Introduction to Unconstrained Optimization

Michaël Baudin

August 2010

Abstract

This document is a small introduction to unconstrained optimization optimization with Scilab. In the first section, we analyze optimization problems and define the associated vocabulary. We introduce level sets and separate local and global optima. We emphasize the use of contour plots in the context of unconstrained and constrained optimization. In the second section, we present the definition and properties of convex sets and convex functions. Convexity dominates the theory of optimization and a lot of theoretical and practical optimization results can be established for these mathematical objects. We show how to use Scilab for these purposes. We show how to define and validate an implementation of Rosenbrock’s function in Scilab. We present methods to compute first and second numerical derivatives with the derivative function. We show how to use the contour function in order to draw the level sets of a function. Exercises (and their answers) are provided.

Contents

1 Overview 4
  1.1 Classification of optimization problems 4
  1.2 What is an optimization problem ? 6
  1.3 Rosenbrock’s function 7
  1.4 Level sets 9
  1.5 What is an optimum ? 11
  1.6 An optimum in unconstrained optimization 13
  1.7 Big and little O notations 13
  1.8 Various Taylor expansions 14
  1.9 Notes and references 15
  1.10 Exercises 15

2 Convexity 15
  2.1 Convex sets 16
  2.2 Convex functions 18
  2.3 First order condition for convexity 19
  2.4 Second order condition for convexity 23
  2.5 Examples of convex functions 24
  2.6 Quadratic functions 27
1 Overview

In this section, we analyze optimization problems and define the associated vocabulary. We introduce level sets and separate local and global optimums. We emphasize the use of contour plots in the context of unconstrained and constrained optimization.

1.1 Classification of optimization problems

In this document, we consider optimization problems in which we try to minimize a cost function

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

(1)

with or without constraints. Several properties of the problem to solve may be taken into account by the numerical algorithms:

- The unknown may be a vector of real or integer values.
- The number of unknowns may be small (from 1 to 10 - 100), medium (from 10 to 100 - 1 000) or large (from 1 000 - 10 000 and above), leading to dense or sparse linear systems.
- There may be one or several cost functions (multi-objective optimization).
- The cost function may be smooth or non-smooth.
- There may be constraints or no constraints.
- The constraints may be bounds constraints, linear or non-linear constraints.
- The cost function can be linear, quadratic or a general non linear function.

An overview of optimization problems is presented in figure 1. In this document, we will be concerned mainly with continuous parameters and problems with one objective function only. From that point, smooth and nonsmooth problems require very different numerical methods. It is generally believed that equality constrained optimization problems are easier to solve than inequality constrained problems. This is because computing the set of active constraints at optimum is a difficult problem.

The size of the problem, i.e. the number $n$ of parameters, is also of great concern with respect to the design of optimization algorithms. Obviously, the rate of convergence is of primary interest when the number of parameters is large. Algorithms which require $n$ iterations, like BFGS’s methods for example, will not be efficient if the number of iterations to perform is much less than $n$. Moreover, several algorithms (like Newton’s method for example), require too much storage to be of a practical value when $n$ is so large that $n \times n$ matrices cannot be stored. Fortunately, good algorithms such as conjugate gradient methods and BFGS with limited memory methods are specifically designed for this purpose.
Figure 1: Classes of optimization problems
1.2 What is an optimization problem?

In the current section, we present the basic vocabulary of optimization.

Consider the following general constrained optimization problem.

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

s.t. $h_i(x) = 0, \quad i = 1, m'$ (3)

$$h_i(x) \geq 0, \quad i = m' + 1, m.$$ (4)

The following list presents the name of all the variables in the optimization problem.

- The variables $x \in \mathbb{R}^n$ can be called the unknowns, the parameters or, sometimes, the decision variables. This is because, in many physical optimization problems, some parameters are constants (the gravity constant for example) and only a limited number of parameters can be optimized.

- The function $f : \mathbb{R}^n \to \mathbb{R}$ is called the objective function. Sometimes, we also call it the cost function.

- The number $m \geq 0$ is the number of constraints and the functions $h$ are the constraints function. More precisely, the functions $\{h_i\}_{i=1}^{m'}$ are the equality constraints functions and $\{h_i\}_{i=m'+1}^{m}$ are inequality constraints functions.

Some additional notations are used in optimization and the following list presents some of the most useful.

- A point which satisfies all the constraints is feasible. The set of points which are feasible is called the feasible set. Assume that $x \in \mathbb{R}^n$ is a feasible point. Then the direction $p \in \mathbb{R}^n$ is called a feasible direction if the point $x + \alpha p \in \mathbb{R}^n$ is feasible for all $\alpha \in \mathbb{R}$.

- The gradient of the objective function is denoted by $g(x) \in \mathbb{R}^n$ and is defined as

$$g(x) = \nabla f(x) = \left( \frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)^T.$$ (5)

- The Hessian matrix is denoted by $H(x)$ and is defined as

$$H_{ij}(x) = (\nabla^2 f)_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}.$$ (6)

If $f$ is twice differentiable, then the Hessian matrix is symmetric by equality of mixed partial derivatives so that $H_{ij} = H_{ji}$.

The problem of finding a feasible point is the feasibility problem and may be quite difficult itself.
A subclass of optimization problems is the problem where the inequality constraints are made of bounds. The *bound-constrained* optimization problem is the following,

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} & \quad f(x) \\
\text{s.t.} & \quad x_i, L \leq x_i \leq x_i, U, \quad i = 1, m.
\end{aligned}
$$

where \(\{x_i, L\}_{i=1,m}\) (resp. \(\{x_i, U\}_{i=1,m}\)) are the lower (resp. upper) bounds. Many physical problems are associated with this kind of problem, because physical parameters are generally bounded.

### 1.3 Rosenbrock’s function

We now present a famous optimization problem, used in many demonstrations of optimization methods. We will use this example consistently throughout this document and will analyze it with Scilab. Let us consider Rosenbrock’s function \cite{numeral} defined by

$$
f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.
$$

The classical starting point is \(x_0 = (-1.2, 1)^T\) where the function value is \(f(x_0) = 24.2\). The global minimum of this function is \(x^* = (1, 1)^T\) where the function value is \(f(x^*) = 0\). The gradient is

$$
g(x) = \begin{pmatrix}
-400x_1(x_2 - x_1^2) - 2(1 - x_1) \\
200(x_2 - x_1^2)
\end{pmatrix} = \begin{pmatrix}
400x_1^3 - 400x_1x_2 + 2x_1 - 2 \\
-200x_1^2 + 200x_2
\end{pmatrix}.
$$

The Hessian matrix is

$$
H(x) = \begin{pmatrix}
1200x_1^2 - 400x_2 + 2 & -400x_1 \\
-400x_1 & 200
\end{pmatrix}.
$$

The following *rosenbrock* function computes the function value \(f\), the gradient \(g\) and the Hessian matrix \(H\) of Rosenbrock’s function.

```plaintext
function [ f , g , H ] = rosenbrock ( x )
 f = 100.0 * (x(2) - x(1)^2)^2 + (1-x(1))^2;
g(1) = -400. * ( x(2) - x(1)**2 ) * x(1) -2. * ( 1. - x(1) )
g(2) = 200. * ( x(2) - x(1)**2 )
H(1,1) = 1200 * x(1)**2 - 400 * x(2) + 2
H(1,2) = -400 * x(1)
H(2,1) = H(1,2)
H(2,2) = 200
endfunction
```

It is safe to say that the most common practical optimization issue is related to a bad implementation of the objective function. Therefore, we must be extra careful when we develop an objective function and its derivatives. In order to help ourselves, we can use the function values which were given previously. In the following session, we check that our implementation of Rosenbrock’s function satisfies \(f(x_0) = 24.2\) and \(f(x^*) = 0\).
derivative numerical derivatives of a function.
Manages arbitrary step size.
Provides order 1, 2 or 4 finite difference formula.
Computes Jacobian and Hessian matrices.
Manages additional arguments of the function to derivate.
Produces various shapes of the output matrices.
Manages orthogonal basis change Q.

Figure 2: The derivative function.

-->x0 = [-1.2 1.0]'
x0 =
 - 1.2
  1.
-->f0 = rosenbrock ( x0 )
f0 =
  24.2
-->xopt = [1.0 1.0]'
xopt =
   1.
   1.
-->fopt = rosenbrock ( xopt )
fopt =
   0.

Notice that we define the initial and final points x0 and xopt as column vectors (as opposed to row vectors). Indeed, this is the natural orientation for Scilab, since this is in that order that Scilab stores the values internally (because Scilab was primarily developed so that it can access to the Fortran routines of Lapack). Hence, we will be consistently use that particular orientation in this document. In practice, this allows to avoid orientation problems and related bugs.

We must now check that the derivatives are correctly implemented in our rosenbrock function. To do so, we may use a symbolic computation system, such as Maple [10], Mathematica [11] or Maxima [1]. An effective way of computing a symbolic derivative is to use the web tool Wolfram Alpha [12], which can be convenient in simple cases.

Another form of cross-checking can be done by computing numerical derivatives and checking against the derivatives computed by the rosenbrock function. Scilab provides one function for that purpose, the derivative function, which is presented in figure 2. For most practical uses, the derivative function is extremely versatile, and this is why we will use it throughout this document.

In the following session, we check our exact derivatives against the numerical derivatives produced by the derivative function. We can see that the results are very close to each other, in terms of relative error.

-->[ f , g , H ] = rosenbrock ( x0 )
H =
  1330.  480.
  480.  200.
g =
\[ f = 24.2 \]

\[ \rightarrow [\text{gfd}, \text{Hfd}] = \text{derivative(rosenbrock, x0, H_form='blockmat')} \]

\[ \text{Hfd} = \\
 1330. \quad 480. \\
 480. \quad 200. \\
\]

\[ \text{gfd} = \\
 -215.6 \quad -88. \\
\]

\[ \rightarrow \text{norm(g-gfd')}/\text{norm(g)} \]

\[ \text{ans} = \\
 6.314\times10^{-11} \]

\[ \rightarrow \text{norm(H-Hfd')}/\text{norm(H)} \]

\[ \text{ans} = \\
 7.716\times10^{-09} \]

It would require a whole chapter to analyse the effect of rounding errors on numerical derivatives. Hence, we will derive here simplified results which can be used in the context of optimization.

The rule of thumb is that we should expect at most 8 significant digits for an order 1 finite difference computation of the gradient with Scilab. This is because the machine precision associated with doubles is approximately \( \epsilon_M \approx 10^{-16} \), which corresponds approximately to 16 significant digits. With an order 1 formula, an achievable absolute error is roughly \( \sqrt{\epsilon_M} \approx 10^{-8} \). With an order 2 formula (the default for the \texttt{derivative} function), an achievable absolute error is roughly \( \epsilon_M^2 \approx 10^{-11} \).

Hence, we are now confident that our implementation of the Rosenbrock function is bug-free.

### 1.4 Level sets

Suppose that we are given function \( f \) with \( n \) variables \( f(x) = f(x_1, \ldots, x_n) \). For a given \( \alpha \in \mathbb{R} \), the equation

\[ f(x) \leq \alpha, \quad (13) \]

defines a set of points in \( \mathbb{R}^n \). For a given function \( f \) and a given scalar \( \alpha \), the level set \( \mathcal{L}(\alpha) \) is defined as

\[ \mathcal{L}(\alpha) = \{ x \in \mathbb{R}^n, \ f(x) \leq \alpha \}. \quad (14) \]

Suppose that an algorithm is given an initial guess \( x_0 \) as the starting point of the numerical method. We shall always assume that the level set \( \mathcal{L}(f(x_0)) \) is bounded.

Now consider the function \( f(x) = e^x \) where \( x \in \mathbb{R} \). This function is bounded below \( (e^x > 0) \) and strictly decreasing. But its level set \( \mathcal{L}(f(x)) \) is unbounded whatever the choice of \( x \) : this is because the exp function is so that \( e^x \to 0 \) as \( x \to -\infty \).

Let us plot several level sets of Rosenbrock’s function. To do so, we can use the \texttt{contour} function. One small issue is that the \texttt{contour} function expects a function
which takes two variables $x_1$ and $x_2$ as input arguments. In order to pass to `contour` a function with the expected header, we define the `rosenbrockC` function, which takes the two separate variables $x_1$ and $x_2$ as input arguments. Then, it gathers the two parameters into one column vector and delegates the work to the `rosenbrock` function.

```latex
function f = rosenbrockC (x1, x2)
    f = rosenbrock([x1, x2]')
endfunction
```

We can finally call the `contour` function and draw various level sets of Rosenbrock’s function.

```latex
xdata = linspace(-2, 2, 100);
ydata = linspace(-1, 2, 100);
contour(xdata, ydata, rosenbrockC, [2 10 100 .. 500 1000 2000])
```

This produces the figure 3.

We can also create a 3D plot of Rosenbrock’s function. In order to use the `surf` function, we must define the auxiliary function `rosenbrockS`, which takes vectors of data as input arguments.

```latex
function f = rosenbrockS(x1, x2)
    f = rosenbrock([x1, x2])
endfunction
```

The following statements allows to produce the plot presented in figure 4. Notice that we must transpose the output of `feval` in order to feed `surf`.

```latex
x = linspace(-2, 2, 20);
y = linspace(-1, 2, 20);
Z = (feval (x, y, rosenbrockS))';
surf(x,y,Z)
```
Figure 4: Three dimensionnal plot of Rosenbrock’s function.

\[
\begin{align*}
\text{h} &= \text{gcf}(); \\
\text{cmap} &= \text{graycolormap}(10); \\
\text{h.color_map} &= \text{cmap};
\end{align*}
\]

On one hand, the 3D plot seems to be more informative than the contour plot. On the other hand, we see that the level sets of the contour plot follow the curve quadratic curve \( x_2 = x_1^2 \), as expected from the function definition. The surface has the shape of a valley, which minimum is at \( x^* = (1, 1)^T \). The whole picture has the form of a banana, which explains why some demonstrations of this test case present it as the banana function. As a matter of fact, the contour plot is often much more simple to analyze than a 3d surface plot. This is the reason why it is used more often in an optimization context, and particularly in this document. Still, for more than two parameters, no contour plot can be drawn. Therefore, when the objective function depends on more than two parameters, it is not easy to represent ourselves its associated landscape.

1.5 What is an optimum ?

In this section, we present the characteristics of the solution of an optimization problem. The definitions presented in this section are mainly of theoretical value, and, in some sense, of didactic value as well. More practical results will be presented later.

For any given feasible point \( x \in \mathbb{R}^n \), let us define \( N(x, \delta) \) the set of feasible points contained in a \( \delta \)– neighbourhood of \( x \).

A \( \delta \)– neighbourhood of \( x \) is the ball centered at \( x \) with radius \( \delta \),

\[
N(x, \delta) = \{ y \in \mathbb{R}^n, \| x - y \| \leq \delta \}. 
\] \hspace{1cm} (15)

There are three different types of minimum :
Figure 5: Different types of optimum – strong local, weak local and strong global.

- strong local minimum,
- weak local minimum,
- strong global minimum.

These types of optimum are presented in figure 5.

**Definition 1.1.** (Strong local minimum) The point $x^*$ is a strong local minimum of the constrained optimization problem 2–4 if there exists $\delta > 0$ such that the two following conditions are satisfied:

- $f(x)$ is defined on $N(x^*, \delta)$, and
- $f(x^*) < f(y)$, $\forall y \in N(x^*, \delta), y \neq x^*$.

**Definition 1.2.** (Weak local minimum) The point $x^*$ is a weak local minimum of the constrained optimization problem 2–4 if there exists $\delta > 0$ such that the three following conditions are satisfied:

- $f(x)$ is defined on $N(x^*, \delta)$,
- $f(x^*) \leq f(y)$, $\forall y \in N(x^*, \delta)$,
- $x^*$ is not a strong local minimum.

**Definition 1.3.** (Strong global minimum) The point $x^*$ is a strong global minimum of the constrained optimization problem 2–4 if there exists $\delta > 0$ such that the two following conditions are satisfied:

- $f(x)$ is defined on the set of feasible points,
- $f(x^*) < f(y)$, for all $y \in \mathbb{R}^n$ feasible and $y \neq x^*$.

Most algorithms presented in this document are searching for a strong local minimum. The global minimum may be found in particular situations, for example when the cost function is convex. The difference between weak and strong local
minimum is also of very little practical use, since it is difficult to determine what are the values of the function for all points except the computed point $x^\star$.

In practical situations, the previous definitions do not allow to get some insight about a specific point $x^\star$. This is why we will derive later in this document first order and second order necessary conditions, which are computable characteristics of the optimum.

1.6 An optimum in unconstrained optimization

Before getting into the mathematics, we present some intuitive results about unconstrained optimization.

Suppose that we want to solve the following unconstrained optimization problem.

$$\min_{x \in \mathbb{R}^n} f(x) \quad (16)$$

where $f$ is a smooth objective function. We suppose here that the hessian matrix is positive definite, i.e. its eigenvalues are strictly positive. This optimization problem is presented in figure 6, where the contours of the objective function are drawn.

The contours are the locations where the objective has a constant value. When the function is smooth and if we consider the behaviour of the function very near the optimum, the contours are made of ellipsoids: when the ellipsoid is more elongated, the eigenvalues are of very different magnitude. This behaviour is the consequence of the fact that the objective function can be closely approximated, near the optimum, by a quadratic function, as expected by the local Taylor expansion of the function. This quadratic function is closely associated with the Hessian matrix of the objective function.

1.7 Big and little O notations

The proof of several results associated with optimality require to make use of Taylor expansions. Since the development of these expansions may require to use the big and little $O$ notations, this is the good place to remind ourselves the definition of the big and little $o$ notations.
Definition 1.4. (Little o notation) Assume that $p$ is a positive integer and $h$ is a function $h : \mathbb{R}^n \to \mathbb{R}$.

We have

$$h(x) = o(\|x\|^p)$$

if

$$\lim_{x \to 0} \frac{h(x)}{\|x\|^p} = 0.$$  \hspace{1cm} (18)

The previous definition intuitively means that $h(x)$ converges faster to zero than $\|x\|^p$.

Definition 1.5. (Big o notation) Assume that $p$ is a positive integer and $h$ is a function $h : \mathbb{R}^n \to \mathbb{R}$. We have

$$h(x) = O(\|x\|^p)$$

if there exists a finite number $M > 0$, independent of $x$, and a real number $\delta > 0$ such that

$$|f(x)| \leq M\|x\|^p,$$  \hspace{1cm} (20)

for all $\|x\| \leq \delta$.

The equality 19 therefore implies that the rate at which $h(x)$ converges to zero increases as $p$ increases.

1.8 Various Taylor expansions

In this section, we will review various results which can be applied to continuously differentiable functions. These results are various forms of Taylor expansions which will be used throughout this document. We will not prove these propositions, which are topic of a general calculus course.

The following proposition makes use of the gradient of the function $f$.

Proposition 1.6. (Mean value theorem) Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable on an open set $S$ and let $x \in S$. Therefore, for all $p$ such that $x + p \in S$, there exists an $\alpha \in [0, 1]$ such that

$$f(x + p) = f(x) + p^T g(x + \alpha p).$$  \hspace{1cm} (21)

The following proposition makes use of the Hessian matrix of the function $f$.

Proposition 1.7. (Second order expansion) Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable on an open set $S$ and let $x \in S$. Therefore, for all $p$ such that $x + p \in S$, there exists an $\alpha \in [0, 1]$ such that

$$f(x + p) = f(x) + p^T g(x) + \frac{1}{2} p^T H(x + \alpha p)p.$$  \hspace{1cm} (22)
There is an alternative form of the second order expansion, which makes use of the $o(\cdot)$ or $O(\cdot)$ notations.

**Proposition 1.8.** (Second order expansion - second form) Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continously differentiable on an open set $S$ and let $x \in S$. Therefore, for all $p$ such that $x + p \in S$

$$f(x + p) = f(x) + p^T g(x) + \frac{1}{2} p^T H(x) p + O(||p||^3).$$

(23)

The last term in the previous equation is often written as $o(||p||^2)$, which is equivalent.

### 1.9 Notes and references

The classification of optimization problems suggested in section 1.1 is somewhat arbitrary. Another classification, which focus on continuous variables is presented in [4], section 1.1.1, "Classification".

Most of the definitions of optimum in section 1.5 are extracted from Gill, Murray and Wright [6], section 3.1, "Characterization of an optimum". Notice that [7] does not use the term strong local minimum, but use the term strict instead.

The notations of section 1.2 are generally used (see for example the chapter "Introduction" in [7] or the chapter "Basic Concepts" in [8]).

### 1.10 Exercises

**Exercise 1.1 (Contours of quadratic functions)** Let us consider the following quadratic function

$$f(x) = b^T x + \frac{1}{2} x^T H x,$$

(24)

where $x \in \mathbb{R}^n$, the vector $b \in \mathbb{R}^n$ is a given vector and $H$ is a definite positive symmetric $n \times n$ matrix. Compute the gradient and the Hessian matrix of this quadratic function. Plot the contours of this function. Consider the special case where

$$H = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}$$

(25)

and $b = (1, 2)^T$. Define a function which implements this function and draw its contours. Use the function derivative and the point $x = (1, 1)^T$ to check your result.

TODO : add the study of another function, Wood’s? TODO : add Powell’s!

### 2 Convexity

In this section, we present the definition and properties of convex sets and convex functions. Convexity dominates the theory of optimization and a lot of theoretical and practical optimization results can be established for these mathematical objects.

We now give a motivation for the presentation of convexity in this document. In the current section, we present a proposition stating that there is an equivalence between the convexity of a function and the positivity of its associated Hessian.
Figure 7: Convex sets - The left set is convex. The middle set is not convex, because the segment joining the two points is not inside the set. The right set is not convex because parts of the edges of the rectangle are not inside the set.

matrix. Later in this document, we will prove that if a point is a local minimizer of a convex function, then it is also a global minimizer. This is not true for a general function: to see this, simply consider that a general function may have many local minimas. In practice, if we can prove that our particular unconstrained optimization problem is associated with a convex objective function, this particularly simplifies the problem, since any stationnary point is a global minimizer.

2.1 Convex sets

In this section, we present convex sets.

We are going to analyse a very accurate definition of a convex set. Intuitively, we can simply say that a set $C$ is convex if the line segment between any two points in $C$, lies in $C$.

**Definition 2.1.** (Convex set) A set $C$ in $\mathbb{R}^n$ is convex if for every $x_1, x_2 \in C$ and every real number $\alpha$ so that $0 \leq \alpha \leq 1$, the point $x = \alpha x_1 + (1 - \alpha)x_2$ is in $C$.

Convex sets and nonconvex sets are presented in figure 7.

A point $x \in \mathbb{R}^n$ of the form $x = \theta_1x_1 + \ldots + \theta_kx_k$ where $\theta_1 + \ldots + \theta_k = 1$ and $\theta_i \geq 0$, for $i = 1, k$, is called a convex combination of the points $\{x_i\}_{i=1,k}$ in the convex set $C$. A convex combination is indeed a weighted average of the points $\{x_i\}_{i=1,k}$. It can be proved that a set is convex if and only if it contains all convex combinations of its points.

For a given set $C$, we can always define the convex hull of this set, by considering the following definition.

**Definition 2.2.** (Convex hull) The convex hull of $C$, denoted by $\text{conv}(C)$, is the set of all convex combinations of points in $C$:

$$\text{conv}(C) = \{\theta_1x_1 + \ldots + \theta_kx_k / x_i \in C, \ \theta_i \geq 0, i = 1, k, \ \theta_1 + \ldots + \theta_k = 1\} \quad (26)$$

Three examples of convex hulls are given in figure 8. The convex hull of a given set $C$ is convex. Obviously, the convex hull of a convex set is the convex set itself, i.e. $\text{conv}(C) = C$ if the set $C$ is convex. The convex hull is the smallest convex set that contains $C$.

We conclude this section by defining a cone.
Figure 8: Convex hull - The left set is the convex hull of a given number of points. The middle (resp. right) set is the convex set of the middle (resp. right) set in figure 7.

Figure 9: A cone

Definition 2.3. (Cone) A set $C$ is a cone if for every $x \in C$ and $\theta \geq 0$, we have $\theta x \in C$.

A set $C$ is a convex cone if it is convex and a cone, which means that for any $x_1, x_2 \in C$ and $\theta_1, \theta_2 \geq 0$, we have

$$\theta_1 x_1 + \theta_2 x_2 \in C.$$  \hspace{1cm} (27)

A cone is presented in figure 9.

An example of a cone which is nonconvex is presented in figure 10. This cone is nonconvex because if we pick a point $x_1$ on the ray, and another point $x_2$ in the grey area, there exists a $\theta$ with $0 \leq \theta \leq 1$ so that a convex combination $\theta x_1 + (1 - \theta) x_2$ is not in the set.

A conic combination of points $x_1, \ldots, x_k \in C$ is a point of the form $\theta_1 x_1 + \ldots + \theta_k x_k$, with $\theta_1, \ldots, \theta_k \geq 0$.

Figure 10: A nonconvex cone
The conic hull of a set $C$ is the set of all conic combinations of points in $C$, i.e.
\[
\left\{ \theta_1 x_1 + \ldots + \theta_k x_k \mid x_i \in C, \; \theta_i \geq 0, \; i = 1, k \right\}
\]  
(28)

The conic hull is the smallest convex cone that contains $C$.

Conic hulls are presented in figure 11.

### 2.2 Convex functions

**Definition 2.4.** (Convex function) A function $f : C \subset \mathbb{R}^n \rightarrow \mathbb{R}$ is convex if $C$ is a convex set and if, for all $x, y \in C$, and for all $\theta$ with $0 \leq \theta \leq 1$, we have
\[
f (\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y).
\]  
(29)

A function $f$ is strictly convex if the inequality 29 is strict.

The inequality 29 is sometimes called *Jensen’s inequality* [5]. Geometrically, the convexity of the function $f$ means that the line segment between $(x, f(x))$ and $(y, f(y))$ lies above the graph of $f$. This situation is presented in figure 12.

Notice that the function $f$ is convex only if its domain of definition $C$ is a convex set. This topic will be reviewed later, when we will analyze a counter example of this.
A function \( f \) is \textit{concave} if \(-f\) is convex. A concave function is presented in figure 13.

An example of a nonconvex function is presented in figure 14, where the function is neither convex, nor concave. We shall give a more accurate definition of this later, but it is obvious from the figure that the \textit{curvature} of the function changes.

We can prove that the sum of two convex functions is a convex function, and that the product of a convex function by a scalar \( \alpha > 0 \) is a convex function (this is trivial). The level sets of a convex function are convex.

### 2.3 First order condition for convexity

In this section, we will analyse the properties of differentiable convex functions. The two first properties of differentiable convex functions are first-order conditions (i.e. makes use of \( \nabla f = g \)), while the third property is a second-order condition (i.e. makes use of the hessian matrix \( H \) of \( f \)).

**Proposition 2.5.** (First order condition of differentiable convex function) Let \( f \) be continuously differentiable. Then \( f \) is convex over a convex set \( C \) if and only if

\[
f(y) \geq f(x) + g(x)^T(y - x)
\]

for all \( x, y \in C \).
The proposition says that the graph of a convex function is above a line which slope is equal to the first derivative of $f$. The figure 15 presents a graphical analysis of the proposition. The first order condition can be associated with the Taylor expansion at the point $x$. The proposition tells that the first order Taylor approximation is under the graph of $f$, i.e. from a local approximation of $f$, we can deduce a global behaviour of $f$. As we shall see later, this makes convex as perfect function candidates for optimization.

**Proof.** In the first part of the proof, let us assume that $f$ is convex over the convex set $C$. Let us prove that the inequality (30) is satisfied. Assume that $x, y$ are two point in $C$. Let $\theta$ be a scalar such that $0 < \theta \leq 1$ (notice that $\theta$ is chosen strictly positive). Because $C$ is a convex set, the point $\theta y + (1 - \theta)x$ is in the set $C$.

This point can be written as $x + \theta(y - x)$. The idea of the proof is to compute the value of $f$ at the two points $x + \theta(y - x)$ and $x$, and to form the factor $(f(x + \theta(y - x)) - f(x)) / \theta$. This fraction can be used to derive the inequality by making $\theta \to 0$, so that we let appear the dot product $g(x)^T(y - x)$. This situation is presented in figure 16.

By the convexity hypothesis of $f$, we deduce:

$$f(\theta y + (1 - \theta)x) \leq \theta f(y) + (1 - \theta)f(x).$$

The left hand side can be written as $f(y + \theta(x - y))$ while the right hand side of the previous inequality can be written in the form $f(x) + \theta(f(y) - f(x))$. This leads to
Figure 17: Proof of the first order condition on convex functions - part 2

the following inequality:

\[ f(x + \theta(y - x)) \leq f(x) + \theta(f(y) - f(x)). \]  \hspace{1cm} (32)

We move the term \( f(x) \) to the left hand side, divide by \( \theta > 0 \) and get the inequality:

\[ \frac{f(x + \theta(y - x)) - f(x)}{\theta} \leq f(y) - f(x). \]  \hspace{1cm} (33)

We can take the limit as \( \theta \to 0 \) since, by hypothesis, the function \( f \) is continuously differentiable. Therefore,

\[ g(x)^T(y - x) \leq f(y) - f(x), \]  \hspace{1cm} (34)

which concludes the first part of the proof.

In the second part of the proof, let us assume that the function \( f \) satisfies the inequality 30. Let us prove that \( f \) is convex.

Let \( y_1, y_2 \) be two points in \( C \) and let \( \theta \) be a scalar such that \( 0 \leq \theta \leq 1 \). The idea is to use the inequality 30, with the carefully chosen point \( x = \theta y_1 + (1 - \theta)y_2 \). This idea is presented in figure 17.

The first order condition 30 at the two points \( y_1 \) and \( y_2 \) gives:

\[ f(y_1) \geq f(x) + g(x)^T(y_1 - x) \]  \hspace{1cm} (35)
\[ f(y_2) \geq f(x) + g(x)^T(y_2 - x) \]  \hspace{1cm} (36)

We can form a convex combination of the two inequalities and get:

\[ \theta f(y_1) + (1 - \theta)f(y_2) \geq \theta f(x) + (1 - \theta)f(x) \]
\[ + g(x)^T (\theta(y_1 - x) + (1 - \theta)(y_2 - x)). \] \hspace{1cm} (37)

The previous inequality can be simplified into:

\[ \theta f(y_1) + (1 - \theta)f(y_2) \geq f(x) + g(x)^T (\theta y_1 + (1 - \theta)y_2 - x). \] \hspace{1cm} (38)

By hypothesis \( x = \theta y_1 + (1 - \theta)y_2 \), so that the inequality can be simplified again and can be written as:

\[ \theta f(y_1) + (1 - \theta)f(y_2) \geq f(\theta y_1 + (1 - \theta)y_2), \] \hspace{1cm} (40)

which shows that the function \( f \) is convex and concludes the proof.
We shall find another form of the first order condition for convexity with simple algebraic manipulations. Indeed, the inequality 30 can be written
\[ g(x)^T(y - x) \leq f(y) - f(x) \]. Multiplying both terms by \(-1\) yields
\[ g(x)^T(x - y) \geq f(y) - f(x) \]. (41)

If we interchange the role of \(x\) and \(y\) in (41), we get
\[ -g(y)^T(x - y) \geq f(y) - f(x) \]. (42)

We can now take the sum of the two inequalities (41) and (42) and finally get:
\[(g(x) - g(y))^T(x - y) \geq 0 \]. (43)

This last derivation proved one part of the following proposition.

**Proposition 2.6.** (First order condition of differentiable convex function - second form) Let \(f\) be continuously differentiable. Then \(f\) is convex over a convex set \(C\) if and only if
\[(g(x) - g(y))^T(x - y) \geq 0 \] (44)
for all \(x, y \in C\).

The second part of the proof is given as an exercise. The proof is based on an auxiliary function that we are going to describe in some detail.

Assume that \(x, y\) are two points in the convex set \(C\). Since \(C\) is a convex set, the point \(x + \theta(y - x) = \theta y + (1 - \theta)x\) is also in \(C\), for all \(\theta\) so that \(0 \leq \theta \leq 1\). Let us define the following function
\[ \phi(\theta) = f(x + \theta(y - x)) \], (45)
for all \(\theta\) so that \(0 \leq \theta \leq 1\).

The function \(\phi\) should really be denoted by \(\phi_{x,y}\), because it actually depends on the two points \(x, y \in C\). Despite this lack of accuracy, we will keep our current notation for simplicity reasons, assuming that the points \(x, y \in C\) are kept constant.

By definition, the function \(\phi\) is so that the two following equalities are satisfied:
\[ \phi(0) = f(x), \quad \phi(1) = f(y) \]. (46)

The function \(f\) is convex over the convex set \(C\) if and only if the function \(\phi\) satisfies the following inequality
\[ \phi(\theta) \geq \theta \phi(1) + (1 - \theta)\phi(0) \], (47)
for all \(x, y \in C\) and for all \(\theta\) so that \(0 \leq \theta \leq 1\). One can easily see that \(f\) is convex if and only if the function \(\phi\) is convex for all \(x, y \in C\). The function \(\phi\) is presented in figure 18.
If the function $f$ is continuously differentiable, then the function $\phi$ is continuously differentiable and its derivative is

$$
\phi'(t) = g(x + \theta(y - x))^T(y - x).
$$

The first order condition (44) is associated with the property that $\phi$ satisfies the inequality

$$
\phi'(t) - \phi'(s) \geq 0, \quad (49)
$$

for all real values $s$ and $t$ so that $t \geq s$. The inequality (49) satisfied by $\phi'$ simply states that $\phi'$ is an increasing function.

If $f$ is twice continuously differentiable, the next section will present a result stating that the second derivative of $f$ is positive. If $f$ is twice continuously differentiable, so is $\phi$ and the inequality (49) can be associated with a positive curvature of $\phi$.

### 2.4 Second order condition for convexity

In this section, we analyse the second order condition for a convex function, i.e. the condition on the Hessian of the function.

**Proposition 2.7.** (Second order condition of differentiable convex function) Let $f$ be twice continuously differentiable on the open convex set $C$. Then $f$ is convex on $C$ if and only if the Hessian matrix of $f$ is positive definite on $C$.

For a univariate function on $\mathbb{R}$, the condition simply reduces to $f''(x) \geq 0$, i.e. $f$ has a positive curvature and $f'$ is a nondecreasing function. For a multivariate function on $\mathbb{R}^n$, the condition signifies that the Hessian matrix $H(x)$ has positive eigenvalues for all $x \in C$.

Notice that the convex set is now assumed to be open. This is an technical assumption, which allows to derive a Taylor expansion of $f$ in the neighbourhood of a given point in the convex set.

Since the proof is strongly associated with the optimality conditions of an unconstrained optimization problem, we shall derive it here.

**Proof.** We assume that the Hessian matrix is positive definite. Let us prove that the function $f$ is convex. Since $f$ is twice continuously differentiable, we can use the
following Taylor expansion of \( f \). By the proposition 1.7, there exists a \( \theta \) satisfying \( 0 \leq \theta \leq 1 \) so that

\[
f(y) = f(x) + g(x^*)^T(y - x) + \frac{1}{2}(y - x)^TH(x + \theta(y - x))(y - x),
\]

for any \( x, y \in C \). Since \( H \) is positive definite, the scalar \((y-x)^TH(x+\theta(y-x))(y-x)\) is positive, which leads to the inequality

\[
f(y) \geq f(x) + g(x^*)^T(y - x).
\]

By proposition 2.5, this proves that \( f \) is convex on \( C \).

Assume that \( f \) is convex on the convex set \( C \) and let us prove that the Hessian matrix is positive definite. We assume that the Hessian matrix is not positive definite and show that it leads to a contradiction. The hypothesis that the Hessian matrix is not positive definite implies that there exists a point \( x \in C \) vector \( p \in C \) such that \( p^TH(x)p < 0 \). Let us define \( y = x + p \). By the proposition 1.7, there exists a \( \theta \) satisfying \( 0 \leq \theta \leq 1 \) so that the equality 50 holds. Since \( C \) is open and \( f \) is twice continuously differentiable, this inequality is also true in a neighbourhood of \( x \). More formally, this implies that we can choose the vector \( p \) to be small enough such that \( p^TH(x+\theta p)p < 0 \) for any \( \theta \in [0,1] \). Therefore, the equality 50 implies

\[
f(y) < f(x) + g(x^*)^T(y - x).
\]

By proposition 2.5, the previous inequality contradicts the hypothesis that \( f \) is convex. Therefore, the Hessian matrix \( H \) is positive definite, which concludes the proof.

### 2.5 Examples of convex functions

In this section, we give several examples of univariate and multivariate convex functions. We also give an example of a nonconvex function which defines a convex set.

**Example 2.1** Consider the quadratic function \( f : \mathbb{R}^n \to \mathbb{R} \) given by

\[
f(x) = f_0 + g^T x + \frac{1}{2}x^T A x,
\]

where \( f_0 \in \mathbb{R}, \ g \in \mathbb{R}^n \) and \( A \in \mathbb{R}^{n \times n} \). For any \( x \in \mathbb{R}^n \), the Hessian matrix \( H(x) \) is equal to the matrix \( A \). Therefore, the quadratic function \( f \) is convex if and only if the matrix \( A \) is positive definite. The quadratic function \( f \) is strictly convex if and only if the matrix \( A \) is strictly positive definite. We will review quadratic functions in more depth in the next section.

There are many other examples of convex functions. Obviously, any linear function is convex. The \( \exp(ax) \) function is convex on \( \mathbb{R} \), for any \( a \in \mathbb{R} \). The function \( x^a \) is convex on \( \mathbb{R} \) for any \( a > 0 \). The function \( -\log(x) \) is convex on \( \mathbb{R}_+ = \{ x > 0 \} \). The function \( x \log(x) \) is convex on \( \mathbb{R}_+ \). The following script produces the two plots presented in figure 19.
As seen in the proof of the proposition 2.7, the hypothesis that the domain of definition of \( f \), denoted by \( C \), is convex cannot be dropped. For example, consider the function \( f(x) = 1/x^2 \), defined on the set \( C = \{ x \in \mathbb{R}, x \neq 0 \} \). The second derivative of \( f \) is positive, since \( f''(x) = 6/x^4 > 0 \). But \( f \) is not a convex function.
Figure 20: The non-convex function $f(x_1, x_2) = x_1/(1 + x_2^2)$.

since $C$ is not a convex set.

Finally, let us consider an example where the function $f$ can define a convex set without being convex itself. Consider the bivariate function

$$f(x_1, x_2) = x_1/(1 + x_2^2),$$

for any $x = (x_1, x_2)^T \in \mathbb{R}^2$ The set of points satisfying the equation $f(x_1, x_2) \geq 0$ is a convex set, since it simplifies to the equation $x_1 \geq 0$. This inequality obviously defines a convex set (since the convex combination of two positive numbers is a positive number). But the function $f$ is non-convex, as we are going to see. In order to see this, the most simple is to create a 3D plot of the function. The following script produces the 3D plot which is presented in the figure 20.

```scilab
function z = f ( x1 , x2 )
    z = x2 ./(1+ x2^2)
endfunction
x = linspace ( -5 ,5 ,20);
y = linspace ( -5 ,5 ,20);
Z = ( eval3d (f,x,y)) ;
surf (x,y,Z)
h = gcf ();
cmap=graycolormap (10);
h.color_map = cmap;
```

The 3D plot suggests to search for the nonconvexity of the function on the line $x_1 = 1$, for example. On this section of the curve, we clearly see that the function is concave for $x_2 = 0$. This leads to the following Scilab session, where we define the two points $a = (1, -1)^T$ and $b = (1, 1)^T$. Then we check that the point $c = ta + (1 - t)b$ is so that the inequality $f(c) > tf(a) + (1 - t)f(b)$ holds for $t = 0.5$.

```scilab
-->a = [1 ; -1];
-->b = [1 ; 1];
```
`spec` Computes the eigenvalue and the eigenvectors.

\[ \text{D=} \text{spec(A)} \] Computes the eigenvalues \( \text{D} \) of \( \text{A} \).

\[ [\text{R, D}] = \text{spec(A)} \] Computes the right eigenvectors \( \text{R} \) of \( \text{A} \).

Figure 21: The `spec` function.

```
--> t = 0.5;
--> c = t*a + (1-t)*b;
--> fa = f(a(1),a(2));
--> fb = f(b(1),b(2));
--> fc = f(c(1),c(2));
--> [ fc t*fa +(1-t)* fb]
ans =
 1. 0.5
```

Therefore, the function \( f \) is nonconvex, although it defines a convex set.

### 2.6 Quadratic functions

In this section, we consider specific examples of quadratic functions and analyse the sign of their eigenvalues to see if a stationary point exist and is a local minimum.

Let us consider the following quadratic function

\[
    f(x) = b^T x + \frac{1}{2} x^T H x,
\]

where \( x \in \mathbb{R}^n \), the vector \( b \in \mathbb{R}^n \) is a given vector and \( H \) is a \( n \times n \) matrix.

In the following examples, we use the `spec` function which is presented in figure 21. This function allows to compute the eigenvalues, i.e. the spectrum, and the eigenvectors of a given matrix. This function allows to access to several Lapack routines which can take into account for symmetric or non-symmetric matrices.

In the context of numerical optimization, we often compute the eigenvalues of the Hessian matrix \( H \) of the objective function, which implies that we consider only real symmetric matrices. In this case, the eigenvalues and the eigenvectors are real matrices (as opposed to complex matrices).

**Example 2.2** *(A quadratic function for which the Hessian has positive eigenvalues.)*

Assume that \( b = 0 \) and the Hessian matrix is

\[
    H = \begin{pmatrix}
        5 & 3 \\
        3 & 2 
    \end{pmatrix}
\]

The eigenvalues are approximately equal to \( (\lambda_1, \lambda_2) \approx (0.1458980, 6.854102) \) and are both positive. Hence, this quadratic function is strictly convex.

The following script allows to produce the contours of the corresponding quadratic function. This produces the plot presented in figure 22.

```
function f = quadraticdefpos ( x1 , x2 )
    x = [x1 x2]';
    H = [5 3; 3 2]
    f = x.' * H * x;
```

27
endfunction
x = linspace (-2,2,100);
y = linspace (-2,2,100);
contour ( x , y , quadraticdefpos , [0.3 2 5 10 20])

The ellipsoids are typical of a strong local optimum.

In the following session, we compute the eigenvalues and the eigenvectors of the Hessian matrix. The spec function is used to compute the eigenvectors \( R \) and the diagonal eigenvalues matrix \( D \) such that \( H = R^T D R \). We can check that, because the matrix is symmetric, the spec function makes so that the eigenvectors are orthogonal and satisfy the equality \( R^T R = I \), where \( I \) is the \( n \times n \) identity matrix.

\[
\text{-->} H = \begin{bmatrix} 5 & 3 \\ 3 & 2 \end{bmatrix}
\]

\[
H =
\begin{bmatrix}
5. & 3. \\
3. & 2. \\
\end{bmatrix}
\]

\[
\text{-->} [R , D] = \text{spec}(H)
\]

\[
D =
\begin{bmatrix}
0.1458980 & 0. \\
0. & 6.854102 \\
\end{bmatrix}
\]

\[
R =
\begin{bmatrix}
0.5257311 & -0.8506508 \\
-0.8506508 & 0.5257311 \\
\end{bmatrix}
\]

\[
\text{-->} R \cdot \,^T \, R
\]

\[
\text{ans} =
\begin{bmatrix}
1. & 0. \\
0. & 1. \\
\end{bmatrix}
\]

Example 2.3 *(A quadratic function for which the Hessian has both positive and*
negative eigenvalues.) Assume that \( b = 0 \) and the Hessian matrix is

\[
H = \begin{pmatrix} 3 & -1 \\ -1 & -8 \end{pmatrix}
\]  

(57)

The eigenvalues are approximately equal to \( (\lambda_1, \lambda_2) \approx (-8.0901699, 3.0901699) \) so that the matrix is indefinite. Hence, this quadratic function is neither convex, nor concave. The following script produces the contours of the corresponding quadratic function.

```matlab
function f = quadraticsaddle (x1, x2)
    x = [x1 x2]';
    H = [3 -1; -1 -8];
    f = x.' * H * x;
endfunction

x = linspace(-2,2,100);
y = linspace(-2,2,100);
contour (x, y, quadraticsaddle, ..
    [-20 -10 -5 -0.3 0.3 2 5 10])
```

The contour plot is presented in figure 23.

The following script allows to produce the 3D plot presented in figure 24.

```matlab
function f = quadraticsaddle (x1, x2)
    x = [x1' x2']';
    H = [3 -1; -1 -8];
    y = H * x;
    n = size(y,"c")
    for i = 1 : n
        f(i) = x(:,i)' * y(:,i)
    end
endfunction

x = linspace(-2,2,20);
y = linspace(-2,2,20);
Z = (eval3d(quadraticsaddle,x,y))';
surf(x,y,Z)
h = gcf();
cmap=graycolormap(10)
h.color_map = cmap;
```

The contour is typical of a saddle point where the gradient is zero, but which is not a local minimum. Notice that the function is unbounded.

**Example 2.4 (A quadratic function for which the Hessian has a zero and a positive eigenvalue.)** Assume that \( b = 0 \) and the Hessian matrix is

\[
H = \begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}
\]  

(58)

The eigenvalues are approximately equal to \( (\lambda_1, \lambda_2) \approx (0, 5) \) so that the matrix is indefinite. Hence, this quadratic function is convex (but not strictly convex). The following script produces the contours of the corresponding quadratic function which are presented in the figure 25.

```matlab
function f = quadraticindef (x1, x2)
    x = [x1 x2]';
```
Figure 23: A saddle point – contour plot with cuts along the eigenvectors. The Hessian matrix has one negative eigenvalue and one positive eigenvalue. The function is quadratic along the eigenvectors directions.

Figure 24: A saddle point – 3D plot
Figure 25: Contour of a quadratic function – One eigenvalue is zero, the other is positive.

\[
H = \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix}
\]

\[
f = x.' * H * x;
\]

endfunction

\[
x = \text{linspace}(-2,2,100);
y = \text{linspace}(-2,2,100);
\]

contour ( x , y , quadraticindef , [0.3 2 5 10 20])

The contour is typical of a weak local optimum. Notice that the function remains constant along the eigenvector corresponding with the zero eigenvalue.

**Example 2.5** (Example where there is no stationnary point.) Assume that \( b = (1, 0)^T \) and the Hessian matrix is

\[
H = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}
\] (59)

The function can be simplified as \( f(x) = x_1 + \frac{1}{2}x_2^2 \). The gradient of the function is \( g(x) = (1, x_2)^T \). There is no stationnary point, which implies that the function is unbounded. The following script produces the contours of the corresponding quadratic function which are presented in the figure 26.

function f = quadraticincomp ( x1 , x2 )
    x = [x1 x2]'
    H = [0 0; 0 1]
    b = [1;0]
    f = x.'*b + x.' * H * x;
endfunction

\[
x = \text{linspace}(-10,10,100);
y = \text{linspace}(-10,10,100);
\]

contour ( x , y , quadraticincomp , [-10 -5 0 5 10 20])

31
2.7 Notes and references

The section 2, which focus on convex functions is inspired from ”Convex Optimization” by Boyd and Vandenberghe [5], chapter 2, ”Convex sets” and chapter 3, ”Convex functions”. A complementary and vivid perspective is given by Stephen Boyd in a series of lectures available in videos on the website of Stanford’s University.

Most of this section can also be found in ”Convex Analysis and Optimization” by Bertsekas [3], chapter 1, ”Basic Convexity Concepts”.

Some of the examples considered in the section 2.5 are presented in [5], section 3.1.5 ”Examples”. The nonconvex function \( f(x_1, x_2) = x_1/(1 + x_2^2) \) presented in section 2.5 is presented by Stephen Boyd in his Stanford lecture ”Convex Optimization I”, Lecture 5.

The quadratic functions from section 2.6 are presented more briefly in Gill, Murray and Wright [6].

The book by Luenberger [9] presents convex sets in Appendix B, ”Convex sets” and presents convex functions in section 6.4, ”Convex and concave functions”.

2.8 Exercises

Exercise 2.1 (Convex hull - 1) Prove that a set is convex if and only if it contains every convex combinations of its points.

Exercise 2.2 (Convex hull - 2) Prove that the convex hull is the smallest convex set that contains \( C \).

Exercise 2.3 (Convex function - 1) Prove that the sum of two convex functions is a convex function.

Exercise 2.4 (Convex function - 2) Prove that the level sets of a convex function are convex.
Exercise 2.5 (Convex function - 3) This exercise is associated with the proposition 2.6, which gives the second form of the first order condition of a differentiable convex function. The first part of the proof has already be proved in this chapter and the exercise is based on the proof of the second part. Let \( f \) be continously differentiable on the convex set \( C \). Prove that if \( f \) satisfies the inequality

\[
(g(x) - g(y))^T (x - y) \geq 0
\] (60)

for all \( x, y \in C \), then \( f \) is convex over the convex set \( C \).

Exercise 2.6 (Hessian of Rosenbrock’s function) We have seen in exercise 1.1 that the Hessian matrix of a quadratic function \( f(x) = b^T x + \frac{1}{2} x^T H x \) is simply the matrix \( H \). In this case the proposition 2.7 states that if the matrix \( H \) is positive definite, then it is convex. On the other hand, for a general function, there might be some points where the Hessian matrix is positive definite and some other points where it is indefinite. Hence the Hessian positivity is only local, as opposed to the global behaviour of a quadratic function. Consider Rosenbrock’s function defined by the equation 10. Use Scilab and prove that the Hessian matrix is positive definite at the point \( x = (1,1)^T \). Check that it is indefinite at the point \( x = (0,1)^T \). Make a random walk in the interval \([-2, 2] \times [-1, 2]\) and check that many points are associated with an indefinite Hessian matrix.
3 Answers to exercises

3.1 Answers for section 1.10

Answer of Exercise 1.1 (Contours of quadratic functions) By differentiating the equation 24 with respect to $x$, we get

$$g(x) = b + \frac{1}{2} H x + \frac{1}{2} x^T H.$$  \hspace{1cm} (61)

We have $x^T H = H^T x$. By hypothesis, the matrix $H$ is symmetric, which implies that $H^T = H$. Hence, we have $x^T H = H x$. This allows to simplify the equation 61 into

$$g(x) = b + H x.$$ \hspace{1cm} (62)

We can differentiate the gradient, so that the Hessian matrix is $H$.

The following function \texttt{quadratic} defines the required function and returns its function value $f$, its gradient $g$ and its Hessian matrix $H$.

\begin{verbatim}
function [f,g,H] = quadratic ( x )
    H = [ 2 1
          1 4 ];
    b = [ 1
          2 ];
    f = b' * x + 0.5 * x' * H * x;
    g = b + H * x
endfunction
\end{verbatim}

The following script creates the contour plot which is presented in figure 27.

\begin{verbatim}
function f = quadraticC ( x1 , x2 )
    f = quadratic ( [x1 x2]' )
endfunction
xdata = linspace (-2,2,100);
ydata = linspace (-2.5,1.5,100);
contour ( xdata , ydata , quadratic , [0.5 2 4 8 12] )
\end{verbatim}

In the following session, we define the point $x$ and compute the function value, the gradient and the Hessian matrix at this point.

\begin{verbatim}
--> x = [ ]
--> 1
--> 1
--> ]
x =
     1.
     1.

--> [ f , g , H ] = quadratic ( x )
H =
     2.  1.
     1.  4.

 g =
     4.
     7.

 f =
     7.
\end{verbatim}
In order to check that the computations are correct, we use the derivative function.

\[ \text{--->[gfd, Hfd] = derivative (quadratic, x, H_form='blockmat')} \]

\[
\begin{align*}
Hfd &= \\
&= \begin{bmatrix} 2 \quad 1 \\ 1 \quad 4 \end{bmatrix} \\
gfd &= \\
&= \begin{bmatrix} 4 \quad 7 \end{bmatrix}
\end{align*}
\]

We finally compute the relative error between the computed gradient and Hessian and the finite difference formulas.

\[ \text{--> norm(g-gfd')/norm(g)} \]
\[
\text{ans} = 3.435D-12
\]

\[ \text{--> norm(H-Hfd)/norm(H)} \]
\[
\text{ans} = 0
\]

The relative error for the gradient indicates that there are approximately 12 significant digits. Therefore, our gradient is accurate. The Hessian matrix is exact.

3.2 Answers for section 2.8

Answer of Exercise 2.1 (Convex hull - 1)

Before really detailing the proof, we can detail an auxiliary result, which will help us in the design of the proof. We are going to prove that a convex combination of 2 points can be combined with a third point so that the result is a convex combination of the 3 points. Let us suppose that \( C \) is a convex set et let us assume that three points \( \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \) are in \( C \). Let us assume that \( \mathbf{x}_2 \) is a convex combination of \( \mathbf{x}_1 \) and \( \mathbf{x}_2 \), i.e.

\[
\mathbf{x}_2 = \theta_2 \mathbf{x}_1 + (1 - \theta_2) \mathbf{x}_2, \tag{63}
\]
with $0 \leq \theta_2 \leq 1$. Let us define $x_3$ as a convex combination of $x_2$ and $x_3$, i.e.

$$x_3 = \theta_3 x_2 + (1 - \theta_3) x_3,$$  \hspace{1cm} (64)

with $0 \leq \theta_3 \leq 1$. This situation is presented in 28. We shall prove that $x_3$ is a convex combination of $x_1, x_2, x_3$. Indeed, we can develop the equation for $x_3$ so that all the points $x_1, x_2, x_3$ appear:

$$x_3 = \theta_3 \theta_2 x_1 + \theta_3 (1 - \theta_2) x_2 + (1 - \theta_3) x_3,$$  \hspace{1cm} (65)

The weights of the convex combination are $\theta_3 \theta_2 \geq 0$, $\theta_3 (1 - \theta_2) \geq 0$ and $1 - \theta_3 \geq 0$. Their sum is equal to 1, as proved by the following computation:

$$\theta_3 \theta_2 + \theta_3 (1 - \theta_2) + 1 - \theta_3 = \theta_3 (\theta_2 + 1 - \theta_2) + 1 - \theta_3$$  \hspace{1cm} (66)

$$= \theta_3 + 1 - \theta_3$$  \hspace{1cm} (67)

$$= 1,$$  \hspace{1cm} (68)

which concludes the proof.

Let us prove that a set is convex if and only if it contains every convex combinations of its points.

It is direct to prove that a set which contains every convex combinations of its points is convex. Indeed, such a set contains any convex combination of two points, which implies that the set is convex.

Let us prove that a convex set is so that every convex combination of its points is convex. The proof can be done by induction on the number $k$ of points in the set $C$. The definition of the convex set $C$ implies that the proposition is true for $k = 2$. Assume that the hypothesis is true for $k$ points, and let us prove that every convex combination of $k + 1$ points is in the set $C$. Let $\{x_i\}_{i=1,k+1} \subseteq C$ and let $\{\theta_i\}_{i=1,k+1} \in \mathbb{R}$ be positive scalars such that $\theta_1 + \ldots + \theta_k + \theta_{k+1} = 1$. Let us prove that

$$x_{k+1} = \theta_1 x_1 + \ldots + \theta_k x_k + \theta_{k+1} x_{k+1} \in C.$$  \hspace{1cm} (69)

All the weights $\{\theta_i\}_{i=1,k+1}$ cannot be equal to 0, because they are positive and their sum is 1. Without loss of generality, suppose that $0 < \theta_{k+1} < 1$ (if not, simply reorder the points $\{x_i\}_{i=1,k+1}$). Therefore $1 - \theta_{k+1} > 0$. Then let us write $x_{k+1}$ in the following form

$$x_{k+1} = (1 - \theta_{k+1}) \left( \frac{\theta_1}{1 - \theta_{k+1}} x_1 + \ldots + \frac{\theta_k}{1 - \theta_{k+1}} x_k \right) + \theta_{k+1} x_{k+1}$$  \hspace{1cm} (70)

$$= (1 - \theta_{k+1}) x_k + \theta_{k+1} x_{k+1},$$  \hspace{1cm} (71)

with

$$x_k = \frac{\theta_1}{1 - \theta_{k+1}} x_1 + \ldots + \frac{\theta_k}{1 - \theta_{k+1}} x_k.$$  \hspace{1cm} (72)
The point $\pi_{k+1}$ is a convex combination of $k$ points because the weights are all positive and their sum is equal to one:

$$\frac{\theta_1}{1-\theta_{k+1}} + \ldots + \frac{\theta_k}{1-\theta_{k+1}} = \frac{1}{1-\theta_{k+1}} (\theta_1 + \ldots + \theta_k) = \frac{1}{1-\theta_{k+1}} \quad (73)$$

$$= 1 - \theta_{k+1} \quad (74)$$

$$= 1. \quad (75)$$

Therefore $\pi_{k+1}$ is a convex combination of two points which are in the set $C$, with weights $1 - \theta_{k+1}$ and $\theta_{k+1}$. Since $C$ is convex, that implies that $\pi_{k+1} \in C$, which concludes the proof.

**Answer of Exercise 2.2 (Convex hull - 2)** Let us prove that the convex hull is the smallest convex set that contains $C$.

There is no particular difficulty in this proof since only definitions are involved. We must prove that if $B$ is any convex set that contains $C$, then $\text{conv}(C) \subseteq B$. The situation is presented in figure 29. Assume that $B$ is a convex set so that $C \subseteq B$. Suppose that $x \in \text{conv}(C)$. We must prove that $x \in B$. But definition of the convex hull, there exist points $\{x_i\}_{i=1,k} \in C$, and there exist $\{\theta_i\}_{i=1,k}$ so that $\theta_i \geq 0$ and $\theta_1 + \ldots + \theta_k = 1$. Since the points $\{x_i\}_{i=1,k}$ are in $C$, and since, by hypothesis, $C \subseteq B$, therefore the points $\{x_i\}_{i=1,k} \in B$. By hypothesis, $B$ is convex, therefore $x$, which is a convex combination of points in $B$, is also in $B$, i.e. $x \in B$.

**Answer of Exercise 2.3 (Convex function - 1)** Let us prove that the sum of two convex functions is a convex function. Assume that $f_1, f_2$ are two convex functions on the convex set $C$ and consider the sum of the two functions $f = f_1 + f_2$. Assume that $x_1, x_2 \in C$. The following inequality, where $\theta$ satisfies $0 \leq \theta \leq 1$,

$$f(\theta x + (1 - \theta)x) = f_1(\theta x_1 + (1 - \theta)x_2) + f_2(\theta x_1 + (1 - \theta)x_2) \leq \theta (f(x_1) + f_2(x_2)) + (1- \theta) (f(x_1) + f_2(x_2)) \quad (76)$$

$$\leq \theta f(x_1) + (1 - \theta)f(x_2) \quad (77)$$

concludes the proof.

**Answer of Exercise 2.4 (Convex function - 2)** Let us prove that the level sets of a convex function are convex.

Assume that $f$ is a convex function on the convex set $C \subset \mathbb{R}^n$. The $\alpha$ be a given real scalar and let us consider the level set associated with $\alpha$ is defined by $L(\alpha) = \{ x \in C, f(x) \leq \alpha \}$. Let $x_1, x_2$ be two points in $L(\alpha)$, let $\theta$ be a scalar so that $0 \leq \theta \leq 1$. Let us prove that $x = \theta x_1 + (1 - \theta)x_2$ is in $L(\alpha)$, i.e. let us prove that $f(x) \leq \alpha$. To do so, we compute $f(x)$ and use the convexity of $f$ to get

$$f(x) = f(\theta x_1 + (1- \theta)x_2) \quad (78)$$

$$\leq \theta f(x_1) + (1- \theta)f(x_2). \quad (79)$$

37
We additionnally use the two inequalities \( f(x_1) \leq \alpha \) and \( f(x_2) \leq \alpha \) and obtain:

\[
\begin{align*}
  f(x) & \leq \theta \alpha + (1 - \theta) \alpha \\
  & \leq \alpha,
\end{align*}
\]

which concludes the proof.

**Answer of Exercise 2.5 (Convex function - 3)** Let \( f \) be continuosly differentiable on the convex set \( C \). Let us prove that if \( f \) satisfies the inequality

\[
(g(x) - g(y))^T (x - y) \geq 0
\]

for all \( x, y \in C \), then \( f \) is convex over the convex set \( C \). This proof is given in a slightly more complex form in [2], chapter 10 "Conditions d’optimalité", section 10.1.2 "Différentiabilité".

Assume that \( x, y \) are two points in the convex set \( C \). Since \( C \) is a convex set, the point \( x + \theta(y - x) = \theta y + (1 - \theta)x \) is also in \( C \), for all \( \theta \) so that \( 0 \leq \theta \leq 1 \). Let us define the following function

\[
\phi(\theta) = f(x + \theta(y - x)),
\]

for all \( \theta \) so that \( 0 \leq \theta \leq 1 \).

The idea of the proof is based on the fact that the convexity of \( \phi \) with respect to \( \theta \in [0, 1] \) is equivalent to the convexity of \( f \) with respect to \( x \in C \). We will prove that

\[
\phi'(t) - \phi'(s) \geq 0,
\]

for all real values \( s \) and \( t \) so that \( t \geq s \). That inequality means that \( \phi' \) is an increasing function, which property is associated with a convex function. Then we will integrate the inequality over well chosen ranges of \( s \) and \( t \) so that the convexity of \( \phi \) appear.

To prove the inequality 84, let us define two real values \( s, t \in \mathbb{R} \). We apply the inequality 82 with the two points \( x + t(y - x) \) and \( x + s(y - x) \). By the convexity of \( C \), these points are in \( C \) so that we get

\[
[g(x + t(y - x)) - g(x + s(y - x))]^T (x + t(y - x) - x - s(y - x)) \geq 0.
\]

The left hand side can be simplified into \((t - s)(\phi'(t) - \phi'(s)) \geq 0 \). For \( t \geq s \), the previous inequality leads to 84.

We now integrate the inequality 84 for \( t \in [\theta, 1] \) and \( s \in [0, \theta] \) so that the order \( t \geq s \) is satisfied and we get

\[
\int_{t \in [\theta, 1]} \int_{s \in [0, \theta]} (\phi'(t) - \phi'(s)) dtds \geq 0.
\]

We first compute the following two integrals

\[
\int_{s \in [0, \theta]} \int_{t \in [\theta, 1]} \phi'(t) dtds = \int_{s \in [0, \theta]} (\phi(1) - \phi(\theta)) ds
\]

\[
= \theta(\phi(1) - \phi(\theta)),
\]

and

\[
\int_{t \in [\theta, 1]} \int_{s \in [0, \theta]} \phi'(s) dtds = \int_{t \in [\theta, 1]} (\phi(\theta) - \phi(0)) dt
\]

\[
= (1 - \theta)(\phi(\theta) - \phi(0)).
\]

By plugging the two integrals 88 and 90 into the inequality 86, we get

\[
\theta(\phi(1) - \phi(\theta)) - (1 - \theta)(\phi(\theta) - \phi(0)) \geq 0,
\]

which simplifies to

\[
\theta \phi(1) + (1 - \theta)\phi(0) - \phi(\theta) \geq 0.
\]
which proves that $\phi$ is convex. If we plug the definition 83 of $\phi$ and the equalities $\phi(0) = f(x)$ and $\phi(1) = f(y)$, we find the convexity inequality for $f$

$$\theta f(y) + (1 - \theta) f(x) \leq f(\theta y + (1 - \theta)x),$$

which concludes the proof.

**Answer of Exercise 2.6 (Hessian of Rosenbrock’s function)** Let us prove that the Hessian matrix of Rosenbrock’s function is positive definite at the point $x = (1,1)^T$. In the following script, we define the point $x$ and use the `rosenbrock` function to compute the associated Hessian matrix. Then we use the `spec` function in order to compute the eigenvalues.

```matlab
-->x = [1 1];
-->[ f , g , H ] = rosenbrock ( x );
-->D = spec ( H )
D =
0.3993608
1001.6006
```

We see that both eigenvalues are strictly positive, although the second is much larger than the first in magnitude. Hence, the matrix $H(x)$ is positive definite for $x = (1,1)^T$. Let us check that it is indefinite at the point $x = (0,1)^T$.

```matlab
-->x = [0 1];
-->[ f , g , H ] = rosenbrock ( x );
-->D = spec ( H )
D =
 398.
 200.
```

We now see that the first eigenvalue is negative while the second is positive. Hence, the matrix $H(x)$ is indefinite for $x = (0,1)^T$. Let us make a random walk in the interval $[-2,2] \times [-1,2]$ and check that many points are associated with an indefinite Hessian matrix. In the following script, we use the `rand` function to perform a random walk in the required interval. To make so that the script always performs the same computation, we initialize the seed of the random number generator to zero. Then we define the lower and the upper bound of the simulation. Inside the loop, we generate a uniform random vector $t$ in the interval $[0,1]^2$ and scale it with the `low` and `upp` arrays in order to produce points in the target interval. We use particular formatting rules for the `mprintf` function in order to get the numbers aligned.

```matlab
rand("seed",0);
ntrials = 10;
low = [ -2 -1 ] ;
upp = [ 2 2 ] ;
for it = 1 : ntrials
    t = rand (2,1);
x = (upp - low) .* t + low;
[ f , g , H ] = rosenbrock ( x );
D = spec ( H );
mprintf("( %3d ) x=[%+5f %+5f], D =[% +7.1 f % +7.1 f]\n",.. it,x(1),x(2),D(1),D(2))
end
```

The previous script produces the following output.

- (1) $x = [-1.154701, +1.268132]$, $D = [ +4.4, +1290.4]$.
- (2) $x = [-1.999115, -0.009019]$, $D = [ +65.0, +4936.4]$.
- (3) $x = [+0.661524, +0.885175]$, $D = [ -78.4, +451.5]$.
- (4) $x = [+1.398981, +1.057193]$, $D = [ +34.6, +2093.1]$.
- (5) $x = [+1.512866, -0.794878]$, $D = [ +77.5, +3189.0]$.
- (6) $x = [+0.243394, +0.987071]$, $D = [ -339.3, +217.6]$.
By increasing the number of trials, we see that the first eigenvalue is often negative.

References


Index

concave, 18
cone, 16
constraints, 6
convex combination, 16
convex cone, 17
convex function, 18
convex hull, 16
convex sets, 16
cost, 6
feasible, 6
Hessian, 6
Jensen’s inequality, 18
level set, 9
objective, 6
Rosenbrock H.H., 7, 9
saddle point, 29
strong global minimum, 12
strong local minimum, 12
Taylor expansions, 14
weak local minimum, 12