q]$, *the Fast Gradient method with appropriate proximal setup in* $t = 1, 2, \ldots$ *steps ensures*

$$\phi(y_t^+) \leq \mathrm{Opt} + O(\ln(n+1)) \frac{\|A\|_{E,2}^2}{t^2} \|x_*\|_E^2$$

where $\|A\|_{E,2} = \max\{\|A(x)\|_2 : \|x\|_E \leq 1\}$

♣ **Note:** *$O(1/t^2)$ rate of convergence is, seemingly, the best one can expect from oracle-based methods in the large scale case.*

The precise statement is as follows:

♡ *Let n be a positive integer. Consider Least Squares problems*

$$\mathrm{Opt} = \min_x \|Ax - b\|_2^2 \qquad (QP)$$

with $n \times n$ symmetric matrices A.

For every positive reals R, L and every number $t \leq n/4$ of steps, for every t-step solution algorithm $\mathcal{B}$ operating with the "multiplication oracle" $u \mapsto Au$ one can find an instance of (QP) such that

- *the spectral norm of A does not exceed L,*
- $\mathrm{Opt} = 0$, *and the $\|\cdot\|_2$-norm of some optimal solution does not exceed R,*
- *the approximate solution y generated by $\mathcal{B}$, as applied to the instance, after t calls to the oracle, satisfies*

$$\|Ay - b\|_2^2 \geq O(1)\frac{L^2R^2}{t^2}$$

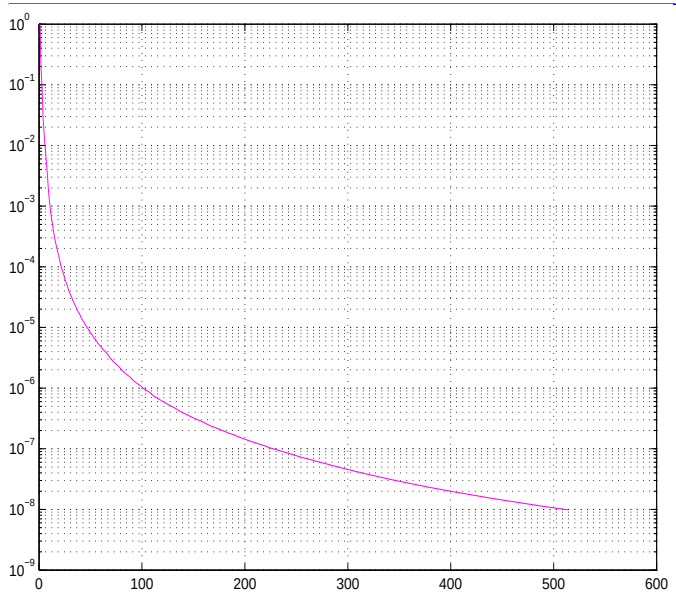
How it Works:
Fast Composite Minimization for LASSO

♣ Test problem:

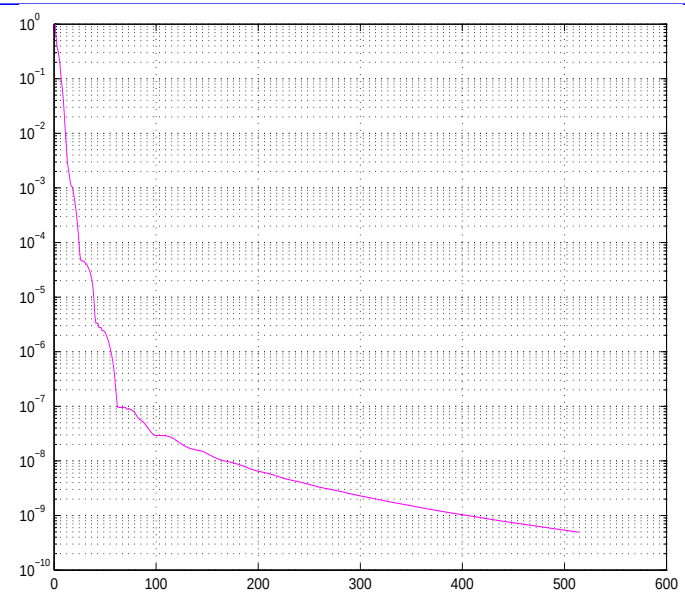
$$\text{Opt} = \min_x \left\{ \phi(x) := 0.01\|x\|_1 + \tfrac{1}{2}\|Ax - b\|_2^2 \right\}$$

with 4096×2048 randomly generated matrix A.

Method	Setup	Iterations	CPU, sec	Nonoptimality
IPM	—	11	103.1	<1.e-12
FGr	Ball setup	512	36.3	2.4e-6
FGr	ℓ_1/ℓ_2 setup	512	36.5	1.2e-7



Ball setup



ℓ_1 setup

Progress in accuracy $\frac{\phi(y_t^+) - \text{Opt}}{\phi(y_0^+) - \text{Opt}}$ vs. t

Platform: 2×3.40 GHz CPU, 16.0 GB RAM, 64-bit Windows 7

Beyond the Scope of Proximal Algorithms: Conditional Gradients

$$\mathrm{Opt} = \min_{x \in X} f(x)$$

♣ **Fact:** All considered so far "computationally cheap" large scale alternatives to IPM's were *proximal type* First Order methods

♠ **But:** *In order to be computationally cheap, a proximal type method should operate with problems on Favorable Geometry domains X* (those allowing for Proximal setup $(\|\cdot\|, \omega(\cdot))$ with moderate ω-capacity Θ, in order to have a reasonable iteration count) *admitting easy to compute prox-mappings* ("Simple Geometry," otherwise an iteration becomes expensive).

♠ Both Favorable and Simple Geometry requirements can be violated. For example,

- when X is a box, Favorable Geometry is missing
- when X is a nuclear norm ball in $\mathbb{R}^{n\times n}$ or a spectahedron (the set of $\succeq 0$ matrices with unit trace) in $\mathbf{S}^n$, we do have Favorable Geometry, but computing the associated prox-mapping requires singular value decomposition of $n \times n$ matrix (or the eigenvalue decomposition of a symmetric $n \times n$ matrix), and both these computations require

$$O(n^3) = O((\dim X)^{3/2}) \text{ a.o.}$$

While much cheaper than the cost $O((\dim X)^3) = O(n^6)$ a.o. of an IPM iteration, $O(n^3)$ a.o. prox-mapping for large n becomes prohibitively time consuming.

Note: nuclear norm balls/spectahedrons arise naturally in many important applications, including, but not reducing to, low rank matrix recovery, multi-class classification in Machine Learning and high dimensional Statistics (and more generally – large scale Semidefinite programming).

♠ Another important example of generic problem with *Complex* Geometry is *Total Variation based Image Reconstruction*

$$\min_{x\in\mathbb{R}^{m\times n}}\left\{\lambda\cdot\mathrm{TV}(x)+\frac{1}{2}\|A(x)-b\|_2^2\right\},$$

where $x=[x_{ij}]\in\mathbb{R}^{m\times n}$ is an $(m\times n)$-pixel image, and $\mathrm{TV}(x)$ is the *Total Variation*:

$$\mathrm{TV}(x)=\sum_{i=1}^{m-1}\sum_{j=1}^{n}|x_{i+1,j}-x_{i,j}|+\sum_{i=1}^{m}\sum_{j=1}^{n-1}|x_{i,j+1}-x_{i,j}|$$

— the ℓ_1-norm of the discrete gradient of $x=[x_{ij}]$. Restricted to the space $\mathbb{M}_0^{m,n}$ of $m\times n$ images *with zero mean*, TV becomes a norm.
For the unit TV-ball, no DGF compatible with the TV norm and leading to easy-to-compute prox mapping is known...

Linear Minimization Oracle

♣ **Observation:** *When $X \subset E$ admits a proximal setup with easy-to-compute prox-mapping, X definitely admits a computationally cheap Linear Minimization Oracle* (LMO) — *a procedure which, given on input a linear form $\langle \eta, \cdot \rangle$, returns*

$$x[\eta] \in \mathrm{Argmin}_{x \in X} \langle \eta, x \rangle$$

Indeed, the optimization program

$$\min_{x \in X} \langle \eta, x \rangle$$

is the "limiting case," as $\theta \to +0$, of the programs

$$\min_{x \in X} \{\theta \omega(x) + \langle \eta, x \rangle\}.$$

♠ **Fact:** Admitting a cheap LMO is a much weaker requirement than admitting proximal setup with cheap prox-mapping, and *there are important domains X with Complex Geometry admitting relatively cheap Linear Minimization Oracle*.

Examples:

A: Nuclear Norm ball $X = \{x \in \mathbb{R}^{m\times n} : \|x\|_{\mathsf{nuc}} \leq 1\}$**.** Here computing $x[\eta]$ reduces to finding the left and the right *leading* singular vectors of $\eta \in \mathbb{R}^{m\times n}$, i.e., to solving the problem

$$\max_{\|u\|_2\leq 1, \|v\|_2\leq 1} u^T\eta v.$$

For large m, n, this is incomparably easier than finding full singular value decomposition of η required to compute prox-mapping.

B: Spectahedron $X = \{x \in \mathbf{S}^n : x \succeq 0, \mathsf{Tr}(x) = 1\}$**.** Here computing $x[\eta]$ reduces to finding the leading eigenvector of $-\eta$, i.e., to solving the problem

$$\min_{\|u\|_2=1} u^T\eta u.$$

For large n, this is incomparably easier than finding full eigenvalue decomposition of η required to compute prox-mapping.

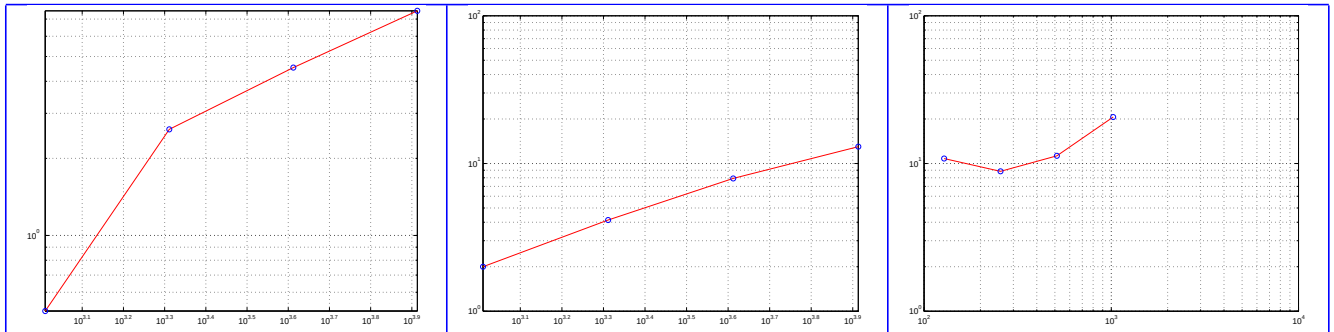
Examples (continued):

C: Unit TV-ball $X = \{x \in \mathbb{M}_0^{m,n} : \mathbf{TV}(x) \leq 1\}$**:** *For* $\eta \in \mathbb{M}_0^{m,n}$*, a point* $x[\eta] \in \mathrm{Argmin}_{x \in X} \mathrm{Tr}(\eta x^T)$ *is readily given by the optimal Lagrange multipliers for the capacitated network flow problem*

$$\max_{t,f} \{t : \Gamma f = t\eta, \|f\|_\infty \leq 1\}$$

Γ: incidence matrix of the network with nodes (i,j), $1 \leq i \leq m$, $1 \leq j \leq n$, and arcs $(i,j) \to (i+1,j)$, $(i,j) \to (i,j+1)$

♠ **Illustration:**



A: CPU ratio "full svd" / "finding leading singular vectors" for $n \times n$ matrix vs. n

n	1024	2048	4096	8192
CPU ratio	0.5	2.6	4.5	7.5

Full svd for $n = 8192$ takes 475.6 sec!

B: CPU ratio "full evd" / "finding leading eigenvector" for $n \times n$ symmetric matrix vs. n

n	1024	2048	4096	8192
CPU ratio	2.0	4.1	7.9	13.0

Full evd for $n = 8192$ takes 142.1 sec!

C: CPU ratio "metric projection" / "LMO computation" for TV ball in $\mathbb{M}_0^{n,n}$ vs. n

n	129	256	512	1024
CPU ratio	10.8	8.8	11.3	20.6

Metric projection onto TV ball for $n = 1024$ takes 1062.1 sec!

Platform: 2×3.40 GHz CPU, 16.0 GB RAM, 64-bit Windows 7

Conditional Gradient Algorithm

$$\mathrm{Opt} = \min_{x\in X} f(x) \qquad (CM)$$
$$[\bullet\ X \subset E\text{: convex compact set} \bullet\ f : X \to R\text{: convex}]$$

W.l.o.g. we assume that X linearly spans the embedding Euclidean space E.

♣ When X is given by Linear Minimization oracle and f is smooth, (CM) can be solved by Conditional Gradient (CndG), a.k.a. Frank-Wolfe, algorithm given by the recurrence

$$\begin{array}{rcl} x_1 \in X,\ x_{t+1} & \in & X : f(x_{t+1}) \le f\left(x_t + \frac{2}{t+1}(x_t^+ - x_t)\right), \\ & & \left[x_t^+ = x[\nabla f(x_t)] \in \mathrm{Argmin}_{y\in X}\langle \nabla f(x_t), y\rangle\right] \\ f_*^t & = & \max_{\tau\le t}\left[f(x_\tau) + \langle \nabla f(x_\tau), x_\tau^+ - x_\tau\rangle\right] \le \mathrm{Opt} \end{array}$$

♠ **Theorem:** *Let $f : X \to \mathbb{R}$ be convex and (κ, L)-smooth:*

$$\forall x, y \in X : f(y) \le f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{\kappa}\|x - y\|_X^\kappa$$
$$\left[\begin{array}{ll} \bullet & L < \infty,\ \kappa \in (1, 2]\text{: parameters} \\ \bullet & \|\cdot\|_X\text{: norm with the unit ball } \frac{1}{2}[X - X] \end{array}\right]$$

When solving (CP) by CndG, one has for $t = 2, 3, ...$

$$f(x_t) - \mathrm{Opt} \le f(x_t) - f_t^* \le \frac{2^{2\kappa}}{\kappa(3-\kappa)} \cdot \frac{L}{(t+1)^{\kappa-1}}$$

$$\forall x, y \in X : f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{\kappa} \|x - y\|_X^\kappa \qquad (!)$$

$$\left[\bullet \quad L < \infty, \ \kappa \in (1, 2]: \text{ parameters } \right]$$

Note: A *sufficient* condition for (!) is Hölder continuity of $\nabla f(x)$:

$$\|\nabla f(x) - \nabla f(y)\|_{X,*} \leq L \|x - y\|_X^{\kappa - 1} \ \forall x, y \in X$$

For convex f and $\kappa = 2$, this condition is also *necessary* for (!).

$$\forall x, y \in X : f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{\kappa} \|x - y\|_X^\kappa$$

♣ Typically, the CndG rate of convergence $O(1/T^{\kappa-1})$ is not the best we can hope for. For example, when $\kappa = 2$ and X is either

- the unit $\|\cdot\|_p$ ball in $\mathbb{R}^n$ with $1 \leq p \leq 2$, or
- the unit nuclear norm ball in $\mathbb{R}^{n \times n}$,

Nesterov's Fast Gradient method converges at the rate

$$O(1) \ln(n+1) L^2/t^2,$$

and CndG only at the rate $O(1)L/t$. In fact,

♡ *In Favorable Geometry case, the only, if any, disadvantage of proximal algorithms as compared to CndG is the necessity to compute prox mappings,* which could be expensive for problems with Complex Geometry.

♠ *Beyond the case of Favorable Geometry, CndG can be optimal.*

Fact: *Let X be n-dimensional box:*

$$X = \{x \in \mathbb{R}^n : \|x\|_\infty \leq 1\}.$$

Then for every $t \leq n$, $L < \infty$, $\kappa \in (1, 2]$, and every utilizing local oracle t-step method $\mathcal{B}$ for minimizing (κ, L)-smooth convex functions over X there exists a function f in the family such that for the approximate minimizer $x_{\mathcal{B}}$ of f generated by $\mathcal{B}$ it holds

$$f(x_{\mathcal{B}}) - \min_X f \geq \frac{O(1)}{\ln(n)} \frac{L}{t^{\kappa-1}}$$

$\Rightarrow$ *When minimizing smooth convex functions, represented by a local oracle, over an n-dimensional box, t-step CndG cannot be accelerated by more than $O(\ln(n))$ factor*, provided $t \leq n$.

- The result remains true when replacing n-dimensional box X with its matrix analogy

$$\{x \in \mathbb{R}^{n \times n} : \text{ spectral norm of } x \text{ is } \leq 1\}$$

- When minimizing (κ, L)-smooth functions over n-dimensional $\|\cdot\|_p$-balls with $2 \leq p \leq \infty$, the rate-of-convergence advantages of proximal algorithms over CndG rapidly deteriorate as p grows and disappears (up to $O(\ln(n))$-factor) when p becomes as large as $O(\ln(n))$.

Proof of Theorem

$$
\begin{array}{ll}
(a) & f(y) \le f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{\kappa}\|y-x\|_X^\kappa \\
(b) & f(x_{t+1}) \le f(x_t + \gamma_t(x_t^+ - x_t)), \\
 & \gamma_t = \frac{2}{t+1},\ x_t^+ \in \mathrm{Argmin}_{y\in X}\langle \nabla f(x_t), y\rangle \\
 & f_*^t := \underbrace{\max_{\tau\le t}\left[f(x_\tau) + \langle \nabla f(x_\tau), x_\tau^+ - x_\tau\rangle\right]}_{\le \min_X f}
\end{array}
$$

$$
? \Rightarrow ? \qquad f(x_t) - f_*^t \le \frac{2^{\kappa+1}L}{\kappa(3-\kappa)}\gamma_t^{\kappa-1} \quad (!_t), t \ge 2
$$

Let

$$
\epsilon_t = f(x_t) - f_*^t,\ e_t = \langle \nabla f(x_t), x_t - x_t^+\rangle
$$

- $f_*^t \ge f(x_t) + \langle \nabla f(x_t), x_t^+ - x_t\rangle \Rightarrow e_t \ge \epsilon_t$

We have

$$
\begin{array}{lll}
(c) & \|x_t - x_t^+\|_X \le 2 & \\
\Rightarrow & f(x_{t+1}) & \le f(x_t + \gamma_t(x_t^+ - x_t)) \text{ [by } (b)] \\
 & & \le f(x_t) + \gamma_t\langle \nabla f(x_t), x_t^+ - x_t\rangle + \frac{L}{\kappa}[2\gamma_t]^\kappa \\
 & & \qquad\qquad \text{[by } (a), (c)] \\
 & & = f(x_t) - \gamma_t e_t + \frac{2^\kappa L}{\kappa}\gamma_t^\kappa \\
 & & \le f(x_t) - \gamma_t\epsilon_t + \frac{2^\kappa L}{\kappa}\gamma_t^\kappa \text{ [since } e_t \ge \epsilon_t] \\
\Rightarrow & \epsilon_{t+1} = f(x_{t+1}) - f_*^{t+1} \le f(x_{t+1}) - f_*^t & \\
 & & \qquad\qquad \text{[since } f_*^{t+1} \ge f_*^t] \\
 & & \le \epsilon_t(1-\gamma_t) + \frac{2^\kappa L}{\kappa}\gamma_t^\kappa
\end{array}
$$

$$[0 \le] \quad \epsilon_{t+1} \le \epsilon_t(1-\gamma_t) + \frac{2^\kappa L}{\kappa}\gamma_t^\kappa \quad (*_t)$$

$$? \Rightarrow ? \quad \epsilon_t \le \frac{2^{\kappa+1}L}{\kappa(3-\kappa)}\gamma_t^{\kappa-1},\ t \ge 2 \quad [\gamma_t = \frac{2}{t+1}] \quad (!_t)$$

- By $(*_2)$, we have $\epsilon_2 \le \frac{2^\kappa L}{\kappa} \Rightarrow \epsilon_2 \le \frac{2^{\kappa+1}L}{\kappa(3-\kappa)}(2/3)^{\kappa-1}$ due to $1 < \kappa \le 2 \Rightarrow (!_2)$ *holds true.*
- Assuming $(!_t)$ true for some $t \ge 2$, we have

$$\begin{aligned}
\epsilon_{t+1} &\le \frac{2^{\kappa+1}L}{\kappa(3-\kappa)}\gamma_t^{\kappa-1}(1-\gamma_t) + \frac{2^\kappa L}{\kappa}\gamma_t^\kappa \text{ [by } (*_t) \text{ and } (!_t)] \\
&= \frac{2^{\kappa+1}L}{\kappa(3-\kappa)}\left[\gamma_t^{\kappa-1} - \frac{\kappa-1}{2}\gamma_t^\kappa\right] \\
&= \frac{2^{\kappa+1}L}{\kappa(3-\kappa)}2^{\kappa-1}\left[(t+1)^{1-\kappa} + (1-\kappa)(t+1)^{-\kappa}\right] \\
&\le \frac{2^{\kappa+1}L}{\kappa(3-\kappa)}2^{\kappa-1}(t+2)^{1-\kappa} \text{ [by convexity of } (t+1)^{1-\kappa}] \\
&= \frac{2^{\kappa+1}L}{\kappa(3-\kappa)}\gamma_{t+1}^{\kappa-1} \Rightarrow (!_{t+1}) \textit{ holds true.}
\end{aligned}$$

Thus, $(!_t)$ holds true for all t, Q.E.D.

Conditional Gradient Algorithm for Norm-regularized Smooth Convex Minimization

Source: Harchaoui, Z., Juditsky, A., Nemirovski, A. Conditional Gradient Algorithms for Norm-Regularized Smooth Convex Optimization. *Mathematical Programming* 152:1-2 (2015), 75–112. `https://www2.isye.gatech.edu/~nemirovs/HarchaouiJudNem.pdf`

♣ "As is", CndG is applicable only to minimizing *smooth* convex functions on *bounded* and closed convex domains.

Question: *How to apply CndG to Composite Minimization problem*

$$\mathrm{Opt} = \min_{x\in\mathbf{K}} \{\lambda\|x\| + f(x)\}$$

- $\mathbf{K}$: closed convex cone in Euclidean space E
- $\|\cdot\|$: norm on E
- $\lambda > 0$: penalty
- $f : \mathbf{K} \to \mathbb{R}$: convex function with Lipshitz continuous gradient:
$$\|\nabla f(x) - \nabla f(y)\|_* \le L_f\|x-y\|,\ x,y \in \mathbf{K}$$

♠ **Main Assumption:** *We have at our disposal LMO oracle for the intersection of the unit $\|\cdot\|$-ball with the cone $\mathbf{K}$. Given on input a linear form $\langle\eta,\cdot\rangle$ on E, the oracle returns*
$$x[\eta] \in \mathrm{Argmin}_x\{\langle\eta,x\rangle : x\in\mathbf{K}, \|x\|\le 1\}$$

Examples:

A. $E = \mathbb{R}^{m\times n}$, $\|\cdot\| = \|\cdot\|_{\mathrm{nuc}}$, $\mathbf{K} = E$

B. $E = \mathbf{S}^n$, $\|\cdot\| = \|\cdot\|_{\mathrm{nuc}}$, $\mathbf{K} = \mathbf{S}^n_+ = \{x\in E : x \succeq 0\}$

C. $E = \mathbb{M}_0^{m,n}$, $\|\cdot\| = \mathrm{TV}(\cdot)$, $\mathbf{K} = E$.

♣ We can reformulate the problem of interest as

$$\mathrm{Opt} = \min_{[x;r]\in\mathbf{K}^+} \{\phi(x,r) := \lambda r + f(x)\}$$
$$\mathbf{K}^+ = \{[x;r] \in E^+ := E \times \mathbb{R} : x \in \mathbf{K},\ \|x\| \le r\}$$

♠ **Assumption:** *There exists $D_* < \infty$ such that*

$$y := [x;r] \in \mathbf{K}^+ \ \&\ r > D_* \Rightarrow \phi(y) > \phi(0),$$

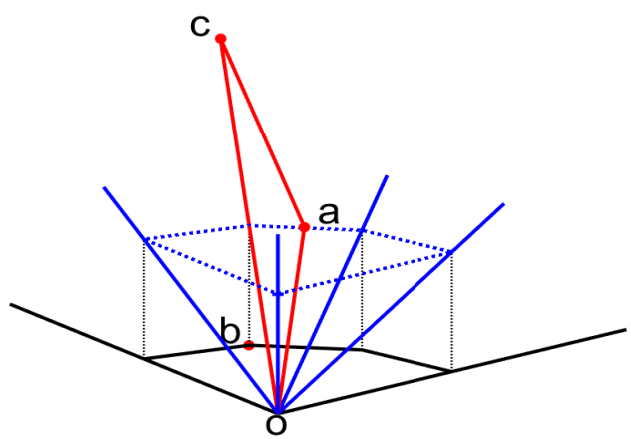
and we are given a finite upper bound D^+ on D_*.
Note: *The efficiency estimate for the forthcoming method depends on D_*, and not on D^+!*
♠ **Algorithm:**
- **Initialization:** Set $y_1 = 0 \in \mathbf{K}^+$
- **Step** $t = 1, 2, \ldots$ Given $y_t = [x_t; r_t] \in \mathbf{K}^+$,
  - compute $\nabla f(x_t)$
  - compute $x_t^+ = x[\nabla f(x_t)] \in \mathrm{Argmin}_x \{\langle \nabla f(x_t), x\rangle : x \in \mathbf{K}, \|x\| \le 1\}$
  - set $\Delta_t = \mathrm{Conv}\left\{y_t, 0, D^+[x_t^+; 1]\right\} \subset \mathbf{K}^+$ and find $y_{t+1} \in \mathbf{K}^+ : \phi(y_{t+1}) \le \min_{y\in\Delta_t} \phi(y)$

Step t is completed; pass to step $t+1$.

$$\min_{x\in\mathbf{K}}[\lambda\|x\|+f(x)] \Leftrightarrow \min_{[x;r]\in\mathbf{K}^+}[\phi(x,r)=\lambda r+f(x)]$$
$$\left[\mathbf{K}^+=\{[x;r]:x\in\mathbf{K},r\geq\|x\|\}\right]$$

♠ **Algorithm:**

- **Initialization:** Set $y_1=0\in\mathbf{K}^+$
- **Step** $t=1,2,...$ Given $y_t=[x_t;r_t]\in\mathbf{K}^+$,
  - compute $\nabla f(x_t)$
  - compute $x_t^+=x[\nabla f(x_t)]\in\text{Argmin}_x\{\langle\nabla f(x_t),x\rangle : x\in\mathbf{K},\|x\|\leq 1\}$
  - set $\Delta_t=\text{Conv}\left\{y_t,0,D^+[x_t^+;1]\right\}\subset\mathbf{K}^+$ and find $y_{t+1}\in\mathbf{K}^+ : \phi(y_{t+1})\leq\min_{y\in\Delta_t}\phi(y)$



Geometry of step

- $\mathbf{K}$: quadrant on the XY plane • black polygon: the set $\{x\in\mathbf{K}:\|x\|\leq 1\}$
- blue polygon: intersection of $\mathbf{K}^+$ with the hyperplane $r=1$
- a: current iterate y_t • b: $x_t^+\in\text{argmin}_{x\in\mathbf{K},\|x\|\leq 1}\langle\nabla f(y_t),x\rangle$ • c: $D_+\cdot[x_t^+;1]$
- $y_{t+1}\in\mathbf{K}^+$: $\phi(y_{t+1})\leq\min_{y\in\Delta_t}\phi(y)$, Δ_t: triangle with vertices o,a,c

Note: One can set $y_{t+1}\in\text{Argmin}_{y\in\Delta_t}\phi(y)$. With this policy, a step requires minimizing *ϕ over a 2D triangle Δ_t*, which can be done within machine precision in $O(1)$ steps (e.g., by the Ellipsoid method).

$$\mathrm{Opt} = \min_{[x;r]\in\mathbf{K}^+} \{\phi(x,r) := \lambda r + f(x)\}$$
$$\mathbf{K}^+ = \{[x;r] \in E^+ := E \times \mathbb{R} : x \in \mathbf{K}, \|x\| \le r\}$$

♣ **Theorem:** *For the outlined algorithm,*

$$\phi(y_t) - \mathrm{Opt} \le \frac{8L_f D_*^2}{t+14},\ t = 2, 3, ...$$

♠ **Bundle Implementation:** We can set

$$y_{t+1} \in \mathrm{Argmin}_y \{\phi(y) : y \in \mathrm{Conv}\{0 \cup Y_t\}\} \qquad (*)$$
$$Y_t \subset \mathbf{K}^+\text{: finite set containing } y_t = [x_t; r_t] \text{ and } D^+[x_t^+; 1], \text{ with}$$
$$x_t^+ \in \mathrm{Argmin}_x \{\langle \nabla f(x_t), x\rangle : x \in \mathbf{K}, \|x\| \le 1\}$$

For example, we can comprise Y_t of $y_t, D^+[x_t^+; 1]$ and several of the previous iterates $y_1, ..., y_{t-1}$.

♡ Bundle approach is especially attractive when

$$f(x) = \Psi(Ax + b)$$

for easy to compute Ψ, like $\Psi(u) = \frac{1}{2}u^T u$. Here computing f, ∇f at a convex (or linear) combination $x = \sum \lambda_i x_i$ of points x_i *with already computed* Ax_i becomes cheap: $Ax = \sum_i \lambda_i (Ax_i)$.

$\Rightarrow$ *the FO oracle for* $(*)$ *is computationally cheap*

$$y_{t+1} \in \mathrm{Argmin}_y \{\phi(y) : y \in \mathrm{Conv}\{0 \cup Y_t\}\} \qquad (*)$$
$$Y_t \subset \mathbf{K}^+\text{: finite set containing } y_t = [x_t; r_t] \text{ and } D^+[x_t^+; 1], \text{ with}$$
$$x_t^+ \in \mathrm{Argmin}_x \{\langle \nabla f(x_t), x\rangle : x \in \mathbf{K}, \|x\| \leq 1\}$$

• For example, with $f(x) = \frac{1}{2}\|Ax - b\|_2^2$, solving $(*)$ reduces to solving $k_t = \mathrm{Card}(Y_t)$-dimensional convex quadratic problem

$$\min_{\lambda \in \mathbb{R}^{k_t}} \left\{ \frac{1}{2}\lambda^T Q_t \lambda + 2q_t^T \lambda : \lambda \geq 0, \sum_j \lambda_j \leq 1 \right\}, \qquad (!)$$
$$Q_t = [x_i^T A^T A x_j]_{i,j}$$

where x_j, $1 \leq j \leq k_t$, are the x-components of the points from Y_t.
$\Rightarrow$ *Assuming that Y_t is a set of moderate cardinality* (say, few tens) *obtained from Y_{t-1} by discarding several "old" points and adding the new points $y_t = [x_t; r_t], D^+[x_t^+; 1]$, updating*

$$[Q_{t-1}, q_{t-1}] \mapsto [Q_t, q_t]$$

basically reduces to computing matrix-vector products Ax_t and Ax_t^+. After Q_t, q_t are computed, (!) *can be solved "in no time" by an IPM.*
Note: Ax_t is computed anyway when computing $\nabla f(x_t)$.

How It Works: TV-based Image Reconstruction

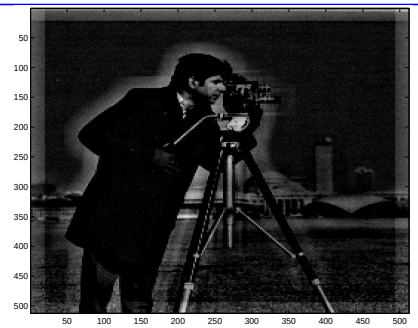
True image

Blurred noisy image, 40% noise

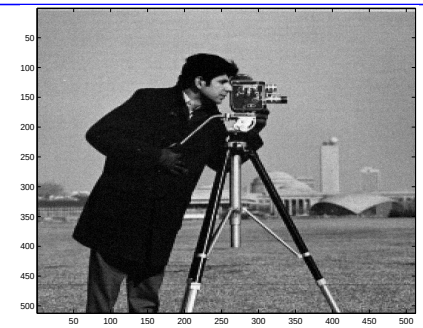
Recovery

Bundle CndG, 256×256 image (65,536 variables)
Recovery in 13 CndG iterations, CPU time 50.0 sec
Error removal: 98.5%, $\phi(y_{13})/\phi(0)$ <4.6e-5

True image



Blurred noisy image, 40% noise



Recovery

Bundle CndG, 512×512 image (262,144 variables)
Recovery in 18 CndG iterations, CPU time 370.3 sec
Error removal: 98.2%, $\phi(y_{18})/\phi(0)$ <1.3e-4

Platform: 2×3.40 GHz CPU with 16.0 GB RAM and 64-bit operating system

♠ **Note:** We used 15-element bundle, adding to it at step t the points $y_t = [x_t; r_t], D^+[x_t^+; 1]$ *and* $[\nabla f(x_t); TV(\nabla f(x_t))]$ and removing (up to) 3 old points according to "first in — first out." *Adding* $[\nabla f(x_t); TV(\nabla f(x_t))]$ *to the bundle dramatically accelerated the algorithm.*

How It Works:
Low Rank Matrix Completion

♠ **Problem:**

$$\mathrm{Opt} = \min_{x \in \mathbb{R}^{n \times n}} \left\{ 0.1\|x\| + \|x - a\|_F^2 \right\}$$

[• $\|\cdot\|$: nuclear norm • $\|\cdot\|_F$: Frobenius norm • $a = \bar{x} + \xi$
$\mathrm{Rank}(\bar{x}) \approx \sqrt{n}$, $\|\bar{x}\| \approx \sqrt{2n/\pi}$, $\|\xi\|_F \approx 0.1\|\bar{x}\|_F$ with i.i.d. Gaussian ξ_{ij}]

- Required relative inaccuracy 0.01

n	Method	CPU, sec	Iterations	Relative inaccuracy
128	CndG	4.5	42	<1.3e-6
	IPM	2675.0	31	<1.e-10
1024	CndG	44.2	31	<0.008
	IPM	not tested		
4096	CndG	1997.7	87	<0.01
	IPM	not tested		
8192†	CndG	1364.5	36	<0.01
	IPM	not tested		

† $\mathrm{Rank}(\bar{x}) = 32$

Platform: 2×3.40 GHz CPU with 16.0 GB RAM and 64-bit operating system

Note: CPU time in 8192×8192 example is less than needed to compute just 3 full svd's of a 8192×8192 matrix $\Rightarrow$ *The time taken by 36 steps of CndG is less than needed to perform just 3 steps of the simplest proximal algorithm, or just 2 steps of Nesterov's Fast Gradient method for Composite minimization!*

Conditional Gradients for Nonsmooth Convex Minimization

Source: Cox, B., Juditsky, A., Nemirovski, A. Dual subgradient algorithms for large-scale nonsmooth learning problems. *Mathematical Programming Series B* **148:1-2** (2014), 143-180.
`https://www2.isye.gatech.edu/~nemirovs/CoxJudNem.pdf`

♠ **Situation and goal:** *Given convex compact domain X represented by Linear Minimization Oracle, we want to solve convex program*

$$\mathrm{Opt} = \min_{x \in X} f(x)$$

where f is a Lipschitz continuous convex function.

Difficulty: Since X is given by LMO, it is problematic to use proximal algorithms; and since f can be nonsmooth, Conditional Gradient cannot be applied directly.

Remedy: Use *Fenchel-type representation*

$$f(x) = \max_{y \in Y} \left[x^T [Ay + a] - \phi(y) \right]$$
$$[\bullet\ Y\text{: convex set} \bullet\ \phi(\cdot) : Y \to \mathbb{R}\text{: convex function}]$$

Note: Fenchel-type representation is a special case of what we called *saddle point representation*

$$f(x) = \max_{y \in Y} \phi(x, y) \qquad [\phi : \text{ convex-concave}]$$

Note: Whenever $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a proper (i.e., with a nonempty domain) convex lower semicontinuous function, it admits *Fenchel* (a.k.a. *Legendre*) *representation*

$$f(x) = \sup_{y \in \mathbb{R}^n} \left[x^T y - f_*(y) \right]$$
$$\left[\begin{array}{c} f_*(y) = \sup_{x \in \mathbb{R}^n} \left[y^T x - f(x) \right]\text{: Fenchel dual of } f \\ f_* \text{ is convex proper lower semicontinuous, } [f_*]_* = f \end{array} \right]$$

$$f(x) = \sup_{y\in\mathbb{R}^n}\left[x^Ty - f_*(y)\right]$$

$$\left[\begin{array}{c} f_*(y) = \sup_{x\in\mathbb{R}^n}\left[y^Tx - f(x)\right]\text{: Fenchel dual of } f \\ f_* \text{ is convex proper lower semicontinuous along with } f\text{, and } [f_*]_* = f \end{array}\right]$$

Note: Fenchel dual "exists in the nature," but, aside of a handful of simple cases, is not available in closed form or in the form allowing for a cheap FO oracle.
In contrast, *Fenchel type* representations typically are readily available.
Example A. When $f(x) = \|Bx - b\|$, computing $f_*(y)$ reduces to solving a nontrivial convex problem

$$f_*(y) = \sup_x\left[y^Tx - \|Bx - b\|\right],$$

while Fenchel-type representation is immediate:

$$f(x) = \max_{y:\|y\|_*\le 1} y^T(Bx - b) = \max_{y:\|y\|_*\le 1}\left[x^T\underbrace{[B^Ty]}_{Ay} - \underbrace{b^Ty}_{\phi(y)}\right]$$

Example B. When summing up two convex functions with known Fenchel duals, the Fenchel dual of the sum is given by difficult to compute "inf-convolution":

$$[f + h]_*(y) = \inf_v\left[f_*(v) + h_*(y - v)\right]$$

In contrast, when summing up two convex functions with known Fenchel-type representations, a Fenchel-type representation of the sum is immediate:

$$f_i(x) = \sup_{y_i\in Y_i}\left[x^T[A_iy_i + a_i] - g_i(y_i)\right],\ 1 \le i \le m$$

$$\Rightarrow \sum_i f_i(x) = \sup_{y=[y_1;\ldots;y_m]\in\underbrace{Y_1\times\ldots\times Y_m}_{Y}}\left[\underbrace{\sum_i x^T[A_iy_i + a_i]}_{x^T[Ay+a]} - \underbrace{\sum_i g_i(y_i)}_{\phi(y)}\right]$$

$$\text{Opt} = \min_{x \in X} f(x) \qquad (P)$$

Assumption: *We know Fenchel-type representation of f:*

$$f(x) = \max_{y \in Y} \left[x^T [Ay + a] - \phi(y) \right]$$

where convex compact set Y admits a computation-friendly proximal setup, and ϕ is a Lipschitz continuous convex function given by First Order oracle.

$\Rightarrow$ Problem of interest (P) is the primal problem associated with the convex-concave saddle point problem

$$\text{Opt} = \min_{x \in X} \max_{y \in Y} \left[x^T [Ay + a] - \phi(y) \right].$$

The dual problem, in minimization form, is

$$[-\text{Opt} =] \quad \min_{y \in Y} \left[g(y) := -\min_{x \in X} x^T [Ay + a] + \phi(y) \right] \qquad (D)$$

and LMO for X induces First Order oracle for G: given $y \in Y$ and computing

$$x_y \in \operatorname*{Argmin}_{x \in X} x^T [Ay + a],$$

we have

$$\begin{array}{rcl} g(y) & = & -x_y^T [Ay + a] + \phi(y) \\ g'(y) & := & -A^T x_y + \phi'(y) \text{ is a subgradient of } g \text{ at } y \end{array}$$

$\Rightarrow$ *we can solve (D) by proximal-type First Order algorithm!*

$$\begin{array}{rcl} \mathsf{Opt} & = & \min_{x\in X}\left\{f(x)=\max\limits_{y\in Y}\left[x^T[Ay+a]-\phi(y)\right]\right\} \quad (P) \\ -\mathsf{Opt} & = & \min_{y\in Y}\left\{g(y)=-\min\limits_{x\in X}x^T[Ay+a]+\phi(y)\right\} \quad (D) \end{array}$$

Question: *How to recover a good approximate solution to (P) from information accumulated when solving (D)?*
Answer: *Use accuracy certificates!*

Accuracy Certificates

Let Z be a convex compact set, $F(\cdot)$ be a vector field on Z. Consider an N-step algorithm which operates with Z and F by generating sequence of *search points* $z_i \in Z$, $i \leq N$ along with the sequence $F(z_i)$, $i \leq N$, of the values of F along the search points.

- Collection $\mathcal{F} = \{z_i \in Z, F(z_i)\}_{i=1}^N$ is called the *the execution protocol* of the algorithm
- An *accuracy certificate* for execution protocol $\mathcal{F}$ is an N-dimensional vector λ of *nonnegative* weights λ_i summing up to 1
- The *resolution* of $(\mathcal{F}, \lambda)$ on Z is defined as

$$\mathrm{Res}(\mathcal{F}, \lambda | Z) = \max_{z \in Z} \left[\sum\nolimits_{i=1}^N \lambda_i \langle F(z_i), z_i - z \rangle \right]$$

Observation: *Every one of considered so far deterministic proximal First Order algorithms for convex minimization and convex-concave saddle point problems worked with some vector field F on a convex compact set Z and in N steps generated some execution protocol $\mathcal{F} = \{z_i \in Z, F(z_i)\}_{i=1}^N$ and accuracy certificate λ. When specifying approximate solution as*

$$z^N = \sum_i \lambda_i z_i,$$

the resolution $\mathrm{Res}(\mathcal{F}, \lambda | Z)$ *was an upper bound on inaccuracy of z^N resulting in efficiency estimates we got.*

Example: Subgradient/Mirror Descent for convex minimization problem $\min_{z\in Z} f(z)$ works with subgradient vector field $F(z) = f'(z)$ of the objective and ensures that

$$\forall z \in Z : \sum_{i=1}^N \gamma_i \langle F(z_i), z_i - z\rangle \leq \Theta + \sum_{i=1}^N \gamma_i^2 \|F(z_i)\|_*^2$$

[Θ : capacity of X w.r.t. DGF in question]

$$\Rightarrow \mathrm{Res}(\mathcal{F}, \lambda|Z) := \max_{z\in Z} \sum_i \lambda_i \langle F(z_i), z_i - z\rangle \leq \mathcal{R} := \frac{\Theta + \sum_{i=1}^N \gamma_i^2 \|F(z_i)\|_*^2}{\sum_{i=1}^N \gamma_i} \qquad (!)$$

$$\left[\lambda_i = \gamma_i / \sum_{j=1}^N \gamma_j\right]$$

Our efficiency estimate for SD/MD was yielded by (!) combined with the relation

$$f(\sum_i \lambda_i z_i) - f(z_*) \leq \sum_i \lambda_i [f(z_i) - f(z_*)] \leq \sum_i \lambda_i \langle F(z_i), z_i - z_*\rangle \leq \mathrm{Res}(\mathcal{F}, \lambda|Z). \qquad (!!)$$

where $z_* \in \mathrm{Argmin}_Z f$.

Note:

- SD/MD ensures (!) *independently of what is the origin of the vector filed F the method works with*
- (!!) holds independently of where the execution protocol with $F = f'$ and the accuracy certificate come from.

♠ *In retrospect, all we cared about when designing algorithms like SD, MD, or their bundle versions, or Mirror Prox, etc., was generating execution protocol and accuracy certificate with as small as possible guaranteed resolution.*

$$\begin{array}{rcll} \mathrm{Opt} & = & \min_{x\in X}\left\{f(x)=\max_{y\in Y}\left[x^T[Ay+a]-\phi(y)\right]\right\} & (P) \\ -\mathrm{Opt} & = & \min_{y\in Y}\left\{g(y)=-\min_{x\in X}x^T[Ay+a]+\phi(y)\right\} & (D) \end{array}$$

♠ **Fact:** *Assume we are solving (D) by First Order method producing in N steps execution protocol*

$$\mathcal{G}=\{y_i\in Y, g'(y_i)=-A^Tx_{y_i}+\phi'(y_i)\}_{i=1}^N$$
$$x_{y_i}\in \mathrm{Argmin}_{x\in X}\, x^T[Ay_i+a]$$

and accuracy certificate λ. Let us set

$$x^N=\sum_{i=1}^N\lambda_i x_{y_i},\ y^N=\sum_{i=1}^N\lambda_i y_i.$$

Then x^N is feasible for (P) and solves (P) within accuracy $\mathrm{Res}:=\mathrm{Res}(\mathcal{G},\lambda|Y)$.

Proof of Fact: Let $x \in X$ and $y \in Y$. We have

$$\text{Res} \geq \sum_i \lambda_i \langle -A^T x_{y_i} + \phi'(y_i), y_i - y \rangle = \sum_i \lambda_i \langle x_{y_i}, A[y - y_i] \rangle + \underbrace{\sum_i \lambda_i \langle \phi'(y_i), y_i - y \rangle}_{\geq \sum_i \lambda_i \phi(y_i) - \phi(y)}$$

$$\geq \sum_i \lambda_i \langle x_{y_i}, Ay + a \rangle - \sum_i \lambda_i \underbrace{\langle x_{y_i}, Ay_i + a \rangle}_{\leq \langle x, Ay_i + a \rangle} + \underbrace{\sum_i \lambda_i \phi(y_i)}_{\geq \phi(y^N)} - \phi(y)$$

$$\geq \sum_i \lambda_i \langle x_{y_i}, Ay + a \rangle - \sum_i \lambda_i \langle x, Ay + a \rangle + \phi(y^N) - \phi(y)$$

$$= \langle x^N, Ay + a \rangle - \langle x, Ay^N + a \rangle + \phi(y^N) - \phi(y)$$

$$\Rightarrow \langle x^N, Ay + a \rangle - \phi(y) \leq \text{Res} + \langle x, Ay^N + a \rangle - \phi(y^N)$$

The resulting inequality holds true for all $x \in X$ and $y \in Y$, implying that

$$f(x^N) = \max_{y \in Y} \left[\langle x^N, Ay + a \rangle - \phi(y) \right] \leq \text{Res} + \min_{x \in X} \left[\langle x, Ay^N + a \rangle - \phi(y^N) \right]$$

$$\leq \text{Res} + \max_{y \in Y} \min_{x \in X} \left[\langle x, Ay + a \rangle - \phi(y) \right] = \text{Res} + \text{Opt}.$$