1. Tutorial

Exercise 1 (Coercive functions and extrema)**.** A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be *coercive* if

$$\lim_{\|x\|\to+\infty} f(x) = +\infty.$$

Show that if f is continuous and coercive then f admits at least one minimum on $\mathbb{R}^n$.

Exercise 2 (Necessary condition)**.** Let f be a differentiable numerical function on an open set U of $\mathbb{R}^n$. Show that if $a \in U$ is a local minimum of f then $\nabla f(a) = 0$.

Exercise 3 (Convex functions and extrema)**.** Let f be a numerical convex function on a convex open set U of $\mathbb{R}^n$. If f is differentiable in $a \in U$ and if $\nabla f(a) = 0$, show that f admits a global minima in a on U. We now suppose that f is strictly convex. Show that the minimum is unique.
Hint: we can use the fact that f (differentiable) is convex on C if for all $x, y \in C$, $f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle$.

Exercise 4 (Calculation of extrema)**.** Consider the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$f(x, y) = 3x^2 + 2y^2 + 2xy + x + y + 10.$$

1. Is this function convex? Justify.
2. We consider the optimization problem $\inf_{(x,y)\in\mathbb{R}^2} f(x, y)$. What can we say about this problem?
3. Solve it.

Exercise 5 (Unimodal function)**.** A function $f : \mathbb{R} \to \mathbb{R}$ is said to be unimodal on the interval $[a, b]$ if it has a unique local minimum on $[a, b]$. Show that a continuous unimodal function is strictly decreasing until the minimum and strictly increasing after the minimum.

Exercise 6 (Characterization of convexity)**.** Let C be a non empty open convex subset of $\mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ a differentiable function on C. Show that the following propositions are equivalent:

1. f is convex on C;
2. for all $x, y \in C$, $f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle$;
3. the application ∇f is monotone on C, that is

$$\forall x, y \in C, \qquad \langle \nabla f(y) - \nabla f(x), y - x\rangle \geq 0.$$

Show that if, in addition, f is twice differentiable on $\mathbb{R}^n$, then

$$f \text{ is convex on } \mathbb{R}^n \quad \Leftrightarrow \quad \forall x \in \mathbb{R}^n, \quad \nabla^2 f(x) \text{ is positive semidefinite.}$$

Exercise 7 (Quadratic function). Let $f : \mathbb{R}^n \to \mathbb{R}$ be defined by $f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$ where A is a symmetric matrix of size $n \times n$ and $b \in \mathbb{R}^n$.

1. Show that $\nabla f(x) = Ax - b$.
2. Deduce the Hessian matrix $H_f(x)$.
3. Propose an optimization algorithm to solve a linear system $Ax = b$ when A is symmetric positive definite.

Exercise 8 (Optimization). Let $n \geq 2$ be a natural number. Consider the application $f : \mathbb{R}^n \to \mathbb{R}$ defined by

$$f(x_1, x_2, \dots, x_n) = \sum_{k=1}^{n} x_k^2 + \Big(\sum_{k=1}^{n} x_k\Big)^2 - \sum_{k=1}^{n} x_k.$$

1. Justify that f is of class $\mathcal{C}^2$ on $\mathbb{R}^n$ and calculate the gradient ∇f as well as the Hessian matrix H_f.
2. Determine the only critical point $\overline{x} = (\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)$ of f on $\mathbb{R}^n$.
3. We wish to prove that $\overline{x}$ is a local minimum of f.
   a. Check that the Hessian matrix $H_f(\overline{x})$ can be written as $H_f(\overline{x}) = 2(I_n + J_n)$ where I_n is the identity matrix of size n and J_n is the matrix of size n whose coefficients are equal to 1.
   b. Determine the rank of J_n. Deduce that 0 is an eigenvalue of J_n. Determine the dimension of the associated eigenspace.
   c. Calculate the product of J_n by the vector $(1, \dots, 1)^T$. Deduce another eigenvalue of J_n.
   d. Deduce the eigenvalues of $H_f(\overline{x})$ and conclude about the nature of the point $\overline{x}$.

Exercise 9 (Least Squares). Given n points (x_i, y_i) of $\mathbb{R}^2$ with x_i not all equal to each other, show that there are unique numbers λ and μ, which minimize the sum

$$\sum_{i=1}^{n} (\lambda x_i + \mu - y_i)^2.$$

Exercise 10 (Hadamard's inequality). We provide the space $E = \mathbb{R}^n$ with the usual scalar product. We denote by $f(v_1, \dots, v_n)$ the determinant of the matrix $n \times n$ of column vectors $v_1, \dots, v_n \in E$.

1. Show that the maximum of f on the set X defined by

   $$\|v_1\| = \cdots = \|v_n\| = 1$$

   is reached and is strictly positive.
2. Show using Lagrange multiplier that if the maximum is reached in $(v_1, \dots, v_n)$, then the v_i form an orthonormal basis of E.
3. Prove that Hadamard's inequality:

   $$|\det(v_1, \dots, v_n)| \leq \|v_1| \cdots \|v_n|,$$

   for any vector $v_1, \dots, v_n$. When do we have equality?

Exercise 11 (Choleski decomposition). Let A be a symmetric positive definite matrix of size n. Let $A = LL^T$ be its Choleski decomposition (L being lower triangular).

1. For $n = 3$, write a_{ij} for $i, j = 1, 2, 3$ as a function of the coefficients of L.

2. Note that if we calculate column by column, we can then calculate in order l_{11}, l_{21}, l_{31}, l_{22}, l_{32} and l_{33}. Write a MATLAB function calculating the Choleski decomposition for any value of n.

3. Compute the Choleski decomposition of the matrix

$$A = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 5 & 1 \\ 1 & 1 & 3 \end{pmatrix}.$$

2. Practical

Exercise 12 (Golden section search)**.**

1. Write a MATLAB program implementing the Golden section search. Your program must take a function and an interval as parameters. The search should continue until the desired accuracy but should not exceed 100 iterations.

2. Write a MATLAB program implementing Newton's method.

3. Test your algorithms on the following examples. You will compare your results with those given by the `fminbnd` function of MATLAB.

   a) $f(x) = \sin(x)$ on $[0, \pi/2]$
   b) $f(x) = (\arctan x)^2$ on $[-1, 1]$
   c) $f(x) = |\ln(x)|$ on $[1/2, 4]$
   d) $f(x) = |x|$ on $[-1, 1]$

Exercise 13 (Rosenbrock's function and Newton's method)**.** The Rosenbrock function is a non-convex function of two variables used as a test for mathematical optimization problems. It was introduced by Rosenbrock in 1960. It is defined by

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

1. Compute the gradient $g(x)$ and the Hessian $H(x)$ of the function f (we will use the Symbolic Math Toolbox).

2. Check that $x^* = [1, 1]^T$ is a local minimum of f.

3. Compute the first 5 iterates of Newton's method for minimizing f starting with $x_0 = [-1, -2]^T$. Draw the level lines of the function f using `ezcontour` in the domain `[-1.5;2;-3;3]`. Display the iterates on the same graph.

4. Compute the norm of the error $\|x - x^*\|$ at each iteration and determine if the convergence rate is quadratic.

Exercise 14 (Optimal step gradient method and Wolfe's method)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be a real valued function of n variables. The constant step gradient method consists in computing the iterations

$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

where α is a constant.

1. Implement the gradient method with $\alpha = 1$. Test your program on the function $f(x) = x_1^2 + 2x_2^2$ starting from $x_0 = (-1, -1)$. Test for several steps of descent: for example $\alpha = 0.1$, $\alpha = 0.1$, $\alpha = 0.5$ and $\alpha = 1$. Comment on this.

2. Do the same with the Rosenbrock function starting for example in $x_0 = (-1, 1.2)$ and $\alpha = 0.001$.

3. In the optimal step gradient method, we look for α_k such that

$$\min_{\alpha_k \geq 0} f(x_k - \alpha_k \nabla f(x_k)).$$

To find α_k, we will use Wolfe's method. Let $g(t) = f(x + td)$. where d is a direction of descent. Given $t \in \mathbb{R}^+$, the Wolfe's linear search method consists in narrowing a confidence interval $[t_g, t_d]$ in which we choose a t that we test.

- Initially, $t_g = 0$, $t_d = +\infty$ and $t = 1$, $m_1 = 0.1$, $m_2 = 0.9$
- if $g(t) \leq g(0) + m_1 t g'(0)$ and $g'(t) \geq m_2 g'(0)$ then stop
- if $g(t) > g(0) + m_1 t g'(0)$ then let $t_d = t$, $t_g = t_g$ and $t = (t_d + t_g)/2$ (if $t_d = +\infty$ then $t = 10t_g$)
- if $g(t) \leq g(0) + m_1 t g'(0)$ and $g'(t) < m_2 g'(0)$ then $t_g = t$, $t_d = t_d$ and $t = (t_d + t_g)/2$ (if $t_d = +\infty$ then $t = 10t_g$)

Implement the optimal step gradient method with Wolfe's method. Test your implementation on the Rosenbrock function.

Exercise 15 (Nelder-Meade algorithm)**.** The Nelder-Mead method is a nonlinear optimization algorithm that was proposed by John Nelder and Roger Mead in 1965. It is a numerical heuristic method that tries to minimize a continuous function in a multidimensional space.

1. Choice of $N+1$ points of the N-dimensional space of the unknowns, forming a simplex: $\{x_1, x_2, \ldots, x_{N+1}\}$,
2. Compute the values of the function f at these points, sort the points so as to have $f(x_1) \leq f(x_2) \leq \cdots \leq f(x_{N+1})$. In fact, it is enough to know the first and the last two.
3. Compute x_0, center of gravity of all points except x_{N+1}.
4. Compute $x_r = x_0 + (x_0 - x_{N+1})$ (reflection of x_{N+1} from x_0).
5. If $f(x_r) < f(x_N)$, compute $x_e = x_0 + 2(x_0 - x_{N+1})$ (simplex expansion). If $f(x_e) < f(x_r)$, replace x_{N+1} by x_e, otherwise, replace x_{N+1} by x_r. Return to step 2.
6. If $f(x_N) < f(x_r)$, compute $x_c = x_{N+1} + 1/2(x_0 - x_{N+1})$ (simplex contraction). If $f(x_c) \leq f(x_N)$, replace x_{N+1} by x_c and return to step 2, otherwise go to step 7.
7. Shrink toward x_1: replace x_i by $x_1 + 1/2(x_i - x_1)$ for $i \geq 2$. Return to step 2.

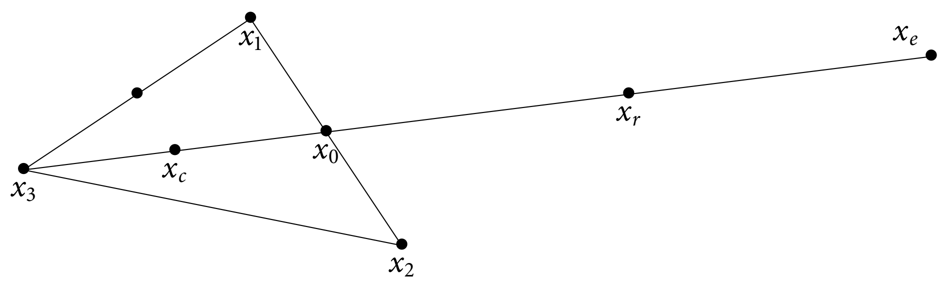

Figure 1: Nelder-Meade algorithm

1. Implement the Nelder-Meade algorithm.

2. Test your code on the Rosenbrock function.

3. Compare your result with the MATLAB command `fminsearch`.