Université catholique de Louvain

LINMA2471

Optimization models and methods II

Course notes

Students:
Ismail Ad'Oul, Noé Antoine, Antoine Aspeel, Nicolas Boutet, Sibo Cheng, Benjamin Chiem, Geoffrey Ciamarra, Antoine de Comité, Alexandre de Touzalin, Boris Dehem, François Delcourt, Renaud Dufays, Antoine Durviaux, Céline Gérard, Andine Havelange, Florimond Houssiau, Sébastien Lagae, Quynh Le, Adissa Laurent, Bruno Losseau, Joachim Lucas, Laura Motte, Pierre-Paul Mouchet, Guillaume Olikier, Caroline Sautelet, Vincent Schellekens & Leïla Van Keirsbilck

2015

Contents

I Notes **1**

A Linear and Convex modeling **2**
1. Definitions and motivation . . . 2
2. Standard forms . . . 4
   a. Linear case: re-writing objective function, variables and constraints . . . 4
   b. Standard form for linear models . . . 5
   c. Standard form for convex models . . . 6
   d. Transforming *any* problem into a convex problem . . . 7
   e. Approximate *any* convex problem by a linear problem . . . 9
3. Modelling Tricks . . . 10
   a. Monotonicity . . . 10
   b. Change of variables . . . 12
   c. Misleading/Deceptive appearances . . . 12
   d. Flexibility . . . 13
   e. Charnes and Cooper . . . 14
4. Convex Optimization: Theorems and properties . . . 15
   a. Convex sets . . . 15
      i. Definition and examples . . . 15
      ii. Properties . . . 15
   b. Convex functions . . . 16
      i. Definition and examples . . . 16
      ii. Properties . . . 16
   c. Properties of convex functions . . . 17
      i. Convexity and differential calculus . . . 17
      ii. Convexity and linear transformations . . . 17
      iii. Partial minimization . . . 18
      iv. Extended real valued functions . . . 19
      v. Composition and product . . . 19
   d. Advantage of convex problems . . . 20
   e. Variants of convex functions . . . 21

B First-order methods **23**
1. Gradient Method . . . 23
2. Step length selection . . . 23
   a. h_k that minimize $f(x_k - h_k \nabla f(x_k))$ . . . 23
   b. $h_k = \alpha$ . . . 24

c. h_k satisfies some "dynamic" conditions (e.g. Wolfe condition) 24
d. Analysis of the $h_k = \frac{1}{L}$ gradient method 26
3. Gradient method for unconstrained problems 28
a. Gradient method for functions of $C_L^{1,1}$ 28
b. Gradient method for functions of $F_L^{1,1}$ 29
4. Gradient method for constrained problems 31
a. Projected gradient method 32
b. Gradient mapping for constrained problems 33
c. Acceleration gradient [Nesterov 1983] 34

C Conic modeling and duality **36**
1. Linear and sublinear convergence 36
2. Conic optimization 36
a. Generalization 37
b. Solving a problem on different cones 38
c. Lorentz cone 38
d. Requirements for K 39
e. Equivalence with convex optimization 40
3. Duality 41
a. Exercise sessions on conic optimization 41
i. Exercise 2 42
ii. Exercise 4 42
iii. Exercise 6 : Conic duality 43
b. Example of duality 43
i. Primal problem 43
ii. Dual Problem 44
c. Theorems and properties of Dual 45
i. Necessary and sufficient condition 46
ii. Theorems 47

D Interior-point methods **49**

II Labs **50**
1. Introduction: AMPL 51

III Exercices **54**

Part I

Notes

Chapter A

Linear and Convex modeling

In this chapter we will see some important classes of optimization problems, and some techniques to re-write an optimization problem into another form.

1. Definitions and motivation

First of all, let's recall what the model of an optimization problem looks like.

Definition A.1. *A **general model** has the following form :*

$$\min_{x \in X \subseteq \mathbb{R}^n} f(x) \tag{A.1}$$

*where x are the **variables**, X is the **feasible set** (also called domain or feasible region) and f is the **objective function**.*

Note that the feasible set X is a subset of a *finite* dimensional space. Optimization within infinite-dimensional spaces are not covered in this course.

Let us next introduce two very important classes of models : the *linear* and *convex* models.

Definition A.2. *A model is called a **linear model** if :*

1. *The objective function is linear/affine*[1]*, that is, of the form $c^T x / c^T x + d$.*
2. *The feasible set X is a **polyhedron**. A polyhedron is an intersection of a finite*[2] *number of closed **half-spaces**. As a reminder, a half-space is the set of points that lie on one side of a hyperplane; in an algebraïc form : $\{x \in \mathbb{R}^n | a^T x \geq b\}$ or $\{x \in \mathbb{R}^n | a^T x \leq b\}$.*

Definition A.3. *A model is called a **convex model** if :*

[1] Note that the independent term of an affine function (d) can be easily dropped out, because it doesn't affect the optimal solution in any way. Therefore, every affine function can be replaced with a purely linear one.

[2] For example, a sphere is therefore *not* a polyhedron, because it is an intersection of an *infinite* number of closed half-planes.

1. *The objective function f is convex (see below).*
2. *The feasible set X is convex (see below).*

We still need to define what are convex functions and sets :

Definition A.4. *A set X is a* ***convex set*** *if it contains the segments between every pair of its points.*

Definition A.5. *A function f is a* ***convex function*** *if its* ***epigraph*** *is convex. The epigraph*[3] *of f is the set of points above (and including) the graph of f. Formally, we write this as : epi $f := \{(x,t)|t \geq f(x)\}$. This notion is illustrated on Figure A.1.*

For the definition of convex functions, we didn't used the concept of derivative, because we want our definition to be as general as possible. In other words, a non-differentiable function can be convex[4]. Note also that *every linear model is a particular case of a convex model.*

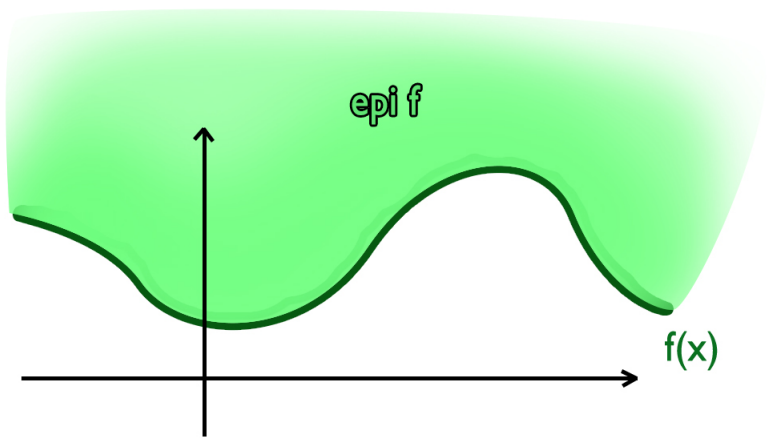


Figure A.1: Illustration of the epigraph of some one-dimensional (and non-convex) function. The epigraph of f is the region above the graph of f.

Now that we know the definitions of linear and convex models, what is the **motivation** to study such models? First, these models are useful to develop *efficient algorithms* with *guarantees* about the exactitude of the optimal solution and the speed of the algorithm.

Also, one could argue that studying linear/convex models is a restriction to the number of problems we will be able to solve, but it turns out that convex problems are *not so rare* in practice. Many problems are, or can be formulated as convex problems. Some problems can even be solved by using an equivalent convex problem : for example, the *branch and bound* algorithm transforms a discrete (and therefore non-convex) problem into a sequence of linear problems. One last -informal- reason to restrict ourselves to convex problem is that, for non-convex problems, there is nothing interesting we can really do or say.

[3]Note that if $f : \mathbb{R}^n \to \mathbb{R}$ then *epi* $f \subseteq \mathbb{R}^{n+1}$

[4]For example, the norm function defined by $f : \mathbb{R} \to \mathbb{R} : x \to |x|$ is convex

2. Standard forms

As the general formulation of convex and linear problems can be very hard to use in order to develop a theory about them, due (mostly) to the variety of constraint types, it is important to define **standard forms**. Standard forms define a unique, specific, formulation of these problems, that is much simpler than the general form.

a. Linear case: re-writing objective function, variables and constraints

Before we define the standard form, let us observe a number of transformations that can be applied to a linear problem without changing its solution (that is, the two problems will be equivalent).

Maximization and minimization In general, a maximization problem can easily be formulated as a minimization problem. Indeed, maximizing a function f is equivalent to minimizing its opposite $-f$. If the solution x is the same, the value of the objective function $-f(x)$ is simply the opposite of that of the original problem.

$$\boxed{\textbf{Max}} \xleftrightarrow{\text{max } f = \text{ - min (-}f\text{)}} \boxed{\textbf{Min}}$$

Non-negative variables The variables used in the linear standard form are non-negative, that is, they are free with the implicit constraint $x \geq 0$. It is possible to transform free variables in non-negative variables by a process known as **splitting**.

$$\boxed{\textbf{X non-negative}} \xleftarrow{\text{splitting}} \boxed{\textbf{X free}}$$

Splitting is done by, for every free variable x_i, adding two non-negative variables x_i^+ and x_i^-, defined by the relationship $x_i = x_i^+ - x_i^-$. Then, x_i is substitued by $x_i^+ - x_i^-$ everywhere in the problem.

However, this can be severely inefficient, since it adds a variable for every free variable. In a problem with n free variables, n additionnal variables are created, thus doubling the size of the problem. As this can be a major performance issue, it is important to do the splitting without creating too many variables.

A simple observation about the current method that can be made is that, if we assume x_i has a unique value in the solution, the values of x_i^+ and x_i^- are not uniquely defined (in some sense, there is one too many degree of freedom). By exploiting this notion, another formulation can be proposed: if a problem has n free variables x_i, we subtitute these by $x_i^+ - x^-$, where x_i^+ and x^- are non-negative variable, and x^- is **the same for all the variables**. This method is better since it only creates **one additional** variable!

Why does this work? If and whenever a solution x^* is found, the value of $x^{-,*}$ will be at most equal to the smaller (i.e. most negative) x_i^*. Then, by definition, $x_i^{+,*} = x_i^* + x^{-,*}$ (with $x^{-,*} \geq 0$), and since $x_i^* \geq -x^{-,*}$, we get that $x_i^{+,*} \geq 0$, and so $x_i^{+,*}$ is indeed non-negative.

As an example, suppose a linear problem with 3 variables, all of them free, has as unique solution $(x_1^*, x_2^*, x_3^*) = (-3, 7, -10)$. A solution, in term of non-negative variables, is thus $(x_1^{+,*}, x_2^{+,*}, x_3^{+,*}, x^{-,*}) = (7, 17, 0, -10)$.

The reverse is also possible, although not really interesting.

$$\boxed{\textbf{X non-negative}} \longrightarrow \begin{cases} \textbf{Constraint: } X \geq 0 \\ \textbf{X free} \end{cases}$$

Equalities and inequalities After treating the objective and the variables, it is important to treat the constraints (which offer the widest range of varieties). Firstly, turning equalities into inequalities is very straightforward:

$$\boxed{\textbf{Equality: } f = g} \longrightarrow \boxed{\textbf{Two inequalities: } f \geq g \textbf{ and } f \leq g}$$

The opposite is both more useful and more subtle:

$$\boxed{\begin{array}{l}\textbf{Equality: } f + s = g \\ \textbf{and slack variable } s\end{array}} \xleftarrow{\text{Slack Variable}} \boxed{\textbf{Inequality } f \geq g}$$

To do this, we must introduce the concept of **slack variable** (*variable d'écart* in French). A slack variable is a non-negative variable added on the greater side of the inequality to make it an equality. Basically, its value is $f - g$, the slack between f and g, representing the *margin* before the constraint becomes an equality.

One last case to be treated is the **lower bound constraint**. This can of course be treated as a constraint, but is redundant with the nonnegativity of the variables in the standard form. A fairly simple solution is to subtitute the variable X with $X - l$, where l is the lower bound.

$$\boxed{\textbf{Lower Bound } X \geq l} \xrightarrow{\text{Substitution}} \boxed{\begin{array}{l}\textbf{non-negative} \\ X' \textbf{ : } X = l + X'\end{array}}$$

b. Standard form for linear models

The standard form of a **linear optimization problem** is:

$$\boxed{\begin{array}{ll} \min_X & c^T X \\ & AX = b \\ & X \geq 0 \quad (\Longleftrightarrow X \text{non-negative}) \end{array}}$$

If the problem has n variables and m constraints, then $c \in \mathbb{R}^{n \times 1}$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^{m \times 1}$. These constitute the only data needed to uniquely define the problem.

The transformations exposed in the previous section illustrate the fact that it is possible to turn any linear optimization problem in the standard form.

c. Standard form for convex models

There is **no known standard form** for convex optimization problems. This can be understood in the sense that the objective function can be very general, as well as the set X. Since, by definition, the set X is uncountable, it is very hard to represent it in a way that can be, for example, treated by a computer. However, it is possible, under certain assumptions, to represent the set X in a purely functional way, as a set of inequalities involving functions.

Most often, the set X is defined as a set of constraints. Let's suppose that there exists a set of functions g_i, $i \in \{1 \geq \cdots \geq m\}$, h_j, $j \in \{1 \geq \cdots \geq l\}$, such that:

$$X = \{x \in \mathbb{R}^n | g_1(x) \leq 0, \ldots, g_m(x) \leq 0 \text{ and } h_1(x) = 0, \ldots, h_l(x) = 0\}$$

This form is general, since the $g_i(x) \geq 0$ constraint is equivalent to $-g_i(x) \leq 0$. The one exception lies in the fact that **strict inequalities** cannot be treated. But, as is explained later, this is not a real issue.

In general, this does not suffice to guarantee that X is convex. The following conditions are **sufficient** to ensure the convexity of the set X:

- the g_i functions are **convex** (this results from the choice that g_i should be smaller than zero: if the opposite were chosen, then the functions should be concave)
- the h_j functions are **linear**: it's tempting to say that convexity is enough, but it is not the case. For example, the function $h : \mathbb{R}^2 \to \mathbb{R} : (x, y) \to x^2 + y^2 - 1$ defined as ensemble X the circle (and not the discus!) of radius 1, which is obviously not convex[5].

In some cases, the constraints $h_j(x) = 0$ can be relaxed to inequalities $h_j(x) = 0$ (Shouldn't it be an inequality?), thus relaxing the linearity constraint on h_j. One of these cases, which will be developed in the next section, is when the objective function is linear.

A note on $\neq$ The standard forms developed in this section do not allow for strict inequalities to be considered. This is because strict inequalities tend to make solutions *disappear*: in linear optimization, solutions are always located on the boundary of a closed polyhedron. By making this polyhedron open (with strict inequalities), the solutions disappear, as there is not admissible point with a minimal value (it is always possible to get *closer* to the boundary, thus reducing the objective function).

A solution to this is to treat strict inequalities as non-strict ones by introducing a *tolerance* $\epsilon > 0$, that describes how close to the open boundary the solutions can lie. The constraint $f > g$ then becomes $f \geq g + \epsilon$. The choice of the tolerance depends on the context of the problem (and is to be discussed with the client, for example).

[5] In fact, each segment binding two points of the circle doesn't include any other point of the circle that its extremity...

d. Transforming *any* problem into a convex problem

We can turn any optimization problem into a convex problem by following two "easy" steps. First, *the objective function can be made linear* by adding a new variable. Then, *the constraints can be made convex* by an operation called taking the convex hull. Let us see these operations in detail.

First step : making the objective function linear Let us assume the following (general) model :

$$\min_{x \in X \subseteq \mathbb{R}^n} f(x)$$

The trick is to re-write this problem introducing a new variable t that is greater or equal[6] to $f(x)$. We now have :

$$\min_{x \in \mathbb{R}^n,\, t \in \mathbb{R}} t \text{ with } x \in X,\ (x,t) \in \mathit{epi}\ f$$

The new objective function is just t and is clearly linear, but the domain is now more complicated. In other words, we have traded simplicity in the objective function (wich is a good thing) by adding complexity in the domain (wich is usually already complex anyway, so it isn't that bad). Note that if the original problem was convex, the new problem is still convex (because it means *epi* f is convex).

Second step : making the constraints convex The feasible set can be made convex by taking the convex hull of the set.

Definition A.6. *The* ***convex hull*** *of a set X (denoted by conv X) is the* ***smallest*** *convex set containing X. The smallest set means : the intersection of all possible convex sets containing X.*

> **Example A.1.** *If we have a simple (non-convex) set containing two points in space, the convex hull of this set is the segment between those two points. This example is illustrated figure A.2. This gives us an (infeasible in practice, see "the catch" to make every problem convex, page 8) algorithm to take the convex hull of any set : just take every pair of points in the set and add the segment between those two points!*

Taking the convex hull of the feasible set X obviously gives us a new optimization problem. What are the optimal solutions of this new problem?

[6] It is intuitively more appealing to impose that $t = f(x)$, so we still have "exactly the same" objective function. But since equality constraints are harder to handle, we prefer the inequality $t \geq f(x)$. So instead of minimizing $f(x)$, we minimize *a higher bound* to $f(x)$, wich is equivalent.

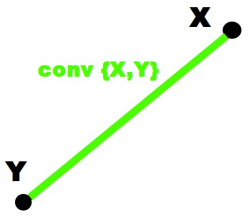


Figure A.2: Illustration of the convex hull of a set X. Note that the convex hull also includes the original set, wich is not very well represented on the figure.

Theorem A.1. *Any optimal solution x^* to the original problem :*

$$\min_{x \in X} c^T x$$

is also optimal for the new problem :

$$\min_{x \in conv\ X} c^T x$$

This also means that the *optimal values* of the two problems are the same. But because $X \subseteq conv\ X$, some optimal solutions of the new problem won't be in the original set X. So, in order to find the original optimal solutions, once we have solved the convex problem, we should always reject the optimal solutions of the new problems that aren't in X. Mathematically speaking :

$$\{x^*_{original\ problem}\} = \{x^*_{new\ convex\ problem}\} \cap X$$

So, to summarize, we can make *every problem in the world* convex, and we can (not yet, but after finishing this course) solve convex problems with good algorithms! This implies that basically any optimization problem can be solved easily! It seems too good to be true, and it is, since there is a **catch**. Although the definition of complex hull is rather simple, taking the convex hull of a general (that is, a little sophisticated) set is a very difficult operation.

So, in general, this approach is useless. There are specific cases, however, where this can be very useful!

The first, is a special case of linear optimization with **or** constraints. In such a problem, the admissible set is the union of (possibly disjoint) polyhedra. Computing the convex hull of the union of polyhedra can be done very efficiently, if the vertices are known, since the only task is to compute the vertices that will stay extreme in the union. The figure A.3 illustrates this example.

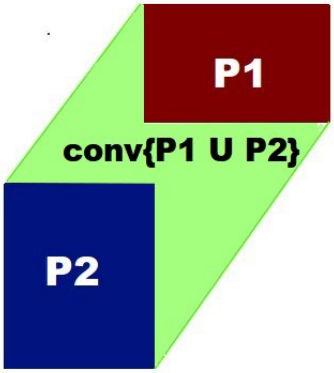


Figure A.3: Two (simple) disjoint polyhedron and the convex hull (in green) of the union.

An interesting point to be raised here, is that such a problem could also be solved by computing the solution on every polyhedron, then choosing the best one. For small problems, this is of course valid, but for high number of dimensions, the cost of solving the problem on each polyhedron becomes prohibitive, while computing the convex hull remains a relatively cheap operation.

The second case where taking the convex hull is useful is for (some) discrete models. Indeed:

$$\min_{x_i \in \{-1,1\}} c^T x \iff \begin{array}{l} min_x c^T x \\ -1 \leq x_i \leq 1 \end{array}$$

The convex hull of $\{-1, 1\}^n$ is $[-1, 1]^n$, and so both problems have the same solutions. The flow problem with integers (as seen in LINMA1702), which retains integer only solutions with relaxation, is an example of taking the convex hull of discrete problems without changing the nature of solutions.

e. Approximate *any* convex problem by a linear problem

It is possible to approximate convex problems by linear ones. Since the objective can always be converted to a linear function by adding a variable, the only work to be done concerns the constraints. Basically, the idea is to approximate the (closed[7]) set X by a finite intersection of half-spaces (thus, linear constraints).

The way to do this is to use **projections** of points on the convex space. The projection of a point u on a set X is defined as the point $u_p \in X$ that minimizes the distance between itself and u.

Theorem A.2. ***Uniqueness and existence of projection.*** *Let X be a* ***closed, non-empty, convex*** *set in $\mathbb{R}^n$, the projection of any exterior point on X exists and is unique.*

[7]Why not open? Because an open set corresponds to strict inequalities, which cannot and will not be treated with the common optimization tools (see "A note on $\neq$", page 6)

It is easy to see that the closeness and non-emptiness guarantee the existence of such projection. The unicity, however, is ensured by the convexity of the set. As an example, it is easy to see that a non-convex set such as the unit circle as an infinite number of projections of the origin.

With this concept of projection, we can introduce the **separation property**: for every exterior point u of a convex closed set X, there exists a plane that *separates* u from X, that is, such that every element of X is on *one side* of the plane, while u is on the *other side*. This results directly from the uniqueness and existence of the projection of u on X, although this was not demonstrated in class. Intuitively, this separation plane can be built perpendicular to the segment joining u and its projection, without intersecting the convex plane.

So, to approximate a convex set by a linear one, the following method should be applied: for every point that should not be in X, create a separation plane. Such plane defines a half-space containing all of X. The intersection of all the half-spaces obtained this way forms a polyhedron containing X.

Moreover, it is interesting to note that an infinite number of points will create the convex set itself! This yields a new definition for a convex set: a convex set can always be written as the infinite intersection of half-spaces. This definition also proves immediately that every polyhedra is convex.

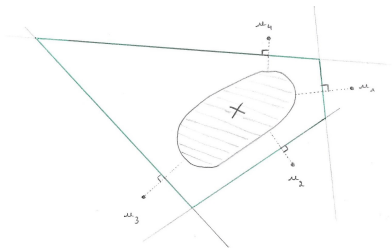


Figure A.4: Approximation of a non-linear problem by a linear one.

3. Modelling Tricks

After studying the standard form of some optimization problems, we will see some modelling tricks which permit to transform a non-linear or non-convex problem into a linear or convex optimization problem.

a. Monotonicity

Definition A.7. *A monotonic function over an interval is a function that is either increasing or decreasing over this interval.*

The transformation that is described in the following lines uses monotonic functions to turn an optimization problem into a simpler one. These operations do not change the problem but can change the solution and the optimal value of the objective of the problem. Let's see some examples.

Example A.2. min $\|x\|_2$ *with* $x \in X$ *is equivalent to* min $\|x\|_2^2$ *with* $x \in X$. *In fact,* $z \to z^2$ *is a monotonic function (increasing in this case) over* $\mathbb{R}$. *By solving this problem, we will find the same optimal solution than the first model but the value of the objective function will be different.*

Example A.3. *These functions are monotonic and can be used like in the previous example to simplify the problem:*

. $z \to e^z$

. $z \to \log(z),\ (z > 0)$

. $z \to -\frac{1}{z},\ (z > 0)$ *for example, we can change* min $\frac{1}{\|x\|_2}$ *to* min $-\|x\|_2$

We not only use this trick to modify the objective function, but also some constraints. For example:

$$f(x) \leq b \Leftrightarrow e^{f(x)} \leq e^b$$

Example A.4. ***Advertisement for bank account:*** *This example allows us to look at the effects of monotonicity in real life optimization problems. We put money into an account where we can't take our money back sooner than 5 years after. The bank guarantees that we have a high percentage when we take back our money after 5 years and assures the three rates over 3 years given on Figure A.5.*
What is the worst global rate compatible with this? Therefore, we are looking for the cumulative effect of the 5 rates (given over a year).

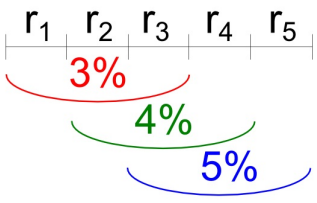


Figure A.5: Rates

To solve this problem, we introduce the variables r_i *corresponding to the rate per year where* $i = 1, ..., 5$. *We don't want a rate which is negative or null so we have the following optimization problem:*

$$\min_{r_i > 0} r_1 r_2 r_3 r_4 r_5$$

under the constraints:

$$r_1 r_2 r_3 \geq 1.03$$

$$r_2 r_3 r_4 \geq 1.04$$

$$r_3 r_4 r_5 \geq 1.05$$

This problem is not linear. We can transform it into a linear problem if we use the logarithm function. Indeed, let $y_i = \log(r_i)\ \forall i = 1, ..., 5$. *We can do this because the logarithm is a monotonic increasing function. The problem can be written as:*

$$\min\ y_1 + y_2 + y_3 + y_4 + y_5$$

under the constraints:

$$y_1 + y_2 + y_3 \geq \log(1.03)$$

$$y_2 + y_3 + y_4 \geq \log(1.04)$$

$$y_3 + y_4 + y_5 \geq \log(1.05)$$

This problem is now linear (and thus convex).

***Remark**: If we don't make the change of variable $y_i = \log(r_i)$ $\forall i = 1, ..., 5$, the variables r_i appear in* log *so the problem isn't linear.*

b. Change of variables

We use change of variables in order to transform the problem into a linear or convex problem. For example, if every variable appears in a logarithm, then we can use the change of variable to remove the logarithm.
Remark: every appearance of the variables needs to match the change of variables!

For example, signomials[8] can be converted into convex functions thanks of this trick. Let's consider the signomial $\frac{x_1 x_2^2}{x_3^{\frac{1}{2}}}$. Let $x_i = e^{y_i}$, we obtain the following expression by change of variable $e^{y_1 + 2y_2 - \frac{1}{2}y_3}$. Moreover, the exponential of a linear function is convex and the change of variables conserves the convexity (see later Should add correct reference).

c. Misleading/Deceptive appearances

In this part, we are looking for a polynomial $p(x)$, $x \in \mathbb{R}$ of degree D which fulfils some characteristic (see Figure A.6):

. For $x \in [0; f_1]$, $p(x) \geq 3$

. For $x \in [f_2; f_3]$, $p(x) \leq 0.5$

Let $p(x) = \sum_{i=0}^{D} a_i x^i$. On the Figure A.6, we observe that the polynomial will not be linear or convex. We then have the impression that the optimization problem is not linear. But it is not the case, given that the variables are not the x_i but the a_i. Furthermore, the constraints (2 examples of constraints are given below) are linear.

$$\sum_{i=0}^{D} a_i 50^i \geq 3 \Leftrightarrow p(50) \geq 3$$

$$\sum_{i=0}^{D} a_i 100^i = 1 \Leftrightarrow p(100) = 1$$

[8] A signomial is an algebraic function of one or more variables of the form:
$f(x_1, x_2, ...x_n) = \sum_i (c_i \prod_j x_j^{a_{ij}})$

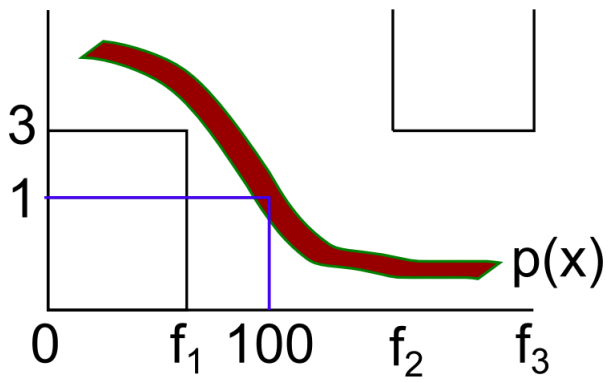


Figure A.6: Polynomial $p(x)$

d. Flexibility

The flexibility of an optimization problem is its capacity to be solved in different ways depending on which criterion we really want to optimize. Let us consider an approximation problem. We have a set of points (x_i, y_i) in $\mathbb{R}^2$, and we want to find the function (which is often a polynomial) which approximates *at best* these points. In other words, we want to find a function f that minimizes the errors ϵ_i at the approximation points:

$$\epsilon_i = |y_i - f(x_i)|$$

The question now is to choose the way to minimize errors ϵ_i. Indeed, we could choose to minimize overall errors with a least square criterion:

$$\min ||\epsilon||_2 = \sqrt{\sum_i \epsilon_i^2}$$

Another choice would be to minimize the sum of the errors:

$$\min ||\epsilon||_1 = \sum_i |\epsilon_i|$$

Finally, a last (often used) way to minimize errors is to minimize the maximum error:

$$\min \max_i \epsilon_i$$

All these considerations prove that, depending on the criterion we choose to optimize (i.e. depending on the context), the result could be different. For example, the problem

$$\begin{array}{ccc} \max & \text{safety} & \\ \text{cost} & \leq & m \end{array}$$

will not lead to the same solution than

$$\begin{array}{ccc} \min & cost & \\ safety & \geq & b \end{array}$$

e. Charnes and Cooper

Consider the non-linear problem which is a division of two linear expression:

$$\min \frac{c^T x + d}{f^T x + g}$$

under the constraints:

$$Ax \leq b$$

In this case, taking the logarithm of the objective function will not change anything: we will have the same problem.

We make the hypothesis that $f^T x + g > 0 \ \forall x$ such that $Ax \leq b$. If not, we have the solution $-\infty$ which is not an interesting solution.

To solve this problem and make it linear, we are going to homogenize the objective function. Let $x = \frac{y}{t}$ with $y \in \mathbb{R}^n$ and $t > 0 \in \mathbb{R}$. If we take $t = 1$ then we get back to the original problem. So, one solution in x correspond to several solutions in (y, t) (for example: $(x, 1),\ (2x, 2),\ ...\ (\lambda x, \lambda)$).
We can write the problem as:

$$\min \frac{\frac{c^T y}{t} + d}{\frac{f^T y}{t} + g}$$

with:

$$A\frac{y}{t} \leq b$$

By simplification, we obtain the following problem:

$$\min \frac{c^T y + dt}{f^T y + gt}$$

with:

$$Ay \leq bt$$

Now, we notice that the objective function's numerator and denominator are linear as the constraint. This problem has a property called homogeneity. If you take any solution, you can multiply any component by the same constant and nothing changes. Mathematically: (y, t) solution $\Rightarrow (\lambda y, \lambda t)$ solution $\forall \lambda \neq 0$

We are going to choose solutions satisfying: $f^T y + gt = 1$. We can do this using the property of homogeneity. This step results in selecting one solution among the collection of solutions multiple of each other. The objective function gets simpler and linear and we add one constraint. So, we have the following linear optimization problem:

$$\min c^T y + dt$$

with:

$$Ay - bt \leq 0$$

$$f^T y + gt = 1$$
$$t \geq 0$$

We compute y^* and t^* from this problem and take $x^* = \frac{y^*}{t^*}$ the solution of the original problem.
Remark: If we have $t = 0$ at the optimum then the problem is unbounded and $x^* \to \infty$ (see Example A.5).

Example A.5.

$$\min \frac{1}{x}$$

with:

$$x \geq 1$$

This problem have an optimal value of 0 so the solution $x^* \to +\infty$.

4. Convex Optimization: Theorems and properties

a. Convex sets

i. Definition and examples

Definition A.8. *A set X is convex if and only if*

$$x,\ y \in X \Rightarrow \lambda x + (1-\lambda)y \in X \quad \forall\, 0 \leq \lambda \leq 1$$

Thus, a set is convex if and only if it contains all the segments joining any pair of its points.

Example A.6. *Here are several examples:*

- $\mathbb{R}^n$, $\mathbb{R}^n_+$, $\emptyset$
- *Hyperplans* ($\{x|\ b^T x = \beta\}$)
- *Open or closed half-spaces* ($\{x|\ b^T x < \beta\}$ *and* $\{x|\ b^T x \leq \beta\}$)
- *Open and closed balls* ($\{x|\ \|x-a\| < r\}$ *and* $\{x|\ \|x-a\| \leq r\}$)

ii. Properties

Property A.1. *Given a collection of convex sets $\{C_i\}_{i\in I} \subseteq \mathbb{R}^n$ (I can be arbitrary), then $\bigcap_{i\in I} C_i$ is convex too.*

It follows that polyhedrons are convex because they are intersection of half-spaces.

Property A.2. *Given a collection of convex sets $C_1,\ C_2,\ C_3,\ ....\ C_n$, their cartesian product $C_1 \times C_2 \times C_3 \times ... \times C_n$ is convex too.*

Property A.3. *If $X \subseteq \mathbb{R}^n$ and $X \subseteq \mathbb{R}^n$ are convex then the Minkowski sum of X and Y, $X + Y = \{x + y \mid x \in X \text{ and } y \in Y\}$ is convex too.*

Remark: The union of convex sets is not always convex!

b. Convex functions

i. Definition and examples

Definition A.9. *A function f with domain D is a convex function of and only if*

$$D \text{ is convex and}$$
$$x, y \in D \Rightarrow f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \quad \forall\, 0 \leq \lambda \leq 1$$

Example A.7. *Here are several examples:*

- *Linear and affine functions are convex ($x \to \alpha x$, $x \to b^T x$ and $x \to b^T x + \alpha$)*
- *The norm function and the square of the norm function are convex functions ($x \to \|x\|$ and $x \to \|x\|^2$)*
- *Quadratic forms ($x \to x^T Q x$) are convex functions when the matrix $Q \in \mathbb{R}^{n \times n}$ is semi positive definite*
- *The functions $x \to e^x$, $x \to log(x)$ and $x \to |x|^p$ $(1 \leq p)$ are convex*

Definition A.10. *A function f is concave $\Leftrightarrow -f$ is convex.*

Remark: Linear and affine functions are convex and concave.

ii. Properties

To know whether a function is convex or not, we have to transform it into its epigraph and check if it is convex or not. But there are some useful properties of convex functions that we can use to spare time.

Property A.4. *If f is a convex function and $c \in \mathbb{R}_0^+$, then cf is convex.*

Property A.5. *If f and g are convex functions, then $f + g$ is convex.*

Property A.6. *Given a collection of convex functions $\{f_i\}_{i \in I} : \mathbb{R}^n \to \mathbb{R}$, $\sup\limits_{i \in I} f_i$ is convex too.*
With $\left[\sup\limits_{i \in I} f_i\right](x) = \sup\limits_{i \in I} f_i(x)$

Property A.7. *Given $f(x,s)$ (with $x \in \mathbb{R}^n$ and $s \in \mathbb{R}$, a parameter) such that $x \to f(x,s)$ is convex for any s,*

$$\int_{s \in S} f(x,s)ds \;\; is \;\; convex$$

c. Properties of convex functions

i. Convexity and differential calculus

Property A.8. *Let f be a differentiable function of which the domain D is open. f is convex if and only if D is convex and*

$$\forall x, y \; \in D, \; f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

This property means that, if f is convex, it will be above all its Taylor approximations of first order. This signifies that at any point, the tangent of the function is under the function. This is useful to make a piecewise approximation of f by linear functions (an example of such an approximation is shown on Figure A.7). In order to obtain such an approximation, we have to choose n points at which we calculate the tangent of the function f and then take the maximum of the n tangents (in the example, $n = 3$).
Besides, the value of the approximation is lower than the real value on any point. It allows us to obtain lower bounds.

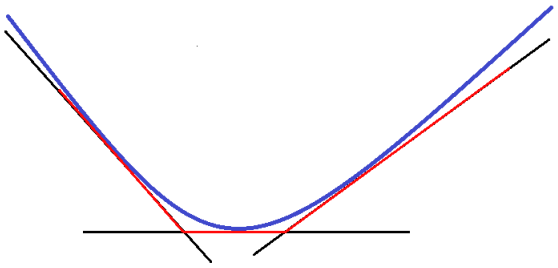


Figure A.7: Illustration of the piecewise linear approximation of a convex function

Property A.9. *Let f be a twice differentiable function of which the domain D is open. f is convex if and only if D is convex and*

$$\forall x \in D, \; \nabla^2 f(x) \geq 0$$

ii. Convexity and linear transformations

Linear transformations preserve convexity. Indeed,

Property A.10. *If $S \subseteq \mathbb{R}^n$ is convex and $\Phi : \; \mathbb{R}^n \to \mathbb{R}^m : \; x \to Ax + b$ is a linear transformation, then the image of S by Φ,*

$$\Phi(S) = \{\Phi(x) |\, x \in S\}$$

is convex too.

Property A.11. *If* $\Phi : \mathbb{R}^n \to \mathbb{R}^m : x \to Ax + b$ *is a linear transformation and* $f : x \to f(x)$ *is a convex function, then the composition*

$$f \circ \Phi = f(\Phi(x)) = f(Ax + b)$$

is convex too.

Property A.12. *If* $S \subseteq \mathbb{R}^n$ *is convex and* $\Theta : \mathbb{R}^m \to \mathbb{R}^n : x \to ax + b$ *is a linear transformation, then the image of* S *by the inverse of* Θ*,*

$$\Theta^{-1}(S) = \{x|\ \Theta(x) \in S\}$$

is convex too.

Property A.13. *Given a convex and affine transformation* $x \mapsto Ax + b$*, the composition* $x \mapsto f(Ax + b)$ *is also convex.*

Example A.8. e^{2x-y+z} *is convex because the exponential is convex and* $2x - y + z$ *is a linear transformation of* x, y *and* z*.*

Example A.9. ***Convex functions***
Any norm $x \mapsto ||x||$ *is convex, thus the distance* $||x - y||$ *between two points* x *and* y *is convex because* $x - y$ *is a linear transformation.*
The maximum distance between a set S *and a point* x *is a convex function. Indeed, taking the maximum between a point and a set requires to take the maximum of all the distances between the point and any point in the set (distance between two points is a convex function):* $f_{S,max} = \max_{s \in S}\{||x - s||\}$

iii. Partial minimization

Property A.14. *(Partial minimization) If the function* $f : (x, y) \mapsto f(x, y)$ *is convex, then* $f_x(y) = \inf_x f(x, y)$ *is convex.*

Example A.10. *If a set* S *is convex, then the minimum distance function between a point* x *and the set* S *is convex. Indeed, one can write the function as follow:*

$$f(x, s) = ||x - s||$$

Since this is a norm, f *is convex. Since the restriction of a convex function stays convex as long as the feasible region stays convex and* S *is a convex set, property A.14 gives that:*

$$f_S(x) = \inf_S f(x, s)$$

is a convex function.

Remark: Property A.14 is a one side property. A counter-example for the reverse side is given by:

$$f_x(y) + \sqrt{||x||}$$

iv. Extended real valued functions

Most of theorems to prove the convexity of a function require the convexity of the domain. However, it is possible to extend a function to tackle this problem.

Example A.11. *Let's take the function* $f : \mathbb{R}_+ \mapsto \mathbb{R} : x \mapsto \frac{1}{x}$ *and extend it such that its domain becomes the whole real line. One consider:*

$$f_e : \mathbb{R} \mapsto \mathbb{R} \cup \{+\infty\} : x \mapsto \begin{cases} \frac{1}{x} \ \textit{if } x > 0 \\ +\infty \ \textit{elsewhere} \end{cases}$$

One can see that the extended function is convex over the whole real line. The epigraph definition still holds since there isn't any point above $+\infty$.

v. Composition and product

Property A.15. *If g is a convex function and f is a convex, increasing and one-dimensional function then the composition function* $h \circ g : x \mapsto h(g(x))$ *is also convex.*

Proof Let's prove this proposition for a simple case. We assume that f and g are both one-dimensional functions and that $f, g \in \mathcal{C}^2$. The general case requires a more difficult proof.
Since f and g are 2 times differentiable, one has:

$$[h(g(x))]'' = [h'(g(x))g'(x)]' = \underbrace{h''(x)(g'(x))^2}_{\text{A}} + \underbrace{h'(g(x))g''(x)}_{\text{B}}$$

Since h is convex, its second derivative is positive and given that a square is positive, one has that A is positive. Furthermore, since g is also convex and h is increasing, one also has that B is positive. One conclude that the second derivative of $h \circ g$ is positive and thus, the function $h \circ g$ is convex.

QED

Remark: Sometimes we need to square a value but also to keep convexity (for example, we don't care about negative deviations on a budget) Should be better expressed. However the traditional square function is not convex on the real line. Let's introduce a restricted square function as follows:

$$f : x \mapsto (x_+)^2 = (\frac{x + |x|}{2})^2$$

We easily see (Figure A.8) that this restricted square function is convex.

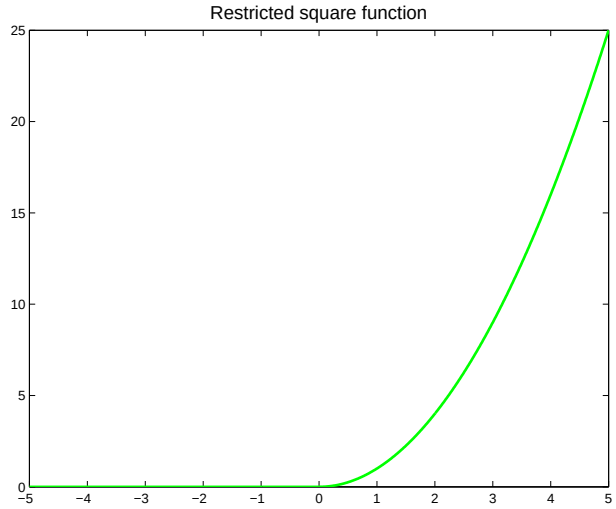


Figure A.8: Restricted square function

Example A.12. *The function* $\left[\log(x+y)_+\right]^2$ *is convex. Indeed, the restricted square and* $-\log$ *are convex functions. Therefore, their composition is convex. Since* $x+y$ *is a linear transformation, it preserves convexity and the whole function is convex.*

Property A.16. *If* f *and* g *are both convex, positive and increasing then their product is convex.*

Proof Again, one proves it in the simple differentiable, one-dimensional case. One has:

$$(fg)'' = \left[f'g + fg'\right]' = f''g + 2f'g' + fg''$$

The result follows immediately since by assumptions one has $f, g, f', g', f'', g'' \geq 0$.

QED

Remark: The previous proof tends to indicate variants of Property A.16. One can see that if f and g are both concave, decreasing and negative then the proposition still holds.

d. Advantage of convex problems

Property A.17. *Let's recall that* $\min_{x\in X} f(x)$ *is convex if* f *is convex,* X *is convex and we are looking for a minima. We study the properties of a convex problem:*

MODEL	*METHODS*
- Local minima are also global *- The set of optimal solutions is convex* *- Using duality we can get guarantees*	*- Methods which only work on convex problems: first order, second order...*

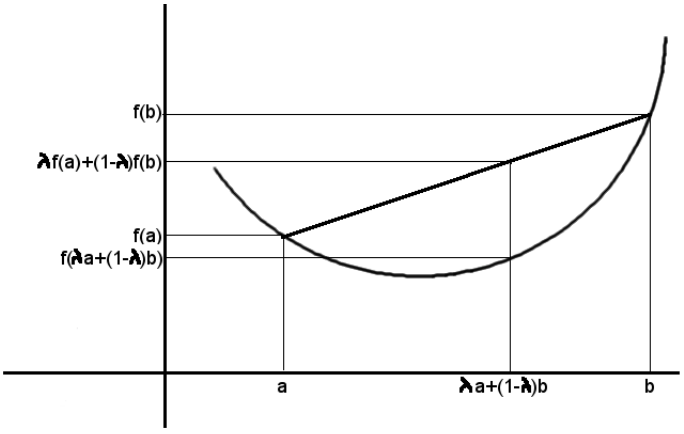
e. Variants of convex functions

It's a difficult thing to know if a problem has a unique solution. There is one class of problems with only one solution: minimization of strictly convex functions.

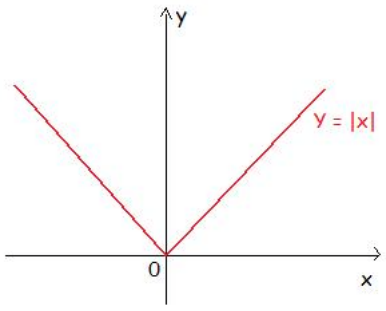
Definition A.11. *Strict convexity: f is strictly convex if and only if*

- *The domain is convex*
- $f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y)\ \forall x, y \in \text{Dom}\ ,\ \forall \lambda \in]0,1[$

Example A.13. *On the following graph, we can observe that the function is strictly convex.*



Example A.14. *The absolute function is not strictly convex. In fact, when we have a flat part in the graph, it can not be strictly convex.*



Example A.15. *The function $x \mapsto ||x||_2 = \sqrt{\sum_i x_i^2}$ is convex but not strictly convex.*

Property A.18. *If we have $\min f(x)_{x \in X}$ with f strictly convex and X a convex set then the problem admit at most one solution.*

Property A.19. *If $f \in C^2$ and $\nabla^2 f(x) > 0$ then f is strictly convex. ($\lambda_i > 0\ \forall i$)*

Property A.20. *If f is convex, then $f + ||x||_2^2$ is strictly convex.*

Proof Assume $f \in C_2$
then

$$\nabla^2(f + ||x||^2) = \nabla^2 f + \nabla^2 ||x||_2^2 = \nabla^2 f + 2I$$

where $||x||_2^2 = \sum_i x_i^2$ and $\lambda_i \geq 0$.

QED

Property A.21.

$$(1)\ \lambda \textit{ is an eigenvalue of } M$$

$$\Leftrightarrow$$

$$(2)\ \lambda + \Delta \textit{ is an eigenvalue of } M + \Delta I$$

for any $\Delta \in \mathbb{R}$ *and* $M \in \mathbb{R}^{n \times n}$ *symmetric.*

Proof

$$(1)\ \exists v \ \ Mv = \lambda v$$

$$(2)\ \exists v \ \ (M + \Delta I)v = (\lambda + \Delta)v$$

QED

Remark: We can have the same propositions and the same proof while adding $\mu > 0$ anywhere: $f + \mu||x||^2$. It is a regularization to make it strictly convex.
Remark: There are functions that have no derivative and are strictly convex.

Chapter B

First-order methods

We've been investing in convex models and trying to find the best formulation of optimization problems. We now explore different methods that allow us to solve those problems and study their properties. In this part we focus on first order methods. Second order methods will be discussed further.

1. Gradient Method

A main example of first order methods is the Gradient Method also known as the Steeped Descent Method. Many practical problems have constraints but let's consider for the moment that we have none.

Problem: $\min_{x \in \mathbb{R}^n} f(x)$

Algorithm B.1: Gradient Algorithm

```
Given x_0, k = 0
Repeat
```
$x_{k+1} = x_k - \underbrace{h_k}_{\in \mathbb{R}} \underbrace{\nabla f(x_k)}_{\in \mathbb{R}^n}$

$k \leftarrow k + 1$

h_k is called the step length and $-\nabla f(x_k)$ is called the direction.

2. Step length selection

a. h_k that minimize $f(x_k - h_k \nabla f(x_k))$

We have to solve for h_k at each step. Even if we only minimize one variable, it's still an iterative method and does not give directly h_k.

b. $h_k = \alpha$

Example B.1. *Let's consider x^2 and see what happens when we change h_k*

$\boxed{h_k = 2}$ $\forall k$
Set x_0
$x_1 = x_0 - 2(2x_0) = -3x_0$
$x_2 = -3x_0 - 2(-6x_0) = 9x_0$
Instead of going to zero we go to $-3x_0$ or $9x_0$. Our step is clearly too large. This is called DIVERGING.

$\boxed{h_k = 1}$ $\forall k$
$x_1 = x_0 - 2x_0 = -x_0$
$x_2 == -x_0 - (-2x_0) = x_0$
This time, we are still too large and this is called CYCLING.

$\boxed{h_k = \frac{1}{2}}$ $\forall k$
$x_1 = x_0 - \frac{1}{2}(2x_0) = 0$
We have a FINITE CONVERGENCE.

$\boxed{h_k = \frac{1}{3}}$ $\forall k$
$x_1 = x_0 - \frac{1}{3}(2x_0) = \frac{1}{3}x_0$
$x_2 = \frac{1}{3}x_0 - \frac{1}{3}(2\frac{1}{3}x_0) = \frac{1}{9}x_0$
$\vdots$
$x_k = \frac{1}{3^k}x_0$

We get closer to x_0.

The simple gradient method is subjected to poor step length selection. However it still good to use a constant step length. We'll explicitly and numerically compute some value of α which guarantee a good behavior.

c. h_k satisfies some "dynamic" conditions (e.g. Wolfe condition)

In order to know what is the best step length, we need to know the function. We focus only on functions that are $C_L^{1,1}$, which means that $f \in C^1$ and ∇f is Lipschitz with constant L. The Lipschitz condition is:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$$

In order words, if we take 2 points, their gradient can't be too different.

Mathematically, we can define:
Definition B.1. *Given $L > 0$, we say $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ has L-*Lipschitz gradient *if and only if $f \in C^1(D)$ and*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\,\|x - y\|$$

for all $x, y \in D$. We denote $C_L^{1,1}(D)$ the set of such functions. We also define

$$F_L^{1,1}(D) = \{f \in C_L^{1,1}(D) \mid f \text{ is convex}\}.$$

Given $f \in C^1$, we also denote $T_y^1(x) = f(y) + \nabla f(y)^T(x-y)$ the first-order Taylor expansion of f around y evaluated at point x.

Example B.2. *Function that are Lipschitz with L=0? Linear function are* $C_0^{1,1}$

Example B.3. *Standard quadratic function* $x \to x^T Qx$ *with no convexity assumption for now. The Lipschitz condition can be expressed as:*

$$\|2Qx - 2Qy\| \leq L\|x - y\| \quad \forall x, y$$

$$2\|Q(x - y)\| \leq L\|x - y\| \quad \forall x, y$$

If $Q = Id$*, it's clear that* $L = 2$*. When it comes to matrix like* Q*, we will use the spectral norm. From the matrix theory, we know that:*

$$\|Qv\|_2 \leq \underbrace{\|Q\|_2}_{max|\lambda_i(Q)|} \|v\|_2$$

One can hence choose

$$L = 2\|Q\|_2$$

For more complicated function, we can try to bound the value of L *and use its estimate.*

As we can see, it's very easy to compute the constant for a quadratic function. But what can we do in a more general case?

Property B.1. *When* $f \in C^2$ *the Lipschitz constant is given by* $L = \max_x \|\nabla^2 f(x)\|$.

This result is very useful for one variable functions but not for multi-variables functions. The computation can be hard in that case.

Example B.4. *Let's apply proposition B.1 with* $f : x \to \sqrt{1 + x^2}$

$$f'(x) = \frac{x}{\sqrt{1+x^2}}$$

$$f''(x) = \frac{\sqrt{1+x^2} - x\frac{x}{\sqrt{1+x^2}}}{1+x^2} = \frac{1}{(1+x^2)(\sqrt{1+x^2})}$$

We easily see that:

$$0 \leq \frac{1}{(1 + x^2)(\sqrt{1 + x^2})} \leq 1$$

The upper bound is clear and we also put the lower bound because we want the absolute value of $f''(x)$ *to be bounded. We will take* $L = 1$*.*

What can we say about the minimum of a function based only on its first derivative? We are tempted to use the Taylor series which approximate the function by its tangent. Unfortunately we can't say anything from this about the minima. One better idea is to construct an upper bound approximation of our function using the Lipschitz constant then try to minimize this bound. We are thus sure that our function will stay below this bound, as shown in Figure B.1.

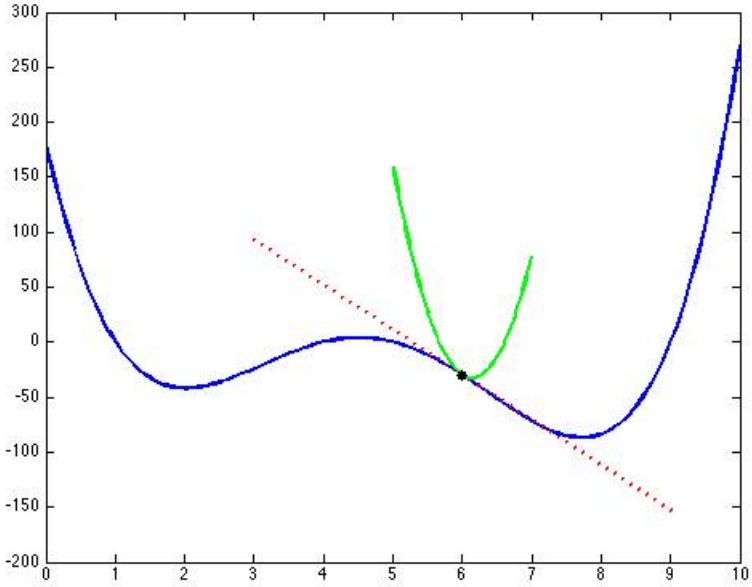


Figure B.1: In blue: f of $C_L^{1,1}$, in red: tangent to the function, in green: the QUB

Lemma B.1. ***Quadratic upper bound (QUB) of $C_L^{1,1}$ function***
Let's $f \in C_L^{1,1}$. Then for any x we have that

$$\hat{f} : y \mapsto f(x) + \nabla f^T(x)(x-y) + \frac{L}{2}\|x-y\|$$

is an upper bound of $f(y)$.

Notice that $\hat{f}$ is a quadratic function of y. What is the minimum of the quadratic upper bound? Our function is now convex and easy to minimize over y.

$$\nabla \hat{f}(y) = 0 \Leftrightarrow \nabla f(x) + \frac{L}{2}2(x-y)(-1) = 0$$

$$\Rightarrow y^* = x - \frac{1}{L}\nabla f(x)$$

We can notice that the last equation is very similar to the gradient method. It is sensible to use $h_k = \frac{1}{L}$ in our method.

d. Analysis of the $h_k = \frac{1}{L}$ gradient method

After one step, taking $y = x - \frac{1}{L}\nabla f(x)$, the upper bound gives:

$$f(x) + \nabla f(x)^T[\frac{-1}{L}\nabla f(x)] + \frac{L}{2}\|\frac{-1}{L}\nabla f(x)\|^2 = f(x) - \frac{\|\nabla f(x)\|^2}{L} + \frac{1}{2L}\|\nabla f(x)\|^2$$

$$= f(x) - \underbrace{\frac{1}{2L}\|\nabla f(x)\|^2}_{\leq 0}$$

Looking at all iterations now:

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

(because the upper bound displays this decrease What does it mean?)

$$f(1) - f(0) \leq -\frac{\|\nabla f(x_0)\|^2}{L}$$
$$f(2) - f(1) \leq -\frac{\|\nabla f(x_1)\|^2}{L}$$
$$\vdots$$
$$f(N+1) - f(N) \leq -\frac{\|\nabla f(x_N)\|^2}{L}$$

Summing the equations above, we get
$f(N+1) - f(0) \leq -\frac{1}{L}\sum_{i=1}^{n} \|\nabla f(x_i)\|^2$
so we deduce
$L(f(0) - f(N+1)) \geq \sum_{i=1}^{n} \|\nabla f(x_i)\|^2 \geq (N+1)\min \|\nabla f(x_i)\|^2$

Hence

$$\min_{i=0,\dots,N} \|\nabla f(x_i)\| \leq \sqrt{\frac{L(f(0) - f(N+1))}{N+1}}$$
$$\leq \sqrt{\frac{L(f(0) - f(x^*))}{N+1}}$$

with x^* minimum point of f.

Lemma B.2. *Assume in addition that f is convex (meaning $f \in F_L^{1,1}$)[1], then we have:*

$$f(N) - f(x^*) \leq \frac{L\|x_0 - x^*\|^2}{2N}$$

(assuming x^* is one of the minimum points).

Example B.5. *Behaviour of the gradient method*

$$2x^2 + y^4 - 2y^2$$

we start from (1,0)

$$\nabla = \begin{pmatrix} 4x \\ 4y^3 - 4y \end{pmatrix}$$

[1] f is convex and $f \in C_L^{1,1}$

$$\nabla^2 = \begin{pmatrix} 4 & 0 \\ 0 & 12y^2 - 4 \end{pmatrix}$$

We have three stationary points $(0,0)$,$(0,1)$,$(0,-1)$.

We observe that

$$\nabla(1,0) = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$$

If we start from $(1,0)$*, we can only converge to* $(0,0)$ *and we couldn't find all the points.*

3. Gradient method for unconstrained problems

a. Gradient method for functions of $C_L^{1,1}$

In the previous section, we studied the gradient method for the general problem.

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

where $f \in C_L^{1,1}(\mathbb{R}^n)$. We proved that $\frac{1}{L}$ was actually the best step length choice and that it guaranteed

$$\min_{0 \le i \le N} \|\nabla f(x_i)\| \le \sqrt{\frac{2L(f(x_0) - f(x^*))}{N+1}}. \tag{B.1}$$

Observe that this inequality

- is scaling independent,
- doesn't say anything about the values of f.

One can show that this inequality is not improvable.

Let us recall the way we obtained the above inequality.

Lemma B.3 (Quadratic bounds in $C_L^{1,1}$)**.** *The following conditions are equivalent :*

1. $f \in C_L^{1,1}(D)$,
2. $f \in C^1(D)$ *and* $|f(y) - T_x^1(y)| \le \frac{L}{2} \|x - y\|^2$ *for all* $x, y \in D$.

Proof See the fourth exercises session.

QED

From this lemma, we concluded that $\frac{1}{L}$ is the optimal step length.

Theorem B.1 (Decrease guarantee)**.** *Let* $f \in C_L^{1,1}$*. Denote* $x^+ = x - \frac{1}{L}\nabla f(x)$ *the next iterate. Then*

$$f(x) - f(x^+) \ge \frac{\|\nabla f(x)\|^2}{2L}.$$

Proof Use the upper bound of lemma B.3 with $y = x^+$.

QED

Actually there exist a family of functions which can be as closed of this bound as you want.

Finally theorem B.1 leads to the inequality (B.1).

b. Gradient method for functions of $F_L^{1,1}$

Let us now consider the same problem with the additional assumption $f \in F_L^{1,1}(\mathbb{R}^n)$. Lemma B.3 can be improved as follows.

Lemma B.4 (Quadratic bounds in $F_L^{1,1}$)**.** *The following conditions are equivalent :*

1. $f \in F_L^{1,1}(D)$,
2. $f \in C^1(D)$ *and* $T_y^1(x) \leq f(x) \leq T_y^1(x) + \frac{L}{2}\|x-y\|^2$ *for all* $x, y \in D$.

Proof See the fourth exercises session.

QED

Lemma B.5. *Let* $f \in C_L^{1,1}$. *For any optimal solution* x^* *and any* x,

$$\frac{\|\nabla f(x)\|^2}{2L} \leq f(x) - f(x^*) \leq \frac{L}{2}\|x - x^*\|^2.$$

Proof The first inequality follows from theorem B.1 since $f(x^+) \geq f(x^*)$. The second inequality follows from lemma B.3 applied with $x = x^*$. Indeed since $f \in C^1$ and x^* is a local extremum, we have $\nabla f(x^*) = 0$.

QED

Theorem B.2 (Convergence of $\frac{1}{L}$-gradient method for $F_L^{1,1}$)**.** *Let* $f \in F_L^{1,1}$. *For any iterate* x_N *and any* x,

$$f(x_N) - f(x^*) \leq \frac{L}{2N}\|x_0 - x^*\|^2.$$

Proof We start from theorem B.1:

$$f(x^+) \leq f(x) - \frac{\|\nabla f(x)\|^2}{2L}.$$

Since $f \in C^1$ and f is convex, we have

$$f(x^*) \geq f(x) + \nabla f(x)^T(x^* - x),$$

right-hand side being tangent equation around x. Combining these two inequalities yields

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

Now, observe that[2]

$$\nabla f(x)^T(x - x^*) - \frac{1}{2L}\|\nabla f(x)\|^2 = \frac{L}{2}\left(\|x - x^*\|^2 - \left\|x - x^* - \frac{1}{L}\nabla f(x)\right\|^2\right).$$

Noting that $x - \frac{1}{L}\nabla f(x) = x^+$, we obtain

$$f(x^+) - f(x^*) \leq \frac{L}{2}\left(\|x - x^*\|^2 - \left\|x^+ - x^*\right\|^2\right).$$

So given $N \in \mathbb{N}$, we have for all $i \in \{0, ..., N-1\}$

$$f(x_{i+1}) - f(x^*) \leq \frac{L}{2}\left(\|x_i - x^*\|^2 - \|x_{i+1} - x^*\|^2\right).$$

Summing those N inequalities yields

$$\begin{aligned}\sum_{i=0}^{N-1} f(x_{i+1}) - Nf(x^*) &\leq \frac{L}{2}\sum_{i=0}^{N-1}\left(\|x_i - x^*\|^2 - \|x_{i+1} - x^*\|^2\right)\\ &= \frac{L}{2}\left(\|x_0 - x^*\|^2 - \|x_N - x^*\|^2\right)\\ &\leq \frac{L}{2}\|x_0 - x^*\|^2.\end{aligned}$$

Notice now that $f(x_N) \leq f(x_i)$ for all $i \in \{0, ..., N-1\}$ so that

$$Nf(x_N) \leq \sum_{i=1}^{N} f(x_i).$$

Using this in the last inequality, we finally get

$$f(x_N) - f(x^*) \leq \frac{L}{2N}\|x_0 - x^*\|^2.$$

QED

Among all methods with

$$x_k \in \text{span}\{x_0, \nabla f(x_0), ..., \nabla f(x_{N-1})\},$$

none of them can guarantee better than

$$f(x_N) - f(x^*) \leq \frac{3}{32}L\frac{\|x_0 - x^*\|^2}{(N+1)^2}$$

for dimension greater or equal to $2N + 1$.

[2] Apply $\|a - b\|^2 = \|a\|^2 - 2a^Tb + \|b\|^2$ to $a = x - x^*$ and $b = \frac{1}{L}\nabla f(x)$. Ok, it's a trick.

4. Gradient method for constrained problems

We now add constraints. We consider problems of the following form:

$$\min_{x \in C} f(x) \qquad \text{with} \quad f \in C_L^{1,1}(C)$$

and

$$\min_{x \in C} f(x) \qquad \text{with} \quad f \in F_L^{1,1}(C).$$

We assume $C \subseteq \mathbb{R}^n$ is a convex and closed set. This implies that the orthogonal projection on C

$$P_C : \mathbb{R}^n \to C : x \mapsto P_C(x)$$

is well defined and unique.

Definition B.2. *Let $\mathcal{C}$ be a closed convex set and f a differentiable function ($f \in C_L^{1,1}(C)$), we say that x^* is a* ***stationary point*** *of the problem $\min_{x \in C} f(x)$ if and only if*

$$\langle \nabla f(x^*), x - x^* \rangle \;\geq\; 0 \qquad \forall x \in \mathcal{C}$$

for all $x \in C$.

This definition can be intuitively interpreted as follows: adding $f(x^*)$ on both sides brings up the first-order Taylor expansion of f around x^*, which is closed to $f(x)$. So this definition essentially means $f(x) \geq f(x^*)$.
In another interpretation, that means that all possible errors $x - x^*$ are in opposite direction with $-\nabla f$. A schema of the situation is shown on figure B.2.

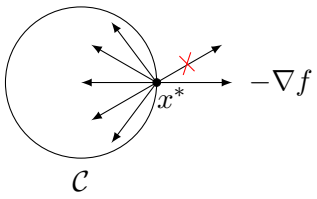


Figure B.2: Example of a stationnary point x^*.

Example B.6. $\mathcal{C} = \{x \in \mathbb{R}^n : x_i \geq 0\}$ *(nonnegative orthant)*

$$x^* \text{ stationnary iff } \sum_i [\nabla f(x^*)]_i \, [x_i - x_i^*] \geq 0 \quad \forall x \geq 0$$

$$x^* \text{ stationnary iff either } [\nabla f(x^*)]_i = 0$$
$$\text{or } [\nabla f(x^*)]_i > 0 \text{ and } x_i^* = 0 \;\forall\, i$$

Indeed, as $\sum_i [x_i - x_i^*] \geq 0$ for all $x \geq 0$, $[\nabla f(x^*)]_i$ must be ≥ 0. If not, we could choose a very large x_i for this component and have a negative sum.

Example B.7. *It also works easily for $\mathcal{C} = \{x | \sum_i x_i = 1\}$ which is a kind of budget constraint. In that case, it is possible to show that $[\nabla f(x^*)]_i = \lambda \;\forall i$. That means that all the gradient components are equal to each other, or economically speaking that the marginal costs are equal to each other. At the optimum, the marginal cost is equal for each component, it does not matter which one you lower.*

Example B.8. *(not treated) The Euclidian bowl.*

Note that if $x^* \in \operatorname{int} C$, then necessarily $\nabla f(x^*) = 0$. Indeed, if $\nabla f(x^*) \neq 0$ and $x^* \in \operatorname{int} C$, we can choose x such that $\nabla f(x^*)$ and $x - x^*$ are of opposite directions. Consequently, their scalar product is negative which contradicts the definition of x^*. This implies that $\nabla f(x^*) = 0$.

Theorem B.3. *Under the above assumptions, if x^* is a local minimum, then x^* is stationary.*

Theorem B.4. *When f is convex, stationary implies optimality.*

a. Projected gradient method

Let us now present the gradient method for constrained problems. The principle is the following:

1. at each step, minimize the quadratic upper bound on set C,
2. which is equivalent to projecting the true minimum of the quadratic upper bound on set C.

Let us show this equivalence. Statements mean

1. choose x^+ minimizing $f(x) + \nabla f(x)^T(x^+ - x) + \frac{L}{2}\|x - x^+\|^2$ over C, where we can ignore the constant term $f(x)$ in the minimization problem,
2. choose x^+ minimizing $\left\|x^+ - (x - \frac{1}{L}\nabla f(x))\right\|^2$ over C. Notice that we can develop

$$\begin{aligned}\left\|x^+ - (x - \tfrac{1}{L}\nabla f(x))\right\|^2 &= \left\|(x^+ - x) + \tfrac{1}{L}\nabla f(x))\right\|^2 \\ &= \|x^+ - x\|^2 + \frac{2}{L}(x^+ - x)^T\nabla f(x) + \frac{1}{L^2}\|\nabla f(x)\|^2\end{aligned}$$

   which is equivalent to 1 since we can ignore the constant term $\|\nabla f(x)\|^2 / L^2$ and multiply by $L/2$ without changing the minimization problem.

This results in the following algorithm.

Algorithm B.2: Projected gradient method

```
Given x_0, L, k = 0
Repeat
    x_{k+1} = P_C(x_k - (1/L) ∇f(x_k))
    k ← k + 1
```

b. Gradient mapping for constrained problems

Definition B.3. *For some $M > 0$, the* ***gradient mapping*** $G_M^C f(x)$ *is the unique vector satisfying*

$$x - \frac{1}{M} G_M^C f(x) = P_C \left[x - \frac{1}{M} \nabla f \right]$$

The role of the gradient mapping is similar to the one of the gradient in the non-constraint case. Notice that if we take $\mathcal{C} = \mathbb{R}^n$, then $G_M^{\mathbb{R}^n} f(x) = \nabla f(x)$. An illustration of the gradient mapping is given in Figure B.3.

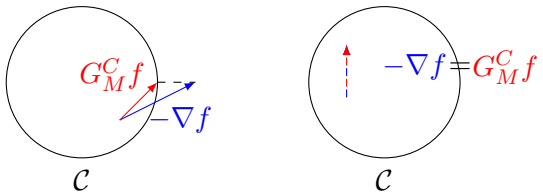


Figure B.3: Illustration of gradient mapping.

The gradient method becomes, in the case of constraint problems :

Algorithm B.3: Gradient Method - Constrained Problem

```
Given x_0, L, k = 0
Repeat
    x_{k+1} = P_C[x_k - (1/L) G_M^C f(x_k)]
    k ← k + 1
```

Property B.2. *For any $M > 0$, we have x^* stationary iff $G_M^C f(x^*) = 0$*

Property B.3. *Given $f \in C_L^{1,1}$, and letting $x^+ = x - \frac{1}{L} G_L^C f(x)$ be the next step, we have*

$$f(x) - f(x^+) \geq \frac{\left\| G_L^C f(x) \right\|^2}{2L}$$

Theorem B.5. *Using those properties, we obtain that for $f \in C_L^{1,1}$, the projected gradient method gives*

$$\min_{0 \leq i \leq N} \left\| G_L^C f(x_i) \right\| \leq \sqrt{\frac{2(f(x_0) - f(x^*))}{L(N+1)}}$$

We have a stronger result in the case of a convex f, as stated by the following theorem.

Theorem B.6. *Let $f \in F_L^{1,1}$. For any iterate x_N and any x,*

$$f(x_N) - f(x^*) \leq \frac{M \left\| x_0 - x^* \right\|^2}{2N}$$

The projected gradient method is quite slow because it is in $\mathcal{O}(\frac{1}{N})$ but also because projection can be complicated. Indeed, projection can be hard to compute if $\mathcal{C}$ is too complex.

Example B.9. *If $\mathcal{C} = \{x \geq 0\}$ this is easy because* $[P_C(x)]_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ 0 & \text{if } x_i < 0 \end{cases}$

Example B.10. *If $\mathcal{C} = \{x | Ax = b\}$ is a subspace, this is expensieve. We have indeed to solve a linear system, which is $\mathcal{O}(n^3)$.*

Example B.11. *If $\mathcal{C} = \{x | Ax \leq b\}$ is a polyhedron, this is even more expensive : the projection problem has to be written as a minimization of the distance (quadratic programming). We could also use a theorem that separates the problem on each facet of the polytope.*

c. Acceleration gradient [Nesterov 1983]

Algorithm B.4: Acceleration gradient

```
Given f ∈ C_L^{1,1}, x_0, L, k = 0 and x_{-1} = x_0
Repeat
    y_k = x_k + β_k(x_k − x_{k−1})
    x_{k+1} = P_C[y_k − (1/L)∇f(y_k)]
    k ← k + 1
```

The first step is called an extrapolation step whereas the second is called a gradient step. Note that y_k is not always in $\mathcal{C}$. The idea is to recycle the previous work because $(x_k - x_{k-1})$ will be close to the gradient.

Theorem B.7. *For $\beta_k = \frac{k-1}{k+2}$, we have*

$$f(x_N) - f(x^*) \leq \frac{2L \|x_0 - x^*\|^2}{(N+1)^2}$$

It can be proved that this rate of $\frac{1}{N^2}$ is not improvable. Accelerated gradient is thus a sublinear method (slower than linear). While the gradient method is really robust, the accelerated method is extremely sensitive. Note that β_k goes to 1 as k goes to infinity. An example is the Huber function : on the linear part, the steps increase in a quadratic way.

Linear convergence (to zero) : $1, \rho, \rho^2, \rho^3, \rho^4, \rho^5, ...$ with $\rho < 1$. If we restrict the class of functions to **strongly convex**, we can get linear convergence.

Definition B.4. *Given $\mu > 0$, f is μ-**strongly convex** iff*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) - \frac{\mu}{2}\lambda(1-\lambda) \|x - y\|^2$$

A strongly-convex function can be viewed as a function that has no flat part.

Proposition B.1. *If $f \in C^2$ then f is μ-strongly convex iff for all x*

$$\lambda_{min}(\nabla^2 f(x)) \geq \mu$$

Lemma B.6. *For any strongly convex function, we have*

$$\|x - x^*\|^2 \leq \frac{2}{\mu}[f(x) - f(x^*)]$$

That means that if I decrease the error on the function value, automatically I decrease the distance to the solution. Using previous results, we obtain :

$$\|x_N - x^*\|^2 \leq \frac{2}{\mu}\frac{L\,\|x_0 - x^*\|^2}{2N} = \frac{L}{\mu}\frac{\|x_0 - x^*\|^2}{N}$$

$\frac{L}{\mu}$ is called the condition number.

Remark: The accelerated gradient method gives better results if we restart the method after a certain number of iteration. For example, if the squared norm of the error is divided by 5 after $N = 10$ iterations, it will be divided by 10 after $N = 20$ iterations. But if we restart the method after 10 iterations and make 10 new iterations, the square norm of the error will be divided by 25 although we made a total of 20 iterations in both cases. A result that is not proven here is that the optimal number of iterations after which the method should be restarted is $\frac{L}{\mu}\mathrm{e}$. In this case, the method is linearly convergent and we have :

$$\|x_N - x^*\|^2 \leq \left(1 - \frac{1}{\mathrm{e}\frac{L}{\mu}}\right)\|x_0 - x^*\|^2$$

Chapter C

Conic modeling and duality

1. Linear and sublinear convergence

For methods with sublinear convergence, the value of the error firstly downs very quickly but after some value, the decreases becomes really slow. For methods with linear convergence, the decrease of the error is, at the beginning, slowler then sublinear methods. However, after the value where the decrease for sublinear methods becomes slow, the decrease for linear method remains constant on a logarithmic scale.

Thus, the choice of the method depends on the application. If one needs a lot of accuracy, it is preferable to use linear methods but if high accuracy is not that much important (i.e. we only look for one digit), one then should use sublinear methods. Some graphs to illustrated these results are available on the *Linear convergence is faster than sublinear convergence (eventually) [missing figure from week 7]* pdf file.

2. Conic optimization

In order to introduce the conic optimization concept, let's begin with an linear example.

Example C.1 (Linear optimisation problem)**.**

$$\begin{aligned} \max_{y_i} \quad & 2y_1 + 3y_2 + 2y_3 \\ \textit{subject to} \quad & y_1 + y_2 \leq 1 && \text{(C.1)} \\ & y_2 + y_3 \leq 2 && \text{(C.2)} \\ & y_3 \leq 3 && \text{(C.3)} \end{aligned}$$

A feasible point is given by $(y_1, y_2, y_3) = (1, 0, 2)$ which provides a cost function value 6.

How to know whether 6 *is strictly lower than or strictly higher than or equal to the optimal objective value ?*

Since a feasible point gives 6 as cost function value, the optimal objective value is not strictly lower than 6. The value 6 can be increased to 7 by taking the feasible point $(y_1, y_2, y_3) = (2, -1, 3)$. So the optimal objective value is not equal to 6 but higher than 6. Concerning the value 7, we could ask our question again, *how to prove that the optimal objective value is higher than or equal to* 7 *?*

We need an upper bound on the value for the objective function. This can be achieved by taking a linear combination of the constraints.

Indeed,

$$\begin{aligned} 2 \cdot \text{ constraint } C.1 + 1 \cdot \text{ constraint } C.2 + 1 \cdot \text{ constraint } C.3 &= 2y_1 + 3y_2 + 2y_3 \\ &\leq 7 \end{aligned}$$

7 is an upper bound for the objective value function. Since we have found a feasible point for which the objective value is equal to 7, we can claim that the optimal objective value is equal to 7.

It is possible to find a bound for the objective value for any linear problem. We want to extend the notion of linear problems while keeping their 'nice' properties of linear problems (i.e. duality and efficient algorithms). In order to extend this notion, we will generalize the inequalities $\leq$ and $\geq$ in the linear problems.

With $K \subseteq \mathbb{R}^n$, we define an generalized type of inequalities

$$a \succeq_K 0 \Leftrightarrow a \in K$$

On the basis of this definition, we also have the following properties

$$\begin{aligned} a \succeq_K b \Leftrightarrow a - b \succeq_K 0 \Leftrightarrow a - b \in K \\ a \preceq_K b \Leftrightarrow b \succeq_K a \Leftrightarrow b - a \succeq_K 0 \Leftrightarrow b - a \in K \end{aligned}$$

Let us also impose two sensible properties for an order
Property C.1. $a \succeq_K 0 \Rightarrow \lambda a \succeq_K 0$ *for any* $\lambda \geq 0$
Property C.2. $a \succeq_K 0$ *and* $b \succeq_K 0 \Rightarrow a + b \succeq_K 0$

The properties C.1 means that K is a *cone* and the properties C.2 means that K is *closed under addition.* The K must be convex, K is said to be a convex cone. In fact, for any cone K (closed under non-negative scalar multiplication), we have the equivalence

$$K \text{ is closed under addition } \Leftrightarrow K \text{ is convex}$$

a. Generalization

From the usual formulation of the problem, one can generalize:

$$\begin{gathered} \max b^T y \text{ such that } A^T y \leqslant \text{c} \\ \Downarrow \\ \max b^T y \text{ such that } A^T y \preceq_K \text{c} \end{gathered}$$

This problem is convex.
Considering the standard linear case, $K = \mathbb{R}^n_+$.

Example C.2. *Going back to example C.1, how would one define the set K?*
As:

$$A^T = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \qquad C = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Defining $K = \mathbb{R}^3_+$ implies:

$$C - A^T = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} y \subset K = \mathbb{R}^3_+$$

$$\begin{pmatrix} 1 - y_1 - y_2 \\ 2 - y_2 - y_3 \\ 3 - y_3 \end{pmatrix} \subset K = \mathbb{R}^3_+$$

b. Solving a problem on different cones

As one consider polyhedron cones, the boundary is linear. Also, applying a linear operation to a linear problem keeps the problem linear. As a result, applying a rotation or a scaling operation to a cone and solving the problem on the newly obtained cone is equivalent to solving it on the first cone.

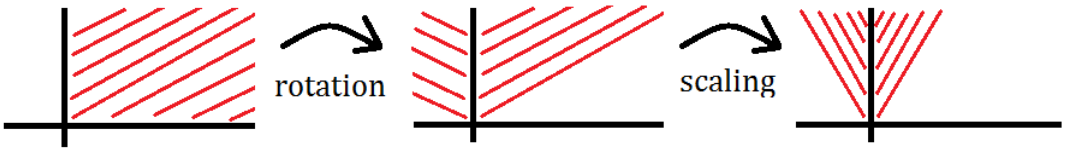


Figure C.1: Solving a problem on one cone is equivalent to solve it on a newly obtained cone via rotation and scaling.

The conic optimization is useful since it is more general than linear optimization. The sets considered are not polyhedrons anymore but rather curved sets. Thus, it can cover more problems.

c. Lorentz cone

A common example of a cone is the Lorentz cone or Ice cream cone. It is defined such as:

$$\mathbb{L}^n = \{(x_0, ...x_n) \in \mathbb{R}^{n+1} \mid \sqrt{x_1^2 + ... + x_n^2} \leq x_0$$

One can verify it is indeed a cone by checking that when multiplying a value in the cone by a scalar it stays in the cone; and that addition is preserved (intuition: rule of

parallelogram). Also, checking if the hull is a cone is the same as checking whether it is convex or not.

Example C.3. *Let the problem:*

$$\begin{gathered}\max(y_1 + y_2)\\ y_1^2 + y_2^2 \leq 7\end{gathered}$$

One notes that the constraint is quadratic and semi-definite positive. One would like to formulate the problem as linear, except for the cone. All that is non-linear shall be put in the cone, hence:

$$\begin{gathered} y_1^2 + y_2^2 \leq 7 \Leftrightarrow \sqrt{y_1^2 + y_2^2} \leq \sqrt{7} \\ \Updownarrow \\ \begin{pmatrix} \sqrt{7} \\ y1 \\ y2 \end{pmatrix} \in \mathbb{L}^2 \;\Leftrightarrow\; \begin{pmatrix} \sqrt{7} \\ y1 \\ y2 \end{pmatrix} \succeq_{\mathbb{L}^2} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\ \Updownarrow \\ \begin{pmatrix} \sqrt{7} \\ 0 \\ 0 \end{pmatrix} \succeq_{\mathbb{L}^2} \begin{pmatrix} 0 \\ -y_1 \\ -y_2 \end{pmatrix} \end{gathered}$$

Finally:

$$C = \sqrt{7}, \qquad A = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

And the problem can be written as:

$$\begin{gathered}\max(y_1 + y_2)\\ -y_1 - y_2 \preceq_{\mathbb{L}^2} \sqrt{7}\end{gathered}$$

d. Requirements for K

1. $x \succeq 0$ and $x \preceq 0 \Rightarrow x = 0$

   which means $K \cap (-K) = \{0\}$ (the cone is *pointed*)

2. We define the strict inequality by $a \succ 0 \Leftrightarrow a \in int\ K$ (and $a \succ b$ iff $a - b \in int\ K$)

   Hence we require $int\ K \neq \emptyset$ (the cone is *solid*)

3. Finally, we would like to be able to take limits:

   $$\text{If } \{x_i\}_{i\to\infty} \text{ with } x_i \succeq_K 0\ \forall i, \text{ then } \lim_{i\to\infty} x_i = \bar{x} \Rightarrow \bar{x} \succeq_K 0$$

   which is equivalent to saying that K is *closed*.

A convex cone K that is solid, pointed and closed will be called a *proper cone.*

As seen in section b., in the conic optimization theory, all the cones can always be merged as an only one according to the following definition.

Definition C.1. *Considering several conic constraints*

$$A_1^T y \preceq_{K_1} c_1 \text{ and } A_2^T y \preceq_{K_2} c_2$$

which are equivalent to

$$c_1 - A_1^T y \in K_1 \text{ and } c_2 - A_2^T y \in K_2$$

one introduces the product cone $K = K_1 \times K_2$ to write

$$(c_1 - A_1^T y, c_2 - A_2^T y) \in K_1 \times K_2$$

$$\Leftrightarrow \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} - \begin{pmatrix} A_1^T \\ A_2^T \end{pmatrix} \in K_1 \times K_2 \Leftrightarrow \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} - \begin{pmatrix} A_1^T \\ A_2^T \end{pmatrix} \succeq_{K_1 \times K_2} 0$$

If K_1 and K_2 are proper, $K_1 \times K_2$ is also proper.

However, in this course, and for the exercises, we will not merge the cones.

e. Equivalence with convex optimization

Since a cone is convex, conic optimization can be related to convex optimization: conic optimization is clearly a special case of convex optimization. What about the reverse statement ?

$$\min_{x \in \mathbb{R}^n} f(x) \text{ such that } x \in X \subseteq \mathbb{R}^n$$

- The objective of a convex problem can be assumed *w.l.o.g.*[1] to be *linear*: $f(x) = c^T x$
- The feasible region of a convex problem can be assumed *w.l.o.g.* to be in the *conic* standard format:
$$X = \{x \in K \text{ and } Ax = b\}$$

$\Rightarrow$ conic optimization *equivalent* to convex optimization.

One can prove that everything in the use of conic optimization, must be convex. Considering the conic constraint:

$$c - A^T y \in K$$

choosing K to be convex is a good way to ensure the constraint to be convex too since linear transformations preserve convexity.

[1] without loss of generality

3. Duality

What about the conic dual problems ? Since we generalized

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

to

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

it could be tempting to generalize

$$\min c^T x \text{ s.t. } Ax = b \text{ and } x \geq 0$$

to

$$\min c^T x \text{ s.t. } Ax = b \text{ and } x \succeq_K 0.$$

However, this is not the right primal-dual pair ! Actually, we must also dualize the cone; and the corresponding correct primal-dual pair is given by

$$\max b^T y \text{ such that } A^T y \preceq_K c$$

$$\min c^T x \text{ such that } Ax = b \text{ and } x \succeq_{K^*} 0$$

where we introduce the dual cone K^* defined as

$$K^* = \{z \in \mathbb{R}^n \text{ such that } x^T z \geq 0 \ \forall x \in K\}$$

- K^* is a convex cone, called the *dual* cone of K;
- K^* is always *closed*, and if K is closed, $(K^*)^* = K$;
- K is *pointed* (resp. solid) $\Rightarrow K^*$ is *solid* (resp. pointed)
- Cartesian products: $(K_1 \times K_2)^* = K_1^* \times K_2^*$;
- $(\mathbb{R}^n+)^* = \mathbb{R}^n+, (\mathbb{L}^n)^* = \mathbb{L}^n, (\mathbb{S}^n_+)^* = \mathbb{S}^n_+$:
  these cones are self-dual, but there are (many) cones which are not.

a. Exercise sessions on conic optimization

In chapter A, we saw how to model convex problems. In chapter B, we studied methods that deal with that type of problems. The disadvantage of those methods is that they can be quite slow, especially when compared with linear solvers (for linear systems). We now look at a certain class of convex problems, conic problems, that conserve some of the properties that make linear systems so effective. In particular, we want our problems to have duality, because a dual problem will be a much better problem to certify optimality. We start by looking at quadratic functions $x^T Q x$

$$\begin{array}{ll} x^T x & \\ \downarrow & Q \geq 0 \Longleftrightarrow Q = LL^T \\ x^T Q & \end{array}$$

$\Updownarrow$
$(L^t x)^T (L^T x)$

This last expression $(L^t x)^T (L^T x)$ is a linear transformation of $x^T x$ so it preserves convexity.

In this section, we will do a quick reminder on some exercises of Exercise session 5 on conic modelling: second order cone and duality.

i. Exercise 2

In this exercise, we have a rotated second order cone $\mathbb{L}_R^n$

$$\mathbb{L}_R^n = \{(x, y, z) \in \mathbb{R}^n \times \mathbb{R}_+ \times \mathbb{R}_+ \mid x_1^2 + x_2^2 + ... + x_n^2 \leq yz\}$$

We can prove that this set is a cone by replacing it by a second order cone. Why do we then use the rotated second order cone? The rotated second order cone is usefull because it makes some models easier (for example, the models in exercise 3).

ii. Exercise 4

The question here is "how do we deal with conic modelling of **functions**"? Let's take example (d).
Function $(x, z) \to \dfrac{\| x \|^2}{z}$ (on the domain $z > 0$). We have

$$\min_{\substack{x \in \mathbb{R}^n \\ z \in \mathbb{R}_{>0}}} \frac{\| x \|^2}{z}$$

This is not a conic problem because there is no linear objective. Our goal is to transform it in order to have a linear objective:

$$\min_{\substack{x \in \mathbb{R}^n \\ z \in \mathbb{R}_{>0}}} \frac{\| x \|^2}{z}$$

$$\Updownarrow$$

$$\min_{\substack{t \in \mathbb{R} \\ x \in \mathbb{R}^n \\ z \in \mathbb{R}_{>0}}} t \qquad s.t.$$

$$\| x \|^2 \leq tz$$

Now, we have a linear objective, and constraints of the rotated second order cone type.

iii. Exercise 6 : Conic duality

In this exercise, we need to find the dual for some optimization problems, using an equivalent conic formulation. To compute a dual problem, we need to reformulate it to a standard conic formulation.
Let's consider the dual and the primal problem:

$$(D) \qquad \begin{array}{c} \max\ b^T y \\ A^T y \preceq_K c \end{array} \quad \longleftrightarrow \quad \begin{array}{c} \min\ c^T x \\ Ax = b \\ x \succeq_{K^*} 0 \end{array} \qquad (P)$$

where (D) stands for the dual problem and (P) the primal problem. Note that for the positive real axis, second order cones and the semi definite cone: $K = K^*$.

b. Example of duality

i. Primal problem

Let's consider the following problem:

$$\begin{array}{ll} \min \sqrt{z^T Q z} & (Q \geq 0) \\ Cz \leq d & \end{array}$$

The function we minimize is quadratic, and Q is positive semi-definite. The constraint $Cz \leq d$ is a linear constraint. We need to transform this formulation into one of the two formulations above.
Let's say we first want to take care of the objective (since it is quadratic).

$$\begin{array}{ll} \min\ t & \\ \sqrt{z^T Q z} \leq t & \qquad Q = LL^T \\ \parallel L^T z \parallel \leq t & \end{array}$$

This is equivalent

$$\begin{pmatrix} t \\ L^T z \end{pmatrix} \in \mathbb{L}^n$$

Now, we want to reformulate the linear constraint:

$$Cz \leq d \Longleftrightarrow d - Cz \geq 0$$

This can be written as

$$d - Cz \in \mathbb{R}^m_+$$

Now, we are going to write everything using the cones:
Let $y = \begin{pmatrix} t \\ z \end{pmatrix} \in \mathbb{R}^{n+1}$ (t is composed of 1 element and z of n elements).

Objective $-\max b^T y$ with $b = \begin{pmatrix} -1 \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}$

Constraint: $c - A^T u \geq_K 0$ with $K = \mathbb{R}^m_+ \times \mathbb{L}^n$ (big vector with the top part being $\mathbb{R}^m_+$ and the bottom part $\mathbb{L}^n$)

$\begin{bmatrix} d - Cz \\ \hline t \\ L^T z \end{bmatrix} \in K$, where $d - Cz \in \mathbb{R}^m_+$ and $\begin{bmatrix} t \\ L^T z \end{bmatrix} \in \mathbb{L}^n$

Note that $d - Cz$ is composed of m elements, t of one element and $L^T z$ of n elements.

We are not completely done yet because we still need c and A^T:

$$c = \begin{pmatrix} d \\ \hline 0 \\ . \\ . \\ . \\ 0 \end{pmatrix} \qquad A^T = \begin{pmatrix} 0 & C \\ -1 & 0^{1\times n} \\ 0^{n\times 1} & -L^T \end{pmatrix}$$

The first column of A^T corresponds to the coefficients of t.

By telling our solver the different matrices and the cone, it will solve the problem. Why is this useful? Because now we can write the dual without a problem.

ii. Dual Problem

$$K^* = (\mathbb{R}^m_+ \times \mathbb{L}^n)^* = (\mathbb{R}^m_+)^* \times (\mathbb{L}^n)^* = \mathbb{R}^m_+ \times \mathbb{L}^n = K$$

Let's consider $x = \begin{pmatrix} u \\ v \end{pmatrix}$ with $u \in \mathbb{R}^m_+$ and $v \in \mathbb{L}^n$

The dual is

$$-\min \begin{bmatrix} d \\ 0 \end{bmatrix}^T \begin{pmatrix} u \\ v \end{pmatrix}$$

$$\begin{bmatrix} 0^{1\times n} & -1 & 0^{1\times m} \\ C^T & 0^{k\times 1} & -L \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -1 \\ 0^{k\times 1} \end{pmatrix} \tag{C.4}$$

Where n, m and k are the dimensions of C and L.

After simplifying, we have

$$-\min d^T u$$

$$u \geq 0$$

$$v_0 \geq \| (v_1 ... v_n) \|$$

$$C^T u - L \begin{pmatrix} v1 \\ . \\ . \\ . \\ v_n \end{pmatrix} = 0$$

$$-v_0 = -1$$

By putting $w = (v_1 ... v_n)$, we get the following simplified model

$$-\min d^T u$$

$$u \geq 0$$

$$\| w \| \leq 1$$

$$C^T u = Lw$$

This model cannot be simplified any further.

c. Theorems and properties of Dual

After this example of conic dual computation, we want to study the properties of the conic dual optimization and some of its theorems. One interesting design allows to prove some theorems. Let us first explain this design to eventually find the dual of the primal conic problem (P):

$$\min \mathbf{b}^T \mathbf{y}$$
$$\mathbf{A}^T \mathbf{y} \preceq_K \mathbf{c}$$

First, to build the standard dual problem, we write the constraints as a system of inequations where coefficients $\lambda_i \geq 0$ multiplies c_i (positivity of $\boldsymbol{\lambda}$ assures the preservation of inequalities):

$$\begin{array}{c} \begin{bmatrix} a_1^T \boldsymbol{y} \leq c_1 \\ a_2^T \boldsymbol{y} \leq c_2 \\ \vdots \\ a_n^T \boldsymbol{y} \leq c_n \end{bmatrix} \begin{array}{c} \times \lambda_1 \\ \times \lambda_2 \\ \vdots \\ \times \lambda_n \end{array} \\ \hline (\sum_{i=1}^n \lambda_i a_i)^T \leq \sum_{i=1}^n \lambda_i c_i \end{array}$$

We can write the new system system as:

$$(\boldsymbol{A\lambda})^T \boldsymbol{y} \leq \boldsymbol{c}^T \boldsymbol{\lambda}$$

The dual goal is to find the best upper bound for this system. This is, that best upper bound is found by minimizing the dual objective.

When $\boldsymbol{A\lambda} = \boldsymbol{b}$, we get the system $\boldsymbol{b}^T \boldsymbol{y} \leq \boldsymbol{c}^T \boldsymbol{\lambda}$. The upper bound depends on $\boldsymbol{\lambda}$ and the dual problem (D) becomes:

$$\begin{aligned} \min \boldsymbol{c}^T \boldsymbol{\lambda} \\ \text{such that } \boldsymbol{A}\boldsymbol{\lambda} = \boldsymbol{b} \\ \boldsymbol{\lambda} \geq \boldsymbol{0} \end{aligned}$$

In a conic problem, we have:

$$\begin{bmatrix} a_1^T \boldsymbol{y} \\ a_2^T \boldsymbol{y} \\ \vdots \\ a_n^T \boldsymbol{y} \end{bmatrix} \preceq_K \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

A simple example on the Lorentz cone show that the standard dual does not always work in conic optimization: let us have:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \preceq_{\mathbb{L}^2} \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

This is valid since on the Lorentz cone we must have in this case the condition $1 \geq ||(-1, 0)||$ satisfied, which is the case. However, if we now multiply $\boldsymbol{c}^T$ by a positive vector $\boldsymbol{\lambda} = (2, 10, 0)$, we get then:

$$\begin{array}{ccc} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \preceq_{\mathbb{L}^2} & \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} & \begin{matrix} \times 2 \\ \times 10 \\ \times 0 \end{matrix} \\ \hline & 0 \overset{?}{\leq} 2 - 10 + 0 & \end{array}$$

In this example, the positivity of $\boldsymbol{\lambda}$ does not preserve the inequalities as we need to in order to use the standard dual formulation.

i. Necessary and sufficient condition

The idea is that when starting with something positive and multiplying by $\boldsymbol{\lambda}$, we must stay positive. For instance a good multiplier on the Lorentz cone would be such that:

$$\begin{array}{ccc} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \preceq_{\mathbb{L}^2} & \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} & \begin{matrix} \times \lambda_1 \\ \times \lambda_2 \\ \vdots \\ \times \lambda_n \end{matrix} \\ \hline & 0 \leq \sum_{i=1}^n \lambda_i x_i & \end{array}$$

The condition to have a correct $\boldsymbol{\lambda}$ is the following: whatever the $\boldsymbol{x}$ we start with, $\boldsymbol{\lambda} \in K^*$ if $\boldsymbol{\lambda}^T \boldsymbol{x} \geq 0$:

$$\boldsymbol{\lambda}^T \boldsymbol{x} \geq 0 \ \forall \boldsymbol{x} \in K \Leftrightarrow \boldsymbol{\lambda} \in K^*$$

Where K^* is the dual cone.

Example C.4. ***Dual cone for*** $\mathbb{R}_+$ ***and*** $\mathbb{R}_+^n$

$\Rightarrow (\mathbb{R}_+)^* = ?$
$(\mathbb{R}_+)^* = \{\lambda | x\lambda \geq 0,\ \forall x \in \mathbb{R}_+\}$

- $\lambda \geq 0$ *: this works because* $\lambda x \geq 0$ *when* $x, \lambda \geq 0$
- $\lambda < 0$ *: one can pick* $x = 1$ *and observe that* $x\lambda < 0$*, that is* $\lambda \notin (\mathbb{R}_+)^*$

$\Rightarrow (\mathbb{R}_+^n)^* = ?$
$(\mathbb{R}_+^n)^* = \{\boldsymbol{\lambda} | \boldsymbol{x}^T \boldsymbol{\lambda} \geq 0,\ \forall \boldsymbol{x} \in \mathbb{R}_+^n\}$

- $\boldsymbol{\lambda} \geq 0$ *: this works because* $\boldsymbol{x}^T\boldsymbol{\lambda} \geq 0$ *when* $\boldsymbol{x}, \boldsymbol{\lambda} \geq 0$
- $\lambda_i < 0$ *: one can pick a* $\boldsymbol{x}$ *such that one of its elements is negative, but not negative everywhere:*

$$\boldsymbol{\lambda} = \begin{bmatrix} ? \\ \vdots \\ < 0 \\ \vdots \\ ? \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

This way, $\lambda_i^T x_i < 0$*, and that implies dual cone is* $(\mathbb{R}_+^n)$*.*

Example C.5. ***Dual cone for*** $\mathbb{L}^n$

$\Rightarrow (\mathbb{L}^n)^* = ?$
$(\mathbb{L}^n)^* = \{\lambda | x^T\lambda \geq 0,\ \forall x \in \mathbb{L}^n\}$

- $\lambda \in \mathbb{L}^n$ *:* $x^T\lambda = x_0\lambda_0 + [x_1...x_n][\lambda_1...\lambda_n]^T$*:*
  Because we know= $x_0 \geq ||x_1...x_n||$ *and* $\lambda_0 \geq ||\lambda_1...\lambda_n||$*, we can say:*

$$\begin{aligned} x^T\lambda &\leq x_0\lambda_0 + ||x_1...x_n||||x_1...x_n|| \\ &\geq x_0\lambda_0 - ||x_1...x_n||||x_1...x_n|| \geq 0 \end{aligned}$$

- $\lambda \notin \mathbb{L}^n$*:* $\lambda_0 < ||\lambda_1..\lambda_n||$
  One can pick x *the following way:* $x_0 = ||\lambda_1...\lambda_n||$*;* $[x_1..x_n] = -[\lambda_1...\lambda_n]$
  In that case,

$$\begin{aligned} x^T\lambda &= \lambda_0||\lambda_1..\lambda_n|| - ||\lambda_1..\lambda_n||^2 \\ &= (\lambda_0 - ||\lambda_1..\lambda_n||)||\lambda_1..\lambda_n|| < 0 \end{aligned}$$

This implies the second order cone is self-dual.

ii. Theorems

We can conclude what has been done with two theorems on conic duality:

Theorem C.1 (Weak duality)**.** *If x is feasible for the primal (P) and y is feasible for the dual (D), then*

$$c^T x \geq b^T y$$

Theorem C.2. *Under some assumptions, optimal value of (P) is equal to the optimal value of (D).*

Chapter D

Interior-point methods

Part II

Labs

1. Introduction: AMPL

AMPL (which stands for "A mathematical programming language") is an algebraic modelling language which enables the solving of high complexity problems for large (or small) scale models and will be used throughout this course. Knowing or identifying what type of problem we're dealing with is important and allows us to decide which will be the most adequate method to solve it. Certain methods work well with certain types of models, some are better than others, etc.

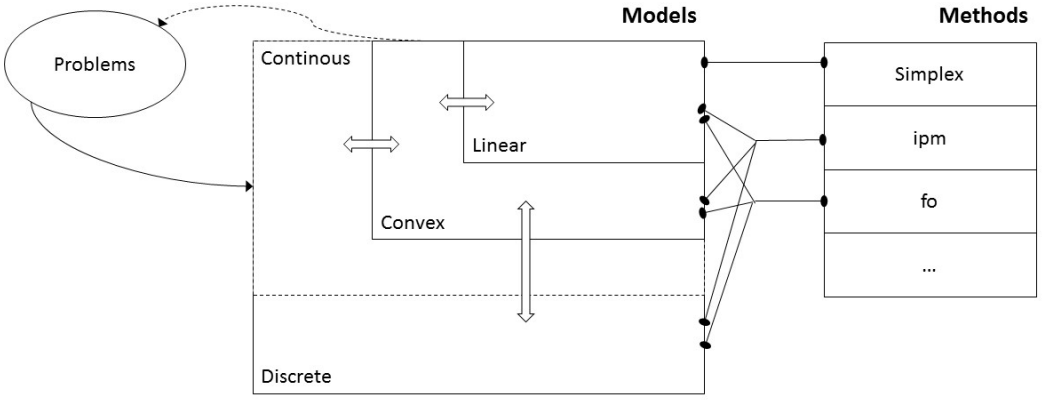


Figure D.1: Visualization of the relation between models and methods

AMPL works by first reading a text file which contains all the useful information of the model, this file usually ends in ".mod", it then parses it and tries to solve the problem. Parameters, variables, objective function and constraints are all defined in this file. The solving part is done by communicating with a solver, AMPL gives it all the information, the solver then sends back the solution. Consider the following optimization problem:

$$\begin{aligned} &\min c^T x \\ &Ay \leq b \\ &l \leq y \leq u \end{aligned}$$

For this problem AMPL will give the solver the usefull values (c,A,b,l,u,...) it neads to solve the problem. There exists quite a few of these solvers, some mode adapted to certain model types (linear, convex,...). We have for example:

- minos (basic solver for linear and nonlinear problems)

- cplex (can be used for linear, convex, mixted integer models)
- gurobi (very similar to cplex)
- knitro (good for nonlinear models)
- ...

This can be summerized by the next table:

Solvers	Problems	Integer variables
CPLEX, GUROBI	linear optimization and convex quadratic optimization	yes
KNITRO, SNOPT, MINOS	nonlinear optimization	yes for KNITRO but loss of efficiency
BARON	global optimization	no

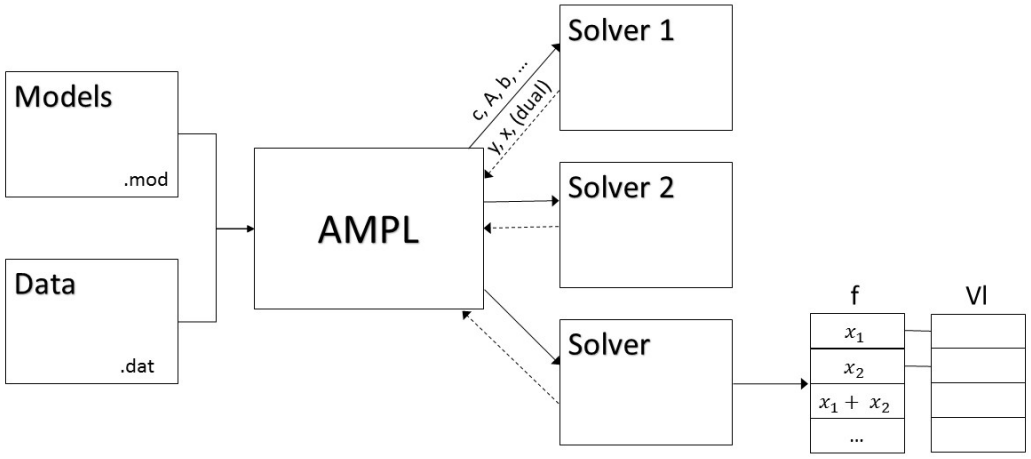


Figure D.2

AMPL can also work with an additionnal data file (".dat") which is used when parameters are left in the model file. This allows us to avoid changing the entire file when looking at different values of parameters, and only having to change them once in the data file.

A few examples and the basic syntax for AMPL can be found in the Tutorials Dropbox, given on Moodle.

Everything in AMPL has a name, whether it be variables, constants, or even constraints (which represent a dual variable) and each command and declaration ends with a semicolon (";"). Certain commands are worth being reminded here, for example:

- Changing solvers: **option solver ... ;**

- Displaying dual variable: **display Protein;**
- Displaying variable: **display Protein.body;**
- Reset the whole model: **reset;**
- Chosing a model file: **data data1.dat;**
- Chosing a data file: **model data1.mod;**
- Solving the chosen model: **solve;**

Part III

Exercices