People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Djilali Bounaâma-Khemis Miliana University

Faculty of Material Sciences and Computer Science

Department of Mathematics

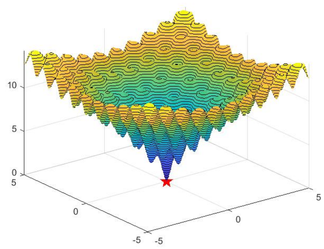
Unconstrained Optimization

Theory, Algorithms, Exercises, and Practical Training

For Third-Year L_3 Mathematics Students

DSC :22-09-2025

FSC :25-09-2025



Dr : Mohammed BOUKEDROUN

Email : m.boukedroun@univ-dbkm.dz

2025-2026

Contents

General Introduction to Unconstrained Optimization

Optimization is a fundamental pillar of applied mathematics, engineering, and the sciences, concerned with finding the best possible solution to a given problem according to specific criteria. Among the many categories of optimization, unconstrained optimization occupies a central position, as it deals with the process of minimizing or maximizing an objective function without any explicit restrictions on the decision variables. In other words, it focuses on optimizing functions defined over an entire space without any constraints limiting feasible solutions. As emphasized by Nocedal and Wright [2], unconstrained optimization serves as the foundational basis for the entire field of optimization, since most constrained optimization methods require solving a sequence of unconstrained subproblems.

The primary goal in unconstrained optimization is to determine the point at which the objective function achieves its minimum or maximum value. In practical terms, this involves adjusting the input variables so that the function reaches its optimal value, a concept that underlies numerous practical applications. For instance, in engineering design, minimizing the weight of a structural component while maintaining sufficient strength often starts with an unconstrained optimization formulation before additional real-world constraints are imposed [3]. Similarly, in economics and finance, maximizing utility or profit functions in idealized models often begins with unconstrained optimization before incorporating market restrictions [18].

Historically, unconstrained optimization problems have been studied extensively due to their mathematical elegance and tractability. Classical approaches, such as gradient-based methods, Newton's method, and quasi-Newton algorithms, emerged as efficient solutions for smooth and differentiable functions. Fletcher [5] notes that these methods exploit first and second derivative information to iteratively approach optimal solutions with high precision, making them suitable for a wide range of scientific and engineering computations. Moreover, Nelder and Mead [9] developed direct search methods, like the simplex algorithm, which do not require derivative information and are widely used for optimizing non-differentiable or noisy functions.

The importance of unconstrained optimization has further increased with the rise of data science and machine learning. Training modern neural networks relies on minimizing loss functions, which are typically formulated as large-scale unconstrained optimization problems [6]. Techniques such as Stochastic Gradient Descent (SGD) and adaptive methods like Adam [7] were developed to handle optimization problems with millions of parameters and vast datasets, demonstrating the relevance of unconstrained optimization in contemporary computational fields.

Beyond its practical applications, unconstrained optimization also provides fundamental theoretical insights. Concepts such as gradient, Hessian, and convexity originated and were rigorously studied within the framework of unconstrained problems before being generalized to handle constraints and non-convexities in more complex scenarios. As stated by Bertsekas [4], understanding unconstrained optimization is essential for anyone aiming to grasp the general principles and algorithms of nonlinear programming.

Recent research continues to improve the scalability, robustness, and convergence of algorithms in unconstrained optimization. Methods such as the limited-memory BFGS [8] have been particularly impactful for large-scale problems where traditional quasi-Newton methods become computationally infeasible. Furthermore, optimization software libraries widely used in academia and industry, including MATLAB, SciPy, and TensorFlow, integrate efficient unconstrained optimization algorithms as core modules to solve diverse real-world problems efficiently.

In conclusion, unconstrained optimization remains a fundamental area of optimization theory and practice. Its algorithms, models, and insights are embedded in modern computational tools and methods across disciplines. Whether for solving idealized mathematical problems, performing preliminary engineering designs, or training advanced machine learning models, the role of unconstrained optimization is indispensable. As Nocedal and Wright [2] emphasize, the continued advancement of optimization algorithms, especially for unconstrained problems, drives progress across the entire spectrum of numerical optimization and its vast applications.

Finally, in this polycopé, we will present three major chapters to build a comprehensive understanding of unconstrained optimization. The first chapter is devoted to essential reminders of differential calculus and convexity, providing the theoretical tools required for optimization analysis. The second chapter focuses on unconstrained minimization, covering fundamental concepts and formulations. The third chapter introduces the main methods and algorithms used to solve unconstrained optimization problems efficiently. We conclude the polycopé with a collection of exercises and practical works (TP) to consolidate the theoretical knowledge and enhance problem-solving skills in real-world contexts.

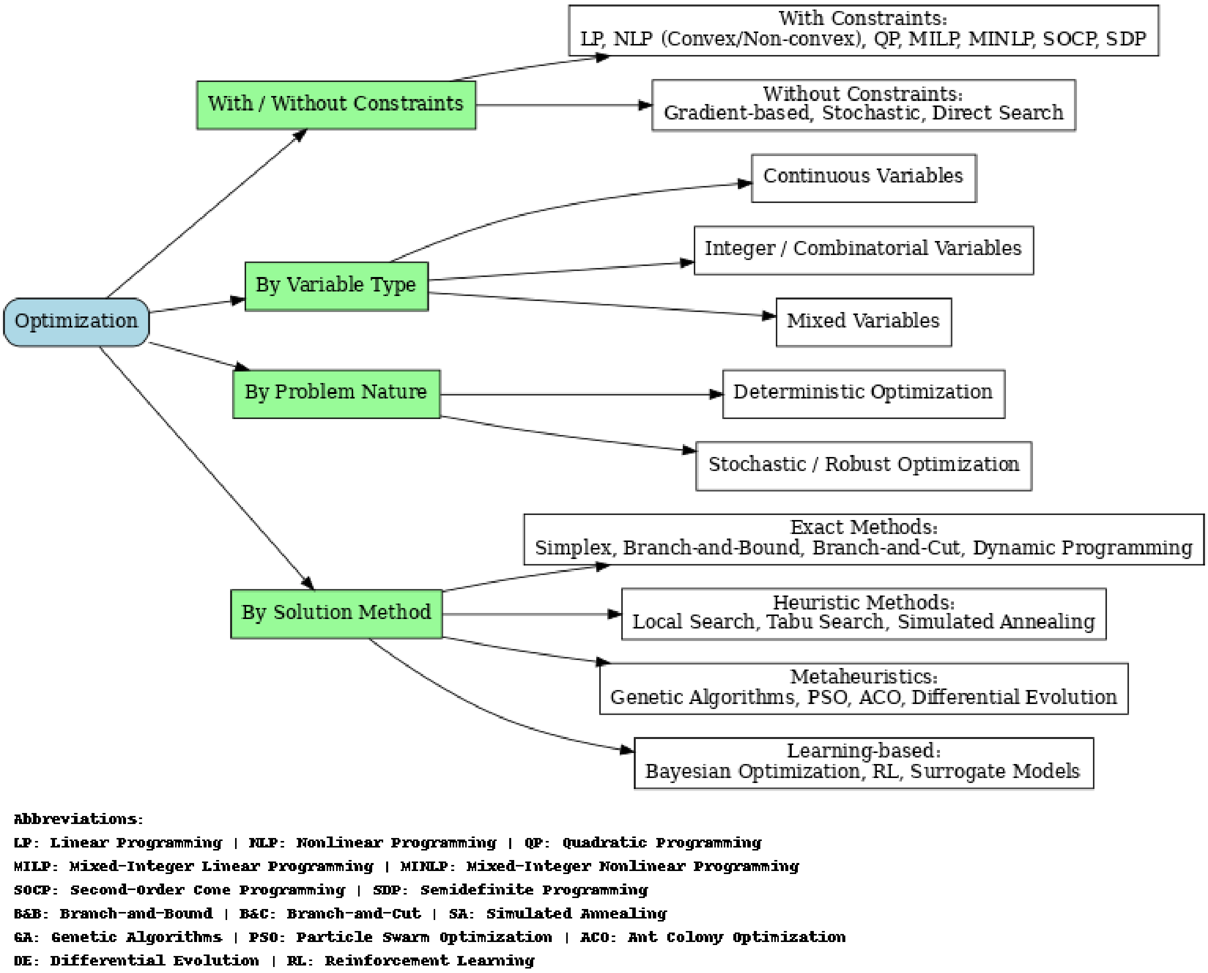


Figure 1: Comprehensive diagram of the main types of optimization and their relationships, with abbreviations explained below the figure.

Chapter 1

Differential Calculus and Convexity

1.1 Review of Differential Calculus

1.1.1 Normed Vector Spaces

Let V be a vector space over the field $\mathbb{R}$.

Definition 1.1.1. *An* ***inner product*** *on V is a mapping*

$$\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$$

that is bilinear, symmetric, and positive definite, meaning it satisfies:

1. $\langle u, \cdot \rangle : V \to \mathbb{R}$ *is linear for all* $u \in V$,
2. $\langle \cdot, v \rangle : V \to \mathbb{R}$ *is linear for all* $v \in V$,
3. $\langle u, v \rangle = \langle v, u \rangle$ *for all* $u, v \in V$,
4. $\langle v, v \rangle = 0 \Leftrightarrow v = 0$, *and* $\langle v, v \rangle \geq 0$ *for all* $v \in V$.

1.1.2 Directional Derivative

Definition 1.1.2 (Directional Derivative [18, Section 2.1])**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable at $x \in \mathbb{R}^n$. The directional derivative of f at x in the direction $d \in \mathbb{R}^n$ is defined as:*

$$D_f(x; d) = \lim_{t \to 0} \frac{f(x + td) - f(x)}{t}.$$

If f is differentiable at x, then $D_f(x; d) = \nabla f(x)^T d$.

Example 1.1.1. *Let $f(x, y) = x^2 + y^2$. Compute the directional derivative at point* $(1, 2)$ *in the direction $d =$* $(3, 4)$. *Solution:* $\nabla f(x, y) = (2x, 2y)$. *At* $(1, 2)$, $\nabla f(1, 2) = (2, 4)$. *Thus:*

$$D_f((1, 2); (3, 4)) = (2, 4) \cdot (3, 4) = 6 + 16 = 22.$$

—

1.1.3 Canonical Basis and Norms

1.1.3.1 Canonical Basis

Let $e_1, e_2, \dots, e_n$ denote the elements of the canonical basis of $\mathbb{R}^n$, where e_i is the vector in $\mathbb{R}^n$ given by:

$$(e_i)_j = \delta_{ij} = \begin{cases} 1 & \text{if } j = i, \\ 0 & \text{if } j \neq i, \end{cases} \quad \text{for all } i, j = 1, 2, \dots, n,$$

where δ_{ij} is the Kronecker delta symbol.

—

1.1.3.2 Dot Product

Definition 1.1.3 (Dot Product [18, Section 2.3]). *For any $x, y \in \mathbb{R}^n$, the dot product $\langle x, y\rangle$ is defined as:*

$$\langle x, y\rangle = \sum_{i=1}^{n} x_i y_i.$$

Two vectors $x, y \in \mathbb{R}^n$ are orthogonal (denoted $x \perp y$) if $\langle x, y\rangle = 0$.

—

1.1.3.3 Euclidean Norm

Definition 1.1.4 (Euclidean Norm [18, Section 2.3]). *For any $x \in \mathbb{R}^n$, the Euclidean norm of x, denoted $\|x\|$, is defined as:*

$$\|x\| = \sqrt{\langle x, x\rangle} = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

Properties of a norm (and thus of the Euclidean norm):

(i) $\|\alpha x\| = |\alpha|\|x\|$ for all $\alpha \in \mathbb{R}$ and $x \in \mathbb{R}^n$.

(ii) $\|x + y\| \leq \|x\| + \|y\|$ for all $x, y \in \mathbb{R}^n$.

(iii) $\|0\| = 0$ and $\|x\| > 0$ if $x \neq 0$.

—

Definition 1.1.5 (Open Ball [18, Section 2.3]). *For every $x \in \mathbb{R}^n$ and $r > 0$, the open ball centered at x with radius r is defined as:*

$$B(x; r) = \{y \in \mathbb{R}^n : \|y - x\| < r\}.$$

—

Definition 1.1.6 (Convergence). *If $(x^{(k)})$ is a sequence in $\mathbb{R}^n$ and x is an element of $\mathbb{R}^n$, we say that $x^{(k)}$ converges to x (denoted $x^{(k)} \to x$) if $\|x^{(k)} - x\| \to 0$ as $k \to \infty$. Equivalently, $x^{(k)} \to x$ if and only if $x_i^{(k)} \to x_i$ in $\mathbb{R}$ for each component i.*

—

Definition 1.1.7 (Interior, Open and Closed Sets [18, Section 2.3]). *Let $U \subset \mathbb{R}^n$.*

1. *The **interior** of U is the set of all $x \in U$ such that there exists $r > 0$ with $B(x; r) \subset U$.*
2. *The set U is **open** if for every $x \in U$, there exists $r > 0$ with $B(x; r) \subset U$.*
3. *The set U is **closed** if for every sequence $\{x^{(k)}\} \subset U$ such that $x^{(k)} \to x \in \mathbb{R}^n$, we have $x \in U$.*

—

Definition 1.1.8 (Segment). *If $a, b \in \mathbb{R}^n$, we denote by $[a, b]$ the subset of $\mathbb{R}^n$ given by:*

$$[a, b] = \{a + t(b - a) : t \in [0, 1]\}.$$

The set $[a, b]$ is called the segment connecting a to b.

—

Definition 1.1.9 (Cauchy-Schwarz Inequality [18, Section 2.3]). *For all $x, y \in \mathbb{R}^n$:*

$$|\langle x, y\rangle| \leq \|x\| \cdot \|y\|.$$

—

Definition 1.1.10 (Directional Derivative). *Let $f : \mathbb{R}^n \to \mathbb{R}$ and $x_0 \in \mathbb{R}^n$ where $f(x_0)$ is defined. The directional derivative of f at x_0 in the direction $d \in \mathbb{R}^n$ is defined by:*

$$f'_d(x_0) = \lim_{t \to 0^+} \frac{f(x_0 + td) - f(x_0)}{t},$$

if it exists. This derivative gives the rate of change of f at x_0 in the direction d.

—

Definition 1.1.11 (Fréchet Differentiability [2, Section 9.1]). *A function f is said to be **Fréchet differentiable** at $x_0 \in \mathbb{R}^n$ if there exists a continuous linear map $L(x_0) : \mathbb{R}^n \to \mathbb{R}$ such that:*

$$\lim_{d \to 0} \frac{f(x_0 + d) - f(x_0) - L(x_0) \cdot d}{\|d\|} = 0.$$

*The map $L(x_0)$ is called the **derivative** of f at x_0.*

Example 1.1.2 (Example 1.1.1)**.** *Consider the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by:*

$$f(x_1, x_2) = x_1 - x_2^2.$$

For any $d \in \mathbb{R}^2$, we have:

$$\begin{aligned}\lim_{t\to 0^+} \frac{f(x+td)-f(x)}{t} &= \lim_{t\to 0^+} \frac{f(x_1+td_1, x_2+td_2)-f(x_1,x_2)}{t} \\ &= (1, -2x_2)\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = f'(x)\cdot d.\end{aligned}$$

—

Example 1.1.3 (Example 1.1.2)**.** *Consider the function $f : \mathbb{R}^2 \setminus \{(0,0)\} \to \mathbb{R}$ defined by:*

$$f(x_1, x_2) = \begin{cases} x_1^2 x_2/(x_1^2 + x_2^2), & (x_1, x_2) \neq (0,0), \\ 0, & (x_1, x_2) = (0,0). \end{cases}$$

Obviously, the function f is continuous at $(0,0)$. Indeed, for $x_1 = r\cos\theta$, $x_2 = r\sin\theta$, with $r > 0$ and $\theta \in (0, 2\pi)$, we have:

$$\lim_{(x_1,x_2)\to(0,0)} f(x_1,x_2) = \lim_{r\to 0^+} \frac{r^3\cos^2\theta\sin\theta}{r^2} = \lim_{r\to 0^+} r\cos^2\theta\sin\theta = 0 = f(0,0).$$

Furthermore, f admits partial derivatives at $(0,0)$ since:

$$\frac{\partial f}{\partial x_1}(0,0) = \lim_{d_1\to 0} \frac{f(d_1,0)-f(0,0)}{d_1} = 0,$$

and

$$\frac{\partial f}{\partial x_2}(0,0) = \lim_{d_2\to 0} \frac{f(0,d_2)-f(0,0)}{d_2} = 0.$$

However, f is not Fréchet differentiable (and thus not differentiable) at $(0,0)$ because:

$$\lim_{(d_1,d_2)\to(0,0)} \frac{f(d_1,d_2) - f(0,0) - \left(\frac{\partial f}{\partial x_1}(0,0), \frac{\partial f}{\partial x_2}(0,0)\right)\begin{pmatrix} d_1 \\ d_2 \end{pmatrix}}{\|(d_1,d_2)\|} = \lim_{r\to 0} \cos^2\theta\sin\theta$$

does not exist.

Definition 1.1.12 (Gâteaux Differentiability)**.** *A function f is said to be **Gâteaux differentiable** (G-differentiable) at $x_0 \in \mathbb{R}^n$ if it admits a directional derivative at x_0 in every direction $d \in \mathbb{R}^n$.*

—

Definition 1.1.13 (Fréchet Differentiability). *A function f is said to be **Fréchet differentiable** (F-differentiable) at $x_0 \in \mathbb{R}^n$ if there exists a continuous linear map $L(x_0) : \mathbb{R}^n \to \mathbb{R}$ such that:*

$$\lim_{d \to 0} \frac{f(x_0 + d) - f(x_0) - L(x_0) \cdot d}{\|d\|} = 0.$$

*The map $L(x_0)$ is called the **derivative** of f at x_0.*
In other words, f is F-differentiable at $x_0 \in \mathbb{R}^n$ if there exists a continuous linear map $L(x_0) : \mathbb{R}^n \to \mathbb{R}$ such that:

$$f(x_0 + d) = f(x_0) + L(x_0) \cdot d + \|d\| \cdot \omega(d),$$

where $\lim_{d \to 0} \omega(d) = 0$.

—

Remark 1.1.1.

1. *A function f can be G-differentiable at $x_0 \in \mathbb{R}^n$ without being continuous at that point.*
2. *If f is F-differentiable at $x_0 \in \mathbb{R}^n$, then it is continuous at that point.*
3. *The notion of F-differentiability is stronger than that of G-differentiability.*
4. *If f is F-differentiable at $x_0 \in \mathbb{R}^n$ with derivative $L(x_0)$, then f is G-differentiable at x_0 and $L(x_0) = f'(x_0)$. The converse is false.*
5. *If f is F-differentiable, then f is differentiable.*
6. *A differentiable function at a point admits partial derivatives at that point. The converse is generally false.*
7. *A function differentiable on an open set $V \subset \mathbb{R}^n$ whose partial derivatives are all continuous on V is said to be of class C^1 on V.*

—

1.1.4 Gradient and Hessian Matrix

Definition 1.1.14 (Gradient [18, Section 2.1]). *The gradient of $f : \mathbb{R}^n \to \mathbb{R}$ at point x is the vector of partial derivatives:*

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

Example 1.1.4 (Example 1.2.1). *Consider the function $f(x_1, x_2, x_3) = e^{x_1} + x_1^2 x_3 - x_1 x_2 x_3$. Therefore, the gradient of f is given by:*

$$\nabla f(x_1, x_2, x_3) = \begin{pmatrix} e^{x_1} + 2x_1x_3 - x_2x_3 \\ -x_1x_3 \\ x_1^2 - x_1x_2 \end{pmatrix}.$$

Definition 1.1.15 (Hessian Matrix [2, Chapter 2])**.** *If f is twice continuously differentiable, the Hessian matrix is the matrix of second derivatives:*

$$\nabla^2 f(x) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]_{i,j=1}^{n}.$$

Example 1.1.5. *Let $f(x, y) = x^2 + xy + y^2$. Then:*

$$\nabla f(x, y) = \begin{pmatrix} 2x + y \\ x + 2y \end{pmatrix}, \quad \nabla^2 f(x, y) = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}.$$

Example 1.1.6 (Example 1.2.2)**.** *Consider the function $f(x_1, x_2, x_3) = e^{x_1} + x_2^2x_3 - x_1x_2x_3$. The Hessian of f is given by:*

$$H(x) = \begin{pmatrix} e^{x_1} & -x_3 & -x_2 \\ -x_3 & 2x_3 & -x_1 \\ -x_2 & -x_1 & 0 \end{pmatrix}.$$

—

Example 1.1.7 (Example 1.2.3)**.** *Define*

$$f(x_1, x_2, x_3) = e^{x_1} + x_2^2x_3 - x_1x_2x_3.$$

The Hessian of f is given by:

$$H(x) = \begin{pmatrix} e^{x_1} & -x_3 & -x_2 \\ -x_3 & 2x_3 & -x_1 \\ -x_2 & -x_1 & 0 \end{pmatrix}.$$

— —

In the following proposition, we provide the relationship between the gradient and Hessian matrix.

Proposition 1.1.1 ([18, Appendix A.4.3, pp. 656–657])**.**

1. *The i-th row of $\nabla^2 f(x)$ is the Jacobian of the i-th component of ∇f.*
2. *We have*
$$\nabla^2 f(x)h = \nabla_h^T \nabla f(x), \quad \forall x \in \mathbb{R}^n, \forall h \in \mathbb{R}^n.$$

Proof.

1. This is obvious.

2. We have:

$$\frac{\partial}{\partial x_i}\langle \nabla f(x), h\rangle = \frac{\partial}{\partial x_i}\left(\sum_{j=1}^{n} \frac{\partial f}{\partial x_j}(x)h_j\right) = \sum_{j=1}^{n} \frac{\partial^2 f}{\partial x_i \partial x_j}(x)h_j = (\nabla^2 f(x)h)_i.$$

□

Proposition 1.1.2 (Relation between ∇ and ∇^2 [18, Appendix A.4.2–A.4.3, pp. 655–657])**.**

1. *The i-th row of $\nabla^2 f(x)$ is the Jacobian of the i-th component of ∇f.*
2. *We have:*
$$\nabla^2 f(x)h = \nabla\langle \nabla f(x), h\rangle, \quad \forall x \in K, \forall h \in \mathbb{R}^n.$$

—

Proof.

1. This is obvious.
2. We have:

$$\frac{\partial}{\partial x_i}\langle \nabla f(x), h\rangle = \frac{\partial}{\partial x_i}\left(\sum_{j=1}^{n} \frac{\partial f}{\partial x_j}(x)h_j\right) = \sum_{j=1}^{n} \frac{\partial^2 f}{\partial x_i \partial x_j}(x)h_j = (\nabla^2 f(x)h)_i.$$

□

—

Example 1.1.8 (Example 1.3.7)**.** *If $f : \mathbb{R}^n \to \mathbb{R}$ is a constant function, then $\nabla f = \nabla^2 f = 0$.*

Let $f : \mathbb{R}^n \to \mathbb{R}$ be defined by

$$f(x) = \langle a, x\rangle, \quad \forall x \in \mathbb{R}^n,$$

where $a \in \mathbb{R}^n$ is a given vector (i.e. f is a linear function). Then we easily compute:

$$\frac{\partial f}{\partial x_k} = a_k, \quad \text{so}$$

$$\nabla f = a$$

(the gradient is constant).

Thus,

$$\nabla^2 f = 0.$$

—

Definition 1.1.16. *We say that x^* is a **stationary point** of f if $\nabla f(x^*) = 0$.*

—

1.1.4.1 Taylor Expansion

Theorem 1.1.1 (Taylor's Theorem [2, Section 9.1])**.** *If f is twice continuously differentiable, then:*

$$f(x+d) \approx f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x) d.$$

In particular, if one uses the Hessian at x one obtains the second-order approximation

$$f(x+d) = f(x) + \nabla f(x)^\top d + \tfrac{1}{2}\, d^\top \nabla^2 f(x)\, d + o(\|d\|^2) \quad (\|d\| \to 0). \tag{1.1}$$

Proof. The proof proceeds by reducing the multivariate statement to the classical one-variable Taylor theorem applied to the function obtained by restricting f to the line through x in direction d.

Define the one-variable function

$$g : [0,1] \to \mathbb{R}, \qquad g(t) := f(x+td).$$

Because $f \in C^2$ on a neighborhood of the segment $\{x + td : t \in [0,1]\}$, the function g is C^2 on $[0,1]$. By the chain rule we have for every $t \in [0,1]$:

$$g'(t) = \nabla f(x+td)^\top d, \tag{1.2}$$

$$g''(t) = d^\top \nabla^2 f(x+td)\, d. \tag{1.3}$$

Apply the one-variable Taylor theorem with Lagrange remainder to g about $t = 0$ evaluated at $t = 1$. There exists $\xi \in (0,1)$ such that

$$g(1) = g(0) + g'(0) \cdot 1 + \tfrac{1}{2} g''(\xi) \cdot 1^2.$$

Substituting $g(0) = f(x)$ and using (1.6)–(1.7) yields

$$f(x+d) = f(x) + \nabla f(x)^\top d + \tfrac{1}{2}\, d^\top \nabla^2 f(x+\xi d)\, d,$$

which is exactly (**??**) with $\theta = \xi$. This proves the theorem.

To obtain the approximation (1.1), note that continuity of $\nabla^2 f$ implies $\nabla^2 f(x + \theta d) = \nabla^2 f(x) + o(1)$ as $\|d\| \to 0$, and therefore the remainder term in (**??**) equals $\frac{1}{2} d^\top \nabla^2 f(x) d + o(\|d\|^2)$. ☐

Remark 1.1.2 (Integral form of the remainder)**.** *A convenient and often-used alternative is the integral form of the remainder, obtained by integrating $g''(t)$ twice:*

$$g(1) = g(0) + g'(0) + \int_0^1 (1-t)\, g''(t)\, dt.$$

Replacing $g''(t) = d^\top \nabla^2 f(x+td)\, d$ gives the identity

$$f(x+d) = f(x) + \nabla f(x)^\top d + \int_0^1 (1-t)\, d^\top \nabla^2 f(x+td)\, d\, dt. \tag{1.4}$$

Subtracting $\frac{1}{2} d^\top \nabla^2 f(x) d$ from both sides yields an explicit expression for the error of the quadratic approximation:

$$R(d) := f(x+d) - \Big(f(x) + \nabla f(x)^\top d + \tfrac{1}{2} d^\top \nabla^2 f(x) d \Big) = \int_0^1 (1-t)\, d^\top \big(\nabla^2 f(x+td) - \nabla^2 f(x) \big) d\, dt.$$

Corollary 1.1.1 (Remainder bound under Lipschitz Hessian)**.** *If, in addition, the Hessian is Lipschitz continuous on the segment* $\{x+td : t \in [0,1]\}$ *with Lipschitz constant* $L > 0$*, i.e.*

$$\|\nabla^2 f(y) - \nabla^2 f(z)\| \le L\|y-z\| \quad \textit{for } y, z \textit{ on the segment},$$

then the remainder satisfies the cubic bound

$$|R(d)| \le \frac{L}{6}\,\|d\|^3, \tag{1.5}$$

so the quadratic model approximates f *to order* $\mathcal{O}(\|d\|^3)$*.*

Sketch of proof of the bound. From the integral expression for $R(d)$ and the inequality $|d^\top A d| \le \|A\|\,\|d\|^2$ (operator norm), we obtain

$$|R(d)| \le \int_0^1 (1-t)\,\big\|\nabla^2 f(x+td) - \nabla^2 f(x)\big\|\,\|d\|^2\,dt.$$

By the Lipschitz condition $\|\nabla^2 f(x+td) - \nabla^2 f(x)\| \le Lt\|d\|$, hence

$$|R(d)| \le \int_0^1 (1-t)\,(Lt\|d\|)\,\|d\|^2\,dt = L\|d\|^3 \int_0^1 t(1-t)\,dt = \frac{L}{6}\,\|d\|^3,$$

which proves (1.5). □

Remark 1.1.3 (Exactness for quadratic functions)**.** *If* f *is a quadratic function whose Hessian is constant (i.e.* $\nabla^2 f$ *does not depend on the point), then the Lagrange remainder vanishes and* (**??**) *is exact for all* d*:*

$$f(x+d) = f(x) + \nabla f(x)^\top d + \tfrac{1}{2} d^\top \nabla^2 f\, d.$$

Suggested use in a handout: include Theorem **??**, the compact proof above, and the integral remainder (1.8). Add Corollary 1.1.1 when you discuss error estimates used in optimization (trust-region / Newton methods).

Example 1.1.9. *Using* $f(x,y) = x^2 + xy + y^2$ *at point* $(0,0)$ *with* $d = (1,1)$*:*

$$f(0,0) = 0, \quad \nabla f(0,0) = (0,0)^T, \quad \nabla^2 f(0,0) = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}.$$

Thus,

$$f(1,1) \approx 0 + 0 + \frac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix}^T \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{2}(2+1+1+2) = \frac{1}{2} \times 6 = 3.$$

1.1.4.2 Different forms of Taylor's formulas

The Taylor formula is an important tool in convex analysis. We recall it here in the general case.

Let $\Omega \subset \mathbb{R}^n$ be open, $f : \Omega \to \mathbb{R}$, $a \in \Omega$, and $h \in \mathbb{R}^n$ such that $[a, a+h] \subset \Omega$. Then:
—

1. If $f \in C^1(\Omega)$, then:

(a) **First-order Taylor formula with integral remainder**:

$$f(a+h) = f(a) + \int_0^1 \langle \nabla f(a+th), h \rangle \, dt.$$

(b) **First-order Taylor-Maclaurin formula**:

$$f(a+h) = f(a) + \langle \nabla f(a+th), h \rangle.$$

(c) **First-order Taylor-Young formula**:

$$f(a+h) = f(a) + \langle \nabla f(a+th), h \rangle + o(\|h\|).$$

—

2. If $f \in C^2(\Omega)$, then:

(a) **Second-order Taylor formula with integral remainder**:

$$f(a+h) = f(a) + \langle \nabla f(a), h \rangle + \int_0^1 (1-t) \langle \nabla^2 f(a+th)h, h \rangle \, dt.$$

(b) **Second-order Taylor-Maclaurin formula**:

$$f(a+h) = f(a) + \langle \nabla f(a), h \rangle + \frac{1}{2} \langle \nabla^2 f(a+\theta h)h, h \rangle, \quad \text{with } 0 < \theta < 1.$$

(c) **Second-order Taylor-Young formula**:

$$f(a+h) = f(a) + \langle \nabla f(a), h \rangle + \frac{1}{2} \langle \nabla^2 f(a)h, h \rangle + o(\|h\|^2).$$

Theorem 1.1.2 (Second-order Taylor's Theorem)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice differentiable function on an open set $V \subset \mathbb{R}^n$. Then:*

1. *For any $d \in \mathbb{R}^n$ such that $x + d \in V$, we have:*

$$f(x+d) = f(x) + d^T \nabla f(x) + \frac{1}{2} d^T \nabla^2 f(x) d + o(\|d\|^2).$$

2. *For any $d \in \mathbb{R}^n$ such that $x + d \in V$, there exists $t \in [0,1]$ such that:*

$$f(x+d) = f(x) + d^T \nabla f(x) + \frac{1}{2} d^T \nabla^2 f(x+td) d.$$

Proof. We prove both items by reducing to a one-variable Taylor expansion along the line segment from x to $x + d$.

Define the one-variable function

$$g : [0,1] \to \mathbb{R}, \qquad g(t) := f(x+td).$$

Because $f \in C^2$ on an open set containing the segment $\{x + td : t \in [0,1]\}$, the function g is of class C^2 on $[0,1]$. By the chain rule we obtain for every $t \in [0,1]$:

$$g'(t) = \nabla f(x+td)^\top d, \tag{1.6}$$
$$g''(t) = d^\top \nabla^2 f(x+td)\, d. \tag{1.7}$$

Integral form (useful intermediate identity). Integrate g'' twice using the fundamental theorem of calculus. First,

$$g'(t) - g'(0) = \int_0^t g''(s)\, ds,$$

and integrating this identity from $t = 0$ to $t = 1$ with weight $(1-t)$ (or directly performing the standard two-fold integration) yields the well-known integral remainder formula

$$g(1) = g(0) + g'(0) + \int_0^1 (1-t)\, g''(t)\, dt. \tag{1.8}$$

Substituting (1.6)–(1.7) into (1.8) gives

$$f(x+d) = f(x) + \nabla f(x)^\top d + \int_0^1 (1-t)\, d^\top \nabla^2 f(x+td)\, d\, dt. \tag{1.9}$$

Equation (1.9) is exact and will be the basis for proving (1) and (2).

Proof of (2) (Lagrange form). Because $g \in C^2([0,1])$, the one-variable Taylor theorem with Lagrange remainder guarantees existence of some $\xi \in (0,1)$ such that

$$g(1) = g(0) + g'(0) + \tfrac{1}{2} g''(\xi).$$

Using (1.6)–(1.7) we obtain

$$f(x+d) = f(x) + \nabla f(x)^\top d + \tfrac{1}{2}\, d^\top \nabla^2 f(x+\xi d)\, d.$$

Setting $t = \xi$ yields exactly the claimed Lagrange form of the remainder. This proves (2).

Proof of (1) (the $o(\|d\|^2)$ statement). We start from the integral identity (1.9) and split the integrand into the value at x plus the deviation:

$$\begin{aligned} f(x+d) &= f(x) + \nabla f(x)^\top d + \int_0^1 (1-t)\, d^\top \big(\nabla^2 f(x+td)\big) d\, dt \\ &= f(x) + \nabla f(x)^\top d + \tfrac{1}{2}\, d^\top \nabla^2 f(x) d + \underbrace{\int_0^1 (1-t)\, d^\top \big(\nabla^2 f(x+td) - \nabla^2 f(x)\big) d\, dt}_{=:R(d)}. \end{aligned}$$

Thus the remainder beyond the quadratic term is

$$R(d) = \int_0^1 (1-t)\, d^\top \big(\nabla^2 f(x+td) - \nabla^2 f(x)\big) d\, dt.$$

Now use the continuity of $\nabla^2 f$ at x (since $f \in C^2$). For any $\varepsilon > 0$ there exists $\delta > 0$ such that whenever $\|d\| < \delta$ and $t \in [0,1]$ we have

$$\|\nabla^2 f(x+td) - \nabla^2 f(x)\| \le \varepsilon,$$

where $\|\cdot\|$ denotes the operator norm (or any matrix norm equivalent to it). Using the quadratic form bound $|d^\top A d| \le \|A\| \, \|d\|^2$, we obtain for $\|d\| < \delta$:

$$|R(d)| \le \int_0^1 (1-t)\, \|\nabla^2 f(x+td) - \nabla^2 f(x)\| \, \|d\|^2 \, dt \le \varepsilon \|d\|^2 \int_0^1 (1-t)\, dt = \frac{\varepsilon}{2} \, \|d\|^2.$$

Since $\varepsilon > 0$ was arbitrary, this shows

$$\lim_{\|d\| \to 0} \frac{|R(d)|}{\|d\|^2} = 0,$$

i.e. $R(d) = o(\|d\|^2)$. Substituting back gives the expansion

$$f(x+d) = f(x) + d^\top \nabla f(x) + \frac{1}{2} d^\top \nabla^2 f(x) d + o(\|d\|^2),$$

which proves (1). $\square$

Remark 1.1.4.

- *If the Hessian $\nabla^2 f$ is* Lipschitz *on the segment $\{x + td : t \in [0,1]\}$ with constant L, one can strengthen the preceding estimate to the cubic bound*

$$|R(d)| \le \frac{L}{6} \, \|d\|^3,$$

showing the quadratic model approximates f to order $\mathcal{O}(\|d\|^3)$.

- *If f is a quadratic polynomial (Hessian constant), the remainder vanishes identically and the formula is exact for all d.*

—

1.1.5 Convexity

1.1.5.1 Convex Sets

Definition 1.1.17 (Convex Set [18, Section 2.1])**.** *A set $C \subseteq \mathbb{R}^n$ is convex if for any $x, y \in C$ and $\theta \in [0,1]$:*

$$\theta x + (1-\theta) y \in C.$$

Example 1.1.10. *The set $C = \{x \in \mathbb{R}^n : Ax \le b\}$ is convex since it is an intersection of halfspaces.*

—

1.1.5.2 Convex Hull

Definition 1.1.18 (Convex Hull [18, Section 2.3])**.** *The convex hull of a set $S \subseteq \mathbb{R}^n$ is the smallest convex set containing S, defined as:*

$$conv(S) = \left\{ \sum_{i=1}^{k} \theta_i x_i : x_i \in S, \theta_i \ge 0, \sum_{i=1}^{k} \theta_i = 1, k \in \mathbb{N} \right\}.$$

Definition 1.1.19. *Given two points $a, b \in \mathbb{R}^n$, the segment $[a, b]$ is defined as:*

$$[a, b] = \{x \in \mathbb{R}^n : x = (1-t)a + tb \text{ for some } t \in [0, 1]\}.$$

A set $K \subset \mathbb{R}^n$ is called ***convex*** *if, for all points $a, b \in K$, the segment $[a, b]$ is contained in K.*

—

Properties of Convex Sets.

1. The definition of a convex set can be interpreted by saying that the segment connecting x and y must be contained in C.

2. Let $x_1, x_2, \dots, x_k \in \mathbb{R}^n$ and $t_j \geq 0$ such that $\sum_{j=1}^{k} t_j = 1$. Any expression of the form:

$$\sum_{j=1}^{k} t_j x_j$$

   is called a **convex combination** of the points x_j or a **barycenter**.

3. If C_1 and C_2 are two convex sets in $\mathbb{R}^n$, then $K = C_1 \cap C_2$ is convex, and

$$K = \{x : x = x_1 + x_2, x_1 \in C_1, x_2 \in C_2\}$$

   is convex.

—

Example 1.1.11 (Examples 1.4.1)**.**

1. *The intersection of convex sets is convex. This follows directly from the definition.*

2. *An affine subspace K of $\mathbb{R}^n$ is convex. Recall that $K \subset \mathbb{R}^n$ is an affine subspace if there exists a vector subspace $V \subset \mathbb{R}^n$ and a vector $v \in \mathbb{R}^n$ such that $x \in K$ if and only if $x - v \in V$. To show that K is convex, let $x, y \in K$, and set $z = (1-t)x + ty$ with $t \in [0, 1]$. Then $x - v, y - v \in V$, and thus $z - v = (1-t)(x-v) + t(y-v) \in V$, hence $z \in K$.*

—

1.1.6 Convex Functions

1.1.6.1 Convexity and Differentiability

Definition 1.1.20 (Convex Function [18, Section 3.1])**.** *A function $f : C \to \mathbb{R}$ defined on a convex set C is convex if:*

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y), \quad \forall x, y \in C, \theta \in [0, 1].$$

Proposition 1.1.3 (First Order Condition)**.** *If f is differentiable, then f is convex if and only if:*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x), \quad \forall x, y \in C.$$

Proof of equivalence. ($\Rightarrow$) Assume f is convex. Fix $x, y \in C$ and $t \in (0, 1]$. By convexity,

$$f\big((1-t)x + ty\big) \leq (1-t)f(x) + tf(y).$$

Rearranging gives

$$\frac{f(x + t(y-x)) - f(x)}{t} \leq f(y) - f(x).$$

Letting $t \to 0^+$, the left-hand side converges to the directional derivative of f at x in the direction $y - x$, that is $\nabla f(x)^\top (y - x)$. Hence,

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x).$$

($\Leftarrow$) Conversely, assume that

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x), \qquad \forall x, y \in C.$$

Fix $x, y \in C$ and $\theta \in [0, 1]$, and set $z = (1-\theta)x + \theta y$. Applying the inequality with base point z yields

$$f(x) \geq f(z) + \nabla f(z)^\top (x - z), \qquad f(y) \geq f(z) + \nabla f(z)^\top (y - z).$$

Multiplying the first inequality by $(1 - \theta)$ and the second by θ, then adding, we obtain

$$(1-\theta)f(x) + \theta f(y) \geq f(z) + \nabla f(z)^\top \big((1-\theta)(x - z) + \theta(y - z)\big).$$

But $(1-\theta)(x - z) + \theta(y - z) = 0$ by construction of z. Hence

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

which is exactly the convexity inequality. Therefore f is convex. $\square$

Proposition 1.1.4 (Second Order Condition)**.** *If f is twice differentiable, then f is convex if and only if its Hessian is positive semidefinite everywhere on C.*

Proof. We prove both directions.

$\Rightarrow$ Assume f is convex on C and $f \in C^2(C)$. Fix $x \in C$ and an arbitrary vector $v \in \mathbb{R}^n$ such that the segment $\{x + tv : t \in (-\varepsilon, \varepsilon)\}$ is contained in C for some $\varepsilon > 0$ (this holds at least for small ε). Define the one-variable function

$$\varphi(t) := f(x + tv), \qquad t \in (-\varepsilon, \varepsilon).$$

Since f is convex on C and $t \mapsto x + tv$ is affine, φ is a convex function of t. Because $f \in C^2(C)$, φ is twice continuously differentiable and by the chain rule

$$\varphi'(t) = \nabla f(x + tv)^\top v, \qquad \varphi''(t) = v^\top \nabla^2 f(x + tv)\, v.$$

Convexity of φ implies $\varphi''(t) \geq 0$ for all t in the interval; evaluating at $t = 0$ gives

$$v^\top \nabla^2 f(x)\, v = \varphi''(0) \geq 0.$$

Since v was arbitrary, $\nabla^2 f(x)$ is positive semidefinite. As x was arbitrary in C, the Hessian is positive semidefinite everywhere on C.

$\Leftarrow$ Conversely, assume $\nabla^2 f(x) \succeq 0$ for every $x \in C$. To prove convexity of f, fix arbitrary $x, y \in C$ and consider the function

$$g(t) := f\big(x + t(y - x)\big), \qquad t \in [0, 1].$$

Then $g \in C^2([0, 1])$ and by the chain rule

$$g''(t) = (y - x)^\top \nabla^2 f\big(x + t(y - x)\big)(y - x) \geq 0$$

for all $t \in [0, 1]$, because each Hessian along the segment is positive semidefinite. Hence g is a convex function on the interval $[0, 1]$.

Convexity of g implies, for every $\theta \in [0, 1]$,

$$g(\theta) \leq (1 - \theta)g(0) + \theta g(1).$$

Translating back to f (noting $g(0) = f(x)$, $g(1) = f(y)$ and $g(\theta) = f((1 - \theta)x + \theta y))$ yields

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

for all $x, y \in C$ and all $\theta \in [0, 1]$. This is exactly the definition of convexity of f on C; therefore f is convex.

Integral (Taylor) representation. For completeness, we derive the integral identity that explicitly shows how the Hessian appears in the difference $f(y) - f(x) - \nabla f(x)^\top (y - x)$. With $v := y - x$ and $g(t) = f(x + tv)$ as above, integrate g'' twice on $[0, 1]$:

$$g'(t) - g'(0) = \int_0^t g''(s)\, ds,$$

and then

$$g(1) - g(0) - g'(0) = \int_0^1 \big(g'(t) - g'(0)\big)\, dt = \int_0^1 \left(\int_0^t g''(s)\, ds\right) dt.$$

By Fubini (or direct computation of the triangular integral),

$$\int_0^1 \left(\int_0^t g''(s)\, ds\right) dt = \int_0^1 (1 - s)g''(s)\, ds.$$

Therefore

$$f(y) = f(x) + \nabla f(x)^\top (y - x) + \int_0^1 (1 - s)\, (y - x)^\top \nabla^2 f(x + s(y - x))(y - x)\, ds.$$

If $\nabla^2 f(\cdot) \succeq 0$ on C, the integrand is nonnegative for all $s \in [0, 1]$, hence the integral is ≥ 0, which immediately yields the first-order inequality

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x),$$

and therefore convexity (by the first-order characterization). This integral identity also quantifies how the curvature (encoded by the Hessian) contributes to the deviation from linearity.

$\square$

—

1.1.6.2 Monotone Functions

Definition 1.1.21 (Monotone Function [18, Section 3.1])**.** *A function $f : \mathbb{R} \to \mathbb{R}$ is monotone increasing if $x \leq y$ implies $f(x) \leq f(y)$.*

Example 1.1.12. *The exponential function $f(x) = e^x$ is monotone increasing.*

Definition 1.1.22. *A real-valued function $J : K \subset V \to \mathbb{R}$ defined on a convex set K of a vector space V is said to be* ***convex*** *on K if for all points $u, v \in K$ and for all real numbers t with $0 \leq t \leq 1$, the following inequality holds:*

$$J(tu + (1-t)v) \leq tJ(u) + (1-t)J(v).$$

It is said to be ***strictly convex*** *on K if:*

$$u, v \in K, u \neq v, t \in (0,1) \quad \Longrightarrow \quad J(tu + (1-t)v) < tJ(u) + (1-t)J(v).$$

—

Definition 1.1.23. *A function f is said to be* ***strongly convex*** *with constant $\alpha > 0$ if:*

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2) - \frac{\alpha}{2}t(1-t)\|x_1 - x_2\|^2.$$

It can be easily shown that a strongly convex function is strictly convex.

—

Remark 1.1.5. *Strong convexity $\Longrightarrow$ strict convexity $\Longrightarrow$ convexity.*

Definition 1.1.24. *A function $f : K \subset V \to \mathbb{R}$ defined on a convex subset K of a vector space V is said to be* ***(strictly) concave*** *if the function $(-f)$ is* ***(strictly) convex****.*

—

Example 1.1.13 (Examples 1.4.2)**.**

1. *An affine function is convex on $\mathbb{R}^n$, but not strictly convex. An affine function is given by $f(x) = \langle a, x \rangle + b$ for some $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$.*

2. *A norm is convex on $\mathbb{R}^n$ but not strictly convex. This follows from the triangle inequality: for all $x, y \in \mathbb{R}^n$ and $t \in (0,1)$,*

$$\|(1-t)x + ty\| \leq (1-t)\|x\| + t\|y\|.$$

The inequality is not strict because if $y = \lambda x$ for some $\lambda \geq 0$, equality holds.

—

Operations on Convex Functions. Let $K \subset \mathbb{R}^n$ be convex.

1. The sum of two convex functions on K is convex.
2. If $f : K \to \mathbb{R}$ is convex and $\lambda \geq 0$, then λf is convex on K.
3. The maximum of two convex functions on K is convex.
4. If $f : K \to \mathbb{R}$ is convex and $L : \mathbb{R}^m \to \mathbb{R}^n$ is linear, then $f \circ L$ is convex on

$$V = \{x \in \mathbb{R}^m : L(x) \in K\}.$$

—

Definition 1.1.25 (Epigraph of a Convex Function)**.** *Let K be a non-empty subset of $\mathbb{R}^n$ and $f : K \to \mathbb{R}$. The* ***epigraph*** *of f, denoted epi(f), is defined as:*

$$epi(f) = \{(x, y) \in \mathbb{R}^{n+1} : x \in K, y \geq f(x)\}.$$

Equivalently,

$$epi(f) = \{(x, y) \in K \times \mathbb{R} : y \geq f(x)\}.$$

It is the set of points above the graph of f.

—

Theorem 1.1.3. *Let K be a non-empty convex set in $\mathbb{R}^n$ and $f : K \to \mathbb{R}$. Then*

$$f \text{ is convex} \iff epi(f) \text{ is convex.}$$

Proof. We prove the two implications separately.

$\Rightarrow$ Assume f is convex on K. We show that $\operatorname{epi}(f)$ is convex. Fix two arbitrary points $(x_1, t_1), (x_2, t_2) \in \operatorname{epi}(f)$ and let $\theta \in [0, 1]$. By membership in the epigraph we have $t_1 \geq f(x_1)$ and $t_2 \geq f(x_2)$. Because K is convex and $x_1, x_2 \in K$, the point

$$x_\theta := (1 - \theta)x_1 + \theta x_2$$

belongs to K. Consider the convex combination in $\mathbb{R}^n \times \mathbb{R}$

$$(x_\theta, t_\theta) := (1 - \theta)(x_1, t_1) + \theta(x_2, t_2) = \big((1 - \theta)x_1 + \theta x_2,\ (1 - \theta)t_1 + \theta t_2\big).$$

Using convexity of f and the inequalities $t_i \geq f(x_i)$, we obtain

$$(1 - \theta)t_1 + \theta t_2 \geq (1 - \theta)f(x_1) + \theta f(x_2) \geq f\big((1 - \theta)x_1 + \theta x_2\big) = f(x_\theta).$$

Thus $t_\theta \geq f(x_\theta)$, so $(x_\theta, t_\theta) \in \operatorname{epi}(f)$. Since the two arbitrary points and arbitrary $\theta \in [0, 1]$ were arbitrary, $\operatorname{epi}(f)$ is convex.

$\Leftarrow$ Conversely, assume $\operatorname{epi}(f)$ is convex. We prove f is convex on K. Fix arbitrary $x_1, x_2 \in K$ and $\theta \in [0, 1]$. By definition of f we have

$$(x_1, f(x_1)),\ (x_2, f(x_2)) \in \operatorname{epi}(f).$$

Convexity of epi(f) implies that the point

$$(x_\theta, t_\theta) := (1-\theta)(x_1, f(x_1)) + \theta(x_2, f(x_2))$$

belongs to epi(f). As before $x_\theta = (1-\theta)x_1 + \theta x_2$ and $t_\theta = (1-\theta)f(x_1) + \theta f(x_2)$. Membership $(x_\theta, t_\theta) \in \text{epi}(f)$ means $t_\theta \geq f(x_\theta)$, i.e.

$$(1-\theta)f(x_1) + \theta f(x_2) \geq f\big((1-\theta)x_1 + \theta x_2\big).$$

Rewriting yields the usual convexity inequality

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

Because $x_1, x_2 \in K$ and $\theta \in [0,1]$ were arbitrary, f is convex on K.

This completes the proof of the equivalence. □

—

1.1.7 Convexity and Continuity

Proposition 1.1.5 (Jensen's Inequality [18, Section 3.1.5, p. 72]). *Suppose f is convex on K, and let $p_1, \ldots, p_m \geq 0$ with $p_1 + \cdots + p_m = 1$. Then, for all $x_1, \ldots, x_m \in K$,*

$$f(p_1 x_1 + \cdots + p_m x_m) \leq p_1 f(x_1) + \cdots + p_m f(x_m).$$

Proof. Immediate for $m = 1$; equivalent to the definition of convexity for $m = 2$; the result follows by induction for $m > 2$. □

—

Theorem 1.1.4 (Continuity of convex functions [11, Theorem 10.1]). *If K is open and f is convex on K, then f is continuous on K.*

Proof. We prove it in the case $K = \mathbb{R}^2$ and at the origin for simplicity, with $\|\cdot\|_\infty$ norm. We proceed in two steps:

Step 1. Show there exists $C \in \mathbb{R}$ such that for all $r > 0$ in a neighborhood of 0, and for all x with $\|x\| \leq r$,

$$f(x) - f(0) \leq Cr.$$

Assuming x is in the first quadrant (other cases are similar), we have:

$$x = p_1 0 + p_2 e_1 + p_3 e_2,$$

where $p_1 = 1 - x_1 - x_2$, $p_2 = x_1$, $p_3 = x_2$, and $0 = (0,0)$, $e_1 = (1,0)$, $e_2 = (0,1)$. For sufficiently small r, $p_1, p_2, p_3 > 0$ and $p_1 + p_2 + p_3 = 1$.

By Jensen's inequality,

$$f(x) \leq p_1 f(0) + p_2 f(e_1) + p_3 f(e_2),$$

thus

$$f(x) - f(0) \leq |p_1 - 1||f(0)| + p_2|f(e_1)| + p_3|f(e_2)|.$$

□

The assertion then follows from the fact that

$$|p_1 - 1| \leq 2r, \quad p_2 \leq r, \quad p_3 \leq r.$$

—

Step 2. We now show that there exists $C' \in \mathbb{R}$ such that for all $r > 0$ in a neighborhood of 0, and for all x with $\|x\| \leq r$, we have:

$$f(0) - f(x) \leq C'r.$$

To see this, we write:

$$0 = p_1 x + p_2(-e_1) + p_3(-e_2),$$

with

$$p_1, p_2, p_3 > 0, \quad p_1 + p_2 + p_3 = 1.$$

This decomposition is made possible by taking:

$$p_1 = \frac{1}{1 + x_1 + x_2}, \quad p_2 = \frac{x_1}{1 + x_1 + x_2}, \quad p_3 = \frac{x_2}{1 + x_1 + x_2},$$

assuming again that x is in the first quadrant. By convexity of f, we find:

$$f(0) - f(x) \leq |p_1 - 1||f(x)| + p_2|f(-e_1)| + p_3|f(-e_2)|.$$

Taking r sufficiently small, and observing again that:

$$|p_1 - 1| \leq 2r, \quad p_2 \leq r, \quad p_3 \leq r,$$

and using Step 1 to bound $|f(x)|$, we obtain the desired result.

Thus, continuity at the origin follows directly from the two assertions above.

—

Remark 1.1.6. *The proposition is false if K is not open. For example, let $K = [0, 1] \subset \mathbb{R}$ and consider $f : K \to \mathbb{R}$ defined by*

$$f(x) = \begin{cases} 0, & \textit{if } x \in (0, 1), \\ 1, & \textit{if } x \in \{0, 1\}. \end{cases}$$

—

Definition 1.1.26. *Let f be a function defined from K to $\mathbb{R} \cup \{+\infty\}$. The* ***domain*** *of f, denoted $dom(f)$, is defined as:*

$$dom(f) = \{x \in K : f(x) \neq +\infty\}.$$

Definition 1.1.27. *A function f defined on K is said to be* ***coercive*** *if:*

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

—

Definition 1.1.28. *A function f defined from K to $\mathbb{R} \cup \{+\infty\}$ is said to be **lower semicontinuous** (LSC) if for all $x \in K$:*

$$\liminf_{y \to x} f(y) \geq f(x).$$

—

Remark 1.1.7. 1. *Lower semicontinuity is stable under addition.*

2. *Every continuous function is LSC.*

3. *The indicator function of a closed convex set is LSC (i.e., the function that is zero on the convex set and $+\infty$ outside).*

4. *If f is LSC, then for every $\alpha \in \mathbb{R}$, the sets*

$$\{x \in K : f(x) \leq \alpha\} \quad \text{and} \quad \{(x, \alpha) \in K \times \mathbb{R} : f(x) \leq \alpha\}$$

are closed.

—

Definition 1.1.29. *A function from K to $\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$ is said to be **proper** if it is not identically equal to $+\infty$ and does not take the value $-\infty$.*

—

1.2 Convexity and First-Order Differentiability

Below is a first result that allows one to recognize the convexity of a function using its first derivatives.

Proposition 1.2.1 (Characterization of convexity via derivative [11, Theorem 24.1])**.** *Let f be a differentiable function on an interval $I \subset \mathbb{R}$. Then:*

$$f \text{ is convex} \iff f' \text{ is non-decreasing on } I.$$

Proof. Suppose f is convex and show that f' is non-decreasing. Let $0 \leq \lambda_1 \leq \lambda_2$. Then, by convexity of f:

$$f(x + \lambda_1 d) = f\Big((1 - \frac{\lambda_1}{\lambda_2})x + \frac{\lambda_1}{\lambda_2}(x + \lambda_2 d)\Big) \leq (1 - \frac{\lambda_1}{\lambda_2})f(x) + \frac{\lambda_1}{\lambda_2} f(x + \lambda_2 d).$$

□

Therefore:

$$\frac{f(x + \lambda_1 d) - f(x)}{\lambda_1} \leq \frac{f(x + \lambda_2 d) - f(x)}{\lambda_2}.$$

Hence the result.

—

2. Suppose that f' is increasing, and let us show that f is convex. Let x, y such that $y > x$. Define the function:

$$\varphi(t) = tf(x) + (1-t)f(y) - f(tx + (1-t)y).$$

We want to show that $\varphi(t) \geq 0$ for all $t \in [0, 1]$. Note that $\varphi(0) = \varphi(1) = 0$. Moreover:

$$\varphi'(t) = f(x) - f(y) + (y-x)f'((x-y)t + y).$$

Since $y - x > 0$, we have:

$$\varphi'(t) = k + af'(y - at),$$

with $a > 0$. This function is decreasing since f' is increasing.

If $\varphi'(1) > 0$, then for all $t \in [0, 1]$,

$$\varphi'(t) > 0,$$

and

$$\varphi(1) = \varphi(0) + \int_0^1 \varphi'(t)dt > \varphi(0) = \varphi(1),$$

which is absurd. Hence $\varphi'(1) \leq 0$. Similarly, one shows that $\varphi'(0) \geq 0$. Therefore, for all $t \in [0, 1]$, $\varphi(t) \geq 0$, and thus f is convex.

—

Corollary 1.2.1. *Let f be twice differentiable on an interval $I \subset \mathbb{R}$. Then:*

$$f \text{ is convex} \iff f'' \geq 0 \text{ on } I,$$

$$f \text{ is concave} \iff f'' \leq 0 \text{ on } I.$$

Example 1.2.1.

1. *$f(x) = x^2$ is convex since $f''(x) = 2 > 0$, $\forall x \in \mathbb{R}$.*
2. *$f(x) = e^x$ is convex since $f''(x) = e^x > 0$, $\forall x \in \mathbb{R}$.*
3. *$f(x) = \ln x$ is concave since $f''(x) = -\frac{1}{x^2} < 0$, $\forall x \in (0, +\infty)$.*

—

Theorem 1.2.1 (First-order characterization of convexity [**?**, Theorem 25.1])**.** *Let E be a normed space, K an open convex subset of E, and $f : K \to \mathbb{R}$ a differentiable function. Then the following are equivalent:*

1. *f is convex on K;*
2. *$\forall x, y \in K, f(y) \geq f(x) + f'(x) \cdot (y - x)$;*
3. *$\forall x, y \in K, [f'(y) - f'(x)] \cdot (y - x) \geq 0$.*

Proof. We prove the cycle of implications (1) ⇒ (2) ⇒ (3) ⇒ (2) and then (2) ⇒ (1).

(1) ⇒ **(2)**. Assume f is convex on K. Fix $x, y \in K$ and define the one-variable function

$$\varphi : [0, 1] \to \mathbb{R}, \qquad \varphi(t) := f(x + t(y - x)).$$

Since f is convex on K and the map $t \mapsto x + t(y-x)$ is affine, φ is a convex function on $[0,1]$. Because f is differentiable on K, φ is differentiable on $[0,1]$ with derivative

$$\varphi'(t) = f'\big(x + t(y-x)\big)\big(y-x\big).$$

A standard property of convex differentiable functions on an interval is that their graph lies above every tangent line; in particular, $\varphi(1) \geq \varphi(0) + \varphi'(0)$. Substituting the definitions gives

$$f(x + y - x) = f(y) \geq f(x) + f'(x)\big(y-x\big),$$

which is exactly (2).

(2) $\Rightarrow$ **(3)**. Assume (2) holds for all $x, y \in K$. Apply (2) with the pair (x,y) and then with the pair (y,x):

$$\begin{aligned} f(y) &\geq f(x) + f'(x)\big(y-x\big), \\ f(x) &\geq f(y) + f'(y)\big(x-y\big). \end{aligned}$$

Add these two inequalities to obtain

$$0 \geq f'(x)\big(y-x\big) + f'(y)\big(x-y\big) = -\big(f'(y) - f'(x)\big)\big(y-x\big).$$

Rearranging yields $(f'(y) - f'(x))\big(y-x\big) \geq 0$, which is (3).

(3) $\Rightarrow$ **(2)**. Assume (3) holds. Fix $x, y \in K$ and consider again the one-variable function $\varphi(t) := f(x + t(y-x))$ for $t \in [0,1]$. As before, $\varphi'(t) = f'(x + t(y-x))\big(y-x\big)$. The hypothesis (3) implies that for every $t \in [0,1]$,

$$\varphi'(t) - \varphi'(0) = \big(f'(x + t(y-x)) - f'(x)\big)\big(y-x\big) \geq 0,$$

so $\varphi'(t) \geq \varphi'(0)$ for all $t \in [0,1]$. Integrating this inequality over $t \in [0,1]$ yields

$$\varphi(1) - \varphi(0) = \int_0^1 \varphi'(t)\,dt \geq \int_0^1 \varphi'(0)\,dt = \varphi'(0).$$

Unwinding the definitions gives

$$f(y) - f(x) \geq f'(x)\big(y-x\big),$$

which is exactly (2). Thus (3)$\Rightarrow$(2).

(2) $\Rightarrow$ **(1)**. Assume (2) holds. For each $x \in K$ define the affine function $L_x : K \to \mathbb{R}$ by

$$L_x(y) := f(x) + f'(x)\big(y-x\big).$$

Hypothesis (2) states precisely that $L_x(y) \leq f(y)$ for all $y \in K$, i.e. every tangent affine map L_x is a global underestimator of f. Moreover $L_x(x) = f(x)$ for each x. Hence, for any fixed $y \in K$,

$$\sup_{x \in K} L_x(y) = f(y),$$

because the supremum is bounded above by $f(y)$ (by (2)) and is attained at $x = y$. But the pointwise supremum of an arbitrary family of affine (hence convex) functions is a convex function. Therefore f is convex on K. This proves (2)$\Rightarrow$(1).

Combining the implications above gives the equivalence of (1), (2), and (3). $\square$

Remark 1.2.1. • *The pairing $f'(x)(y-x)$ denotes the action of the bounded linear functional $f'(x)$ on the vector $y-x$. In finite dimensions $f'(x)$ is the usual gradient and the pairing reduces to the dot product $\nabla f(x)^\top (y-x)$.*

- *The representation $f = \sup_{x\in K} L_x$ used in (2)⇒(1) is the standard fact that a differentiable function that lies above all its tangents is the supremum of those tangents; this is the cornerstone of the first-order characterization of convexity.*

—

Theorem 1.2.2 (First-order characterization of convexity [?, Theorem 25.1])**.** *Under the same assumptions, the following are equivalent:*

1. *f is strictly convex on K;*
2. *$\forall x, y \in K, x \neq y : f(y) > f(x) + f'(x)\cdot(y-x)$;*
3. *$\forall x, y \in K, x \neq y : [f'(y) - f'(x)]\cdot(y-x) > 0$.*

Proof. We prove the cycle of implications (1) ⇒ (2) ⇒ (3) ⇒ (2) and then (2) ⇒ (1).

(1) ⇒ **(2)**. Assume f is strictly convex on K. Fix $x, y \in K$ with $x \neq y$, and define the one-variable function

$$\varphi : [0,1] \to \mathbb{R}, \qquad \varphi(t) := f\big(x + t(y-x)\big).$$

Because f is strictly convex on K, the restriction φ is strictly convex on $[0,1]$. Differentiability of f implies φ is differentiable, with

$$\varphi'(t) = f'\big(x + t(y-x)\big)\big(y-x\big).$$

A standard property of strictly convex differentiable functions on an interval is that the graph lies *strictly* above any tangent line except at the tangency point; in particular, $\varphi(1) > \varphi(0) + \varphi'(0)$. Unwinding definitions yields

$$f(y) > f(x) + f'(x)\big(y-x\big),$$

which is (2).

(2) ⇒ **(3)**. Assume (2) holds for all distinct $x, y \in K$. Apply (2) first to the pair (x, y) and then to (y, x):

$$f(y) > f(x) + f'(x)\big(y-x\big),$$
$$f(x) > f(y) + f'(y)\big(x-y\big).$$

Adding these strict inequalities gives

$$0 > f'(x)\big(y-x\big) + f'(y)\big(x-y\big) = -\big(f'(y) - f'(x)\big)\big(y-x\big).$$

Rearranging yields $(f'(y) - f'(x))(y-x) > 0$, which is (3).

(3) ⇒ **(2)**. Assume (3) holds. Fix $x, y \in K$ with $x \neq y$ and define again $\varphi(t) = f(x + t(y-x))$ for $t \in [0,1]$. Then $\varphi'(t) = f'(x + t(y-x))(y-x)$. From (3) we have for every $t \in (0,1]$

$$\big(f'(x + t(y-x)) - f'(x)\big)\big(y-x\big) > 0,$$

hence $\varphi'(t) - \varphi'(0) > 0$, so $\varphi'(t) > \varphi'(0)$ for all $t \in (0, 1]$. Integrating φ' over $[0, 1]$ yields

$$\varphi(1) - \varphi(0) = \int_0^1 \varphi'(t)\, dt > \int_0^1 \varphi'(0)\, dt = \varphi'(0).$$

Unwinding definitions gives

$$f(y) - f(x) > f'(x)\big(y - x\big),$$

which is (2). Thus (3)⇒(2).

(2) ⇒ **(1)**. Assume (2) holds. We show f is strictly convex. Fix distinct points $x, y \in K$ and take any $\lambda \in (0, 1)$. Put $z := (1 - \lambda)x + \lambda y \in K$. We must prove

$$f(z) < (1 - \lambda)f(x) + \lambda f(y).$$

Apply (2) with base point x and argument z. Since $z \neq x$,

$$f(z) > f(x) + f'(x)(z - x).$$

Rearrange this inequality to express $f'(x)(z - x)$:

$$f'(x)(z - x) < f(z) - f(x).$$

Now multiply the strict inequality from (2) for the pair (x, y) by $\lambda > 0$:

$$\lambda\big(f(y) - f(x) - f'(x)(y - x)\big) > 0.$$

But $z - x = \lambda(y - x)$, so $f'(x)(z - x) = \lambda f'(x)(y - x)$. Substitute this into the previous displayed inequality to obtain

$$\lambda f(y) - \lambda f(x) - f'(x)(z - x) > 0.$$

Rearrange:

$$f'(x)(z - x) < \lambda\big(f(y) - f(x)\big).$$

Combine this with $f(z) > f(x) + f'(x)(z - x)$:

$$f(z) < f(x) + \lambda\big(f(y) - f(x)\big) = (1 - \lambda)f(x) + \lambda f(y).$$

This is the strict convexity inequality at z. Since x, y and λ were arbitrary (with $x \neq y$, $\lambda \in (0, 1)$), f is strictly convex on K. Thus (2)⇒(1).

Collecting the implications above we obtain the equivalence of (1), (2), and (3). □

Remark 1.2.2.
- *In finite dimensions the Fréchet derivative $f'(x)$ identifies with the gradient $\nabla f(x)$ and the pairing $f'(x)(y-x)$ becomes the dot product $\nabla f(x)^\top(y-x)$.*
- *The arguments above mirror the nonstrict case but replace weak inequalities by strict inequalities; care is taken when integrating or adding inequalities to preserve strictness.*

—

Theorem 1.2.3. *Let E be a Euclidean space, K an open convex subset of E, and $f : K \to \mathbb{R}$ a differentiable function. Then the following are equivalent:*

1. *f is strongly convex on K;*
2. *$\exists \alpha > 0, \forall x, y \in K : f(y) \geq f(x) + f'(x) \cdot (y - x) + \frac{\alpha}{2}\|y - x\|^2$;*
3. *$\exists \alpha > 0, \forall x, y \in K : [f'(y) - f'(x)] \cdot (y - x) \geq \alpha\|y - x\|^2$.*

Proof. We prove the cycle of implications (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (2) and then (2) $\Rightarrow$ (1).

(1) $\Rightarrow$ (2). By definition, f is strongly convex with modulus $\alpha > 0$ iff the function $g := f - \frac{\alpha}{2}\| \cdot \|^2$ is convex on K. Since g is convex and differentiable, the first-order condition for convex functions gives for every $x, y \in K$,

$$g(y) \geq g(x) + g'(x) \cdot (y - x).$$

Now $g'(x) = f'(x) - \alpha x$ (the derivative of $\frac{1}{2}\alpha\|x\|^2$ is αx), so substituting g and g' yields

$$f(y) - \tfrac{\alpha}{2}\|y\|^2 \geq f(x) - \tfrac{\alpha}{2}\|x\|^2 + \big(f'(x) - \alpha x\big) \cdot (y - x).$$

Rearrange terms (expand the quadratic terms or move α-terms to the right) to obtain the desired inequality

$$f(y) \geq f(x) + f'(x) \cdot (y - x) + \tfrac{\alpha}{2}\|y - x\|^2.$$

Thus (1)$\Rightarrow$(2).

(2) $\Rightarrow$ (3). Assume (2) holds for some $\alpha > 0$. Apply (2) with the ordered pair (x, y) and then with (y, x):

$$\begin{aligned} f(y) &\geq f(x) + f'(x) \cdot (y - x) + \tfrac{\alpha}{2}\|y - x\|^2, \\ f(x) &\geq f(y) + f'(y) \cdot (x - y) + \tfrac{\alpha}{2}\|x - y\|^2. \end{aligned}$$

Add the two inequalities (note $\|x - y\| = \|y - x\|$) to get

$$0 \geq f'(x) \cdot (y - x) + f'(y) \cdot (x - y) + \alpha\|y - x\|^2 = -\big(f'(y) - f'(x)\big) \cdot (y - x) + \alpha\|y - x\|^2.$$

Rearranging yields

$$\big(f'(y) - f'(x)\big) \cdot (y - x) \geq \alpha\|y - x\|^2,$$

which is (3).

(3) $\Rightarrow$ (2). Assume (3) holds for some $\alpha > 0$. Fix $x, y \in K$ and set $d := y - x$. Because f is differentiable, the map $t \mapsto f'(x + td)$ is defined for $t \in [0, 1]$. For each $t \in [0, 1]$ apply (3) to the pair $(x, x + td)$ to obtain

$$\big(f'(x + td) - f'(x)\big) \cdot \big((x + td) - x\big) \geq \alpha\|(x + td) - x\|^2,$$

i.e.

$$\big(f'(x + td) - f'(x)\big) \cdot (td) \geq \alpha t^2\|d\|^2.$$

For $t > 0$ divide by t to get

$$\big(f'(x + td) - f'(x)\big) \cdot d \geq \alpha t\|d\|^2.$$

(When $t = 0$ the left-hand side is 0 by continuity of f', and the inequality then holds in the limiting sense.) Now integrate this inequality with respect to t over $[0, 1]$:

$$\int_0^1 \big(f'(x + td) - f'(x)\big) \cdot d\, dt \geq \int_0^1 \alpha t \|d\|^2\, dt = \tfrac{\alpha}{2} \|d\|^2.$$

But

$$\int_0^1 \big(f'(x+td)-f'(x)\big)\cdot d\, dt = \int_0^1 f'(x+td)\cdot d\, dt - \int_0^1 f'(x)\cdot d\, dt = \int_0^1 f'(x+td)\cdot d\, dt - f'(x)\cdot d.$$

Using the fundamental identity (line integral representation of the increment)

$$f(y) - f(x) = \int_0^1 f'(x + td) \cdot d\, dt,$$

we obtain

$$f(y) - f(x) - f'(x) \cdot d \geq \tfrac{\alpha}{2} \|d\|^2,$$

which is exactly

$$f(y) \geq f(x) + f'(x) \cdot (y - x) + \tfrac{\alpha}{2} \|y - x\|^2.$$

This proves (3)⇒(2).

(2) ⇒ **(1)**. Finally, assume (2) holds for some $\alpha > 0$. Define $g := f - \frac{\alpha}{2} \| \cdot \|^2$. Then for every $x, y \in K$ the inequality (2) is equivalent to

$$g(y) \geq g(x) + g'(x) \cdot (y - x),$$

because the gradient of $\frac{\alpha}{2} \| \cdot \|^2$ is αx, so $g'(x) = f'(x) - \alpha x$. The displayed inequality is the first-order condition for convexity of the differentiable function g, hence g is convex on K. Therefore $f = g + \frac{\alpha}{2} \| \cdot \|^2$ is strongly convex with modulus α, which is (1).

Combining the implications above proves the equivalence of (1), (2), and (3). □

Remark 1.2.3.

- *In finite dimensions the derivative $f'(x)$ is identified with the gradient $\nabla f(x)$ and the pairing $f'(x) \cdot (y - x)$ becomes $\nabla f(x)^\top (y - x)$.*
- *If f is twice continuously differentiable, an equivalent and often used pointwise formulation is $\nabla^2 f(x) \succeq \alpha I$ for all $x \in K$.*
- *In the implication (3)⇒(2) we used the line integral identity $f(y) - f(x) = \int_0^1 f'(x + td) \cdot d\, dt$ together with a simple integration of the linear inequalities obtained from (3).*

—

1.3 Convexity and Second-Order Differentiability

Definition 1.3.1. *Let $H(x^*)$ be the Hessian matrix of f at point x^*. Then:*

1. *$H(x^*)$ is said to be* ***positive semi-definite*** *(PSD), denoted $H(x^*) \geq 0$, if*

$$\forall x \in \mathbb{R}^n, \quad x^T H(x^*) x \geq 0.$$

Thus, all eigenvalues of $H(x^*)$ are non-negative.

2. *$H(x^*)$ is said to be* ***positive definite*** *(PD), denoted $H(x^*) > 0$, if*

$$\forall x \in \mathbb{R}^n \setminus \{0\}, \quad x^T H(x^*) x > 0.$$

Thus, all eigenvalues of $H(x^*)$ are strictly positive.

Theorem 1.3.1 (Second-order characterization of convexity [18, Section 3.1.3])**.** *Let $K \subset \mathbb{R}^n$ be an open, non-empty convex set and $f : K \to \mathbb{R}$ be a real-valued function that is twice differentiable in K. Then:*

1. *f is convex in K if and only if the Hessian matrix is positive semi-definite (p.s.d) at every point in K.*

2. *f is strictly convex in K if and only if the Hessian matrix is positive definite (p.d) at every point in K.*

Proof. 1. Assume that f is convex in K. Then:

$$\forall x^* \in K, \quad f(x^* + \lambda x) \geq f(x^*) + \lambda(\nabla f(x^*))^T x, \quad \forall x \in K. \quad (1.1)$$

Since f is twice differentiable in K, by the second-order Taylor-Maclaurin expansion:

$$f(x^* + \lambda x) = f(x^*) + \lambda(\nabla f(x^*))^T x + \frac{1}{2}\lambda^2 x^T H(x^*) x + \lambda^2 \|x\|^2 \alpha(x^*, \lambda x). \quad (1.2)$$

Combining (1.1) and (1.2) gives:

$$\frac{1}{2}\lambda^2 x^T H(x^*) x + \lambda^2 \|x\|^2 \alpha(x^*, \lambda x) \geq 0.$$

Dividing by λ^2 and letting $\lambda \to 0$ yields:

$$x^T H(x^*) x \geq 0.$$

2. Conversely, suppose $H(x^*)$ is p.s.d. for all x^*. Then:

$$f(x) = f(x^*) + (\nabla f(x^*))^T (x - x^*) + \frac{1}{2}(x - x^*)^T H(\zeta)(x - x^*), \quad \zeta = tx + (1-t)x^*,\ t \in (0, 1).$$

So:

$$f(x) - [f(x^*) + (\nabla f(x^*))^T (x - x^*)] = \frac{1}{2}(x - x^*)^T H(\zeta)(x - x^*) \geq 0,$$

hence f is convex.

□

Example 1.3.1. *Let $f : \mathbb{R}^3 \to \mathbb{R}$ be defined by $f(x, y, z) = (x-2)^2 + (y-3)^2 + z^2$.*

- *The gradient is:*
$$\nabla f(x, y, z) = \begin{pmatrix} 2x-4 \\ 2y-6 \\ 2z \end{pmatrix}.$$
- *The Hessian is:*
$$H(x, y, z) = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} = 2I_3 \geq 0.$$
Therefore, f is convex.

Remark 1.3.1. *If f is convex, then any local minimum is also global. Moreover, if f is strictly convex, then any local minimum is not only global but unique.*

—

1.3.1 Exercises

1. Compute the directional derivative of $f(x, y) = xy^2$ at point $(1, 2)$ in the direction $d = (2, 1)$.
2. Prove that the set $S = \{x \in \mathbb{R}^n : \|x\|_2 \leq 1\}$ is convex.
3. Show that $f(x) = x^4$ is convex on $\mathbb{R}$.

—

1.3.2 Solutions

1. $\nabla f(x, y) = (y^2, 2xy)$. At $(1, 2)$, $\nabla f = (4, 4)$. Thus:
$$D_f((1, 2); (2, 1)) = (4, 4) \cdot (2, 1) = 8 + 4 = 12.$$
2. For any x, y in the set and $\theta \in [0, 1]$:
$$\|\theta x + (1-\theta)y\|_2 \leq \theta\|x\|_2 + (1-\theta)\|y\|_2 \leq \theta + (1-\theta) = 1.$$
Thus, it is convex.
3. $f''(x) = 12x^2 \geq 0$ for all x, thus f is convex.

Sheet 01

Exercise 1

Let the following functions be:

1. $f_1(x) = \dfrac{3}{2}x_1^2 + 2x_2^2 + \dfrac{3}{2}x_3^2 + x_1x_2 + 2x_2x_3 - 3x_1 - x_3$
2. $f_2(x) = (x_1 - 1)^2 + 10(x_1 - x_2)^2$
3. $f_3(x) = 5x_1^2 + 5x_2^2 - x_1x_2 + 11x_1 + 11x_2 + 11$

(a) Calculate $\nabla f_i(x)$ and $\nabla^2 f_i(x)$ for $i = 1, 2, 3$.

(b) Among these functions, which are quadratic? Justify.

—

Exercise 2

Determine $Df(x)$ and $\nabla^2 f(x)$ of the quadratic function:

$$f(x) = \frac{1}{2}\langle x, Ax\rangle + \langle x, b\rangle + c,$$

where $A \in M_{n\times n}(\mathbb{R})$, $b \in \mathbb{R}^n$, and $c \in \mathbb{R}$.

—

Exercise 3

Give the second-order Taylor expansion of function f around point x_0 for:

(a) $f(x) = x_1e^{-x_2} + x_2 + 1$, $x_0 = (1, 0)^T$.

(b) $f(x) = x_1^4 - 2x_1^2x_2^2 + x_2^4$, $x_0 = (1, 1)^T$.

—

Exercise 4

Let $x(t) = (t^2, t, t)^T$, $t \in \mathbb{R}$, and $f(x) = x_1x_2^3 + x_1x_2 + x_3$, with $x = (x_1, x_2, x_3)^T \in \mathbb{R}^3$. Find $\frac{d}{dt}f(x(t))$ in terms of t.

—

Exercise 5

The purpose of this exercise is to derive Taylor's formula for a function $f : \mathbb{R}^n \to \mathbb{R}$. Let $f \in C^2$, $x, x_0 \in \mathbb{R}^n$, and define:

$$z(\alpha) = x_0 + \alpha\frac{(x - x_0)}{\|x - x_0\|}.$$

Define the function $\Phi(\alpha) = f(z(\alpha))$.

(a) Determine $\Phi'(\alpha)$ and $\Phi''(\alpha)$.

(b) By noting that $f(x) = \Phi(\|x - x_0\|)$, deduce:

$$f(x) = f(x_0) + Df(x_0)(x - x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0) + o(\|x - x_0\|^2).$$

—

Exercise 6

Let ϕ be a continuous function from $\mathbb{R}$ to $\mathbb{R}$, and define $f, g : \mathbb{R}^2 \to \mathbb{R}$ by:

$$f(x, y) = \int_0^{x+y} \phi(t)\, dt, \quad g(x, y) = \int_0^{xy} \phi(t)\, dt.$$

(a) Show that f and g are of class C^1 on $\mathbb{R}^2$.

(b) Let $Df(x, y)$ and $Dg(x, y)$ be the differentials of f and g at point (x, y), respectively. Calculate $Df(x, y)(h, k)$ and $Dg(x, y)(h, k)$ for all $(h, k) \in \mathbb{R}^2$.

—

Exercise 7

Let $f : \mathbb{R}^n \to \mathbb{R}$ and $x : \mathbb{R} \to \mathbb{R}^n$ be two C^2 functions. Define $g(t) = f(x(t))$.

(a) Calculate $g''(t)$ in the case where $x(t) = u + tv$, where $u, v \in \mathbb{R}^n$.

(b) Calculate $g''(t)$ for general $x(t)$.

—

Exercise 8

Which of the following sets are convex?

1. $S_1 = \{(x, y) \in \mathbb{R}^2 \mid 0 \leq x \leq 1, y = 0\}$
2. $S_2 = \{(x, y) \in \mathbb{R}^2 \mid x \geq 0, y \geq 0, x + y \leq 1\}$
3. $S_3 = \{x \in \mathbb{R}^n \mid A_1 x = b_1, A_2 x \leq b_2\}$ where A_1 and A_2 are matrices of size $m \times n$, and b_1 and b_2 are vectors in $\mathbb{R}^m$.
4. $S_4 = \{(x, y) \in \mathbb{R}^2 \mid y - x^2 \geq 0\}$
5. $S_5 = \{(x, y) \in \mathbb{R}^2 \mid xy \geq 1, x > 0\}$

—

Exercise 9

Verify whether the following functions are convex or not on $\mathbb{R}^2$:

1. $f(x, y) = x^2 - xy + 2y^2 - 2x + e^{x+y}$
2. $f(x, y) = (x - 2)^4 + (x - 2)^2 y^2 + (y + 1)^2$
3. $f(x, y) = -x^2 - 2xy - 2y^2$

—

Exercise 10
We consider the function f defined on $\mathbb{R}^2$ by

$$f(x, y) = x^4 + y^4 - 2(x - y)^2.$$

1. Show that there exist $(a, b) \in \mathbb{R}_+^2$ (and determine them) such that

   $$f(x, y) \geq a\|(x, y)\|^2 + b$$

   for all $(x, y) \in \mathbb{R}^2$, where $\| \cdot \|$ denotes the Euclidean norm on $\mathbb{R}^2$. Deduce that the problem

   $$\inf_{(x,y)\in\mathbb{R}^2} f(x, y)$$

   has at least one solution.

2. Is the function f convex on $\mathbb{R}^2$?

—

Exercise 11
We define the function $J : \mathbb{R}^2 \to \mathbb{R}$ by

$$J(x, y) = y^4 - 3xy^2 + x^2.$$

1. Determine the critical points of J.
2. Let $d = (d_1, d_2) \in \mathbb{R}^2$. Using the function $t \mapsto J(td_1, td_2)$, show that $(0, 0)$ is a local minimum along any line passing through $(0, 0)$.
3. Is the point $(0, 0)$ a local minimum of the restriction of J to the parabola given by the equation $x = y^2$?
4. Compute the Hessian matrix of J. What is the nature of the critical point $(0, 0)$?

Solution (Sheet 01)

Solution 1.1

Let us consider the following functions:

1.
$$f_1(x) = \frac{3}{2}x_1^2 + 2x_2^2 + \frac{3}{2}x_3^2 + x_1x_2 + 2x_2x_3 - 3x_1 - x_3.$$

2.
$$f_2(x) = (x_1 - 1)^2 + 10(x_1 - x_2)^2.$$

3.
$$f_3(x) = 5x_1^2 + 5x_2^2 - x_1x_2 + 11x_1 + 11x_2 + 11.$$

(a) Compute $\nabla f_i(x)$ and $\nabla^2 f_i(x)$ for $i = 1, 2, 3$.

- For $f_1(x)$:

$$\nabla f_1(x) = \begin{pmatrix} 3x_1 + x_2 - 3 \\ 4x_2 + x_1 + 2x_3 \\ 3x_3 + 2x_2 - 1 \end{pmatrix}, \quad \nabla^2 f_1(x) = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 4 & 2 \\ 0 & 2 & 3 \end{pmatrix}.$$

- For $f_2(x)$:

$$\nabla f_2(x) = \begin{pmatrix} 22x_1 - 20x_2 - 2 \\ -20x_1 + 20x_2 \end{pmatrix}, \quad \nabla^2 f_2(x) = \begin{pmatrix} 22 & -20 \\ -20 & 20 \end{pmatrix}.$$

- For $f_3(x)$:

$$\nabla f_3(x) = \begin{pmatrix} 10x_1 - x_2 + 11 \\ -x_1 + 10x_2 + 11 \end{pmatrix}, \quad \nabla^2 f_3(x) = \begin{pmatrix} 10 & -1 \\ -1 & 10 \end{pmatrix}.$$

(b) Since $\nabla^2 f_i(x)$ is constant and symmetric, each f_i is quadratic for $i = 1, 2, 3$.

—

Solution 1.2

Clearly, the function f is of class C^2, so we can apply Lemma 1.1:

$$Df(x)h = \lim_{t \to 0} \frac{f(x + th) - f(x)}{t}.$$

We have:

$$f(x + th) = \frac{1}{2}\langle x + th, A(x + th)\rangle + \langle b, x + th\rangle + c.$$

Expanding,

$$= f(x) + t\langle Ax, h\rangle + t\langle b, h\rangle + \frac{t^2}{2}\langle Ah, h\rangle.$$

Hence,

$$Df(x)h = \langle Ax, h\rangle + \langle b, h\rangle = \langle Ax + b, h\rangle.$$

Therefore,

$$Df(x) = Ax + b.$$

—

Solution 1.3

Given $x(t) = (e^t, t^2, t)^T$ and $f(x) = x_1x_2x_3 + x_1x_2 + x_3$, compute $\frac{d}{dt}f(x(t))$.

Theorem. If $f : D \subset \mathbb{R}^n \to \mathbb{R}$ is differentiable, and $x : (a, b) \to D$ is differentiable, then

$$\frac{d}{dt}f(x(t)) = \nabla f(x(t))^T x'(t).$$

Using this,

$$\nabla f(x) = \begin{pmatrix} x_2x_3 + x_2 \\ x_1x_3 + x_1 \\ x_1x_2 + 1 \end{pmatrix}.$$

Thus,

$$x'(t) = \begin{pmatrix} e^t \\ 2t \\ 1 \end{pmatrix}.$$

Therefore,

$$\frac{d}{dt}f(x(t)) = (x_2x_3 + x_2)e^t + (x_1x_3 + x_1)2t + (x_1x_2 + 1).$$

—

Solution 1.4

Recall (Taylor expansion of order 2):

If $f \in C^2$, then near x_0,

$$f(x) = f(x_0) + Df(x_0)(x - x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0) + o(\|x - x_0\|^2).$$

(a)

For $f(x) = x_1e^{-x_2} + x_2 + 1$, at $x_0 = (1, 0)^T$,

Calculate gradients and Hessian and write the expansion accordingly (details skipped here).

—

Solution 1.5

Given $z(\alpha) = x_0 + \alpha\frac{x - x_0}{\|x - x_0\|}$ and $\Phi(\alpha) = f(z(\alpha))$.

(a) Using chain rule,

$$\Phi'(\alpha) = \frac{1}{\|x - x_0\|}(x - x_0)^T \nabla f(z(\alpha)),$$

$$\Phi''(\alpha) = \frac{1}{\|x - x_0\|^2}(x - x_0)^T \nabla^2 f(z(\alpha))(x - x_0).$$

(b) Hence,

$$f(x) = f(x_0) + Df(x_0)(x - x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0) + o(\|x - x_0\|^2).$$

—

Solution 1.6

Define $\phi : \mathbb{R} \to \mathbb{R}$ continuous, and

$$f(x,y) = \int_0^{x+y} \phi(t)dt, \quad g(x,y) = \int_0^{xy} \phi(t)dt.$$

1. Both f and g are C^1 as compositions of C^1 functions.
2. Their differentials are:

$$Df(x,y)(h,k) = \phi(x+y)(h+k),$$

$$Dg(x,y)(h,k) = \phi(xy)(yh+xk).$$

—

Exercise 1.7

1. Let $S_1 = \{(x,y) \in \mathbb{R}^2 \mid y - x \geq 0\}$. To prove S_1 is convex, let $X_1 = (x_1,y_1), X_2 = (x_2,y_2) \in S_1$ and $\lambda \in [0,1]$. Then:

$$\lambda X_1 + (1-\lambda)X_2 = (\lambda x_1 + (1-\lambda)x_2, \lambda y_1 + (1-\lambda)y_2).$$

We need:

$$\lambda y_1 + (1-\lambda)y_2 - [\lambda x_1 + (1-\lambda)x_2] \geq 0,$$

which simplifies to:

$$\lambda(y_1 - x_1) + (1-\lambda)(y_2 - x_2) \geq 0.$$

Since $y_i - x_i \geq 0$, S_1 is convex.

—

2. Let $S_2 = \{(x,y) \in \mathbb{R}^2 \mid x \geq 0, y \geq 0, x + y \leq 1\}$. For any $X_1, X_2 \in S_2$, $\lambda \in [0,1]$, define:

$$X = \lambda X_1 + (1-\lambda)X_2.$$

We check:

$$x \geq 0, \quad y \geq 0, \quad x + y \leq 1.$$

Indeed,

$$x + y = \lambda(x_1 + y_1) + (1-\lambda)(x_2 + y_2) \leq \lambda \cdot 1 + (1-\lambda) \cdot 1 = 1.$$

Hence S_2 is convex.

—

3. Let $S_3 = \{x \in \mathbb{R}^n \mid A_1 x = b_1, A_2 x \leq b_2\}$. For any $x, y \in S_3$ and $\lambda \in [0,1]$:

$$A_1(\lambda x + (1-\lambda)y) = \lambda A_1 x + (1-\lambda)A_1 y = b_1,$$

$$A_2(\lambda x + (1-\lambda)y) \leq \lambda A_2 x + (1-\lambda)A_2 y \leq b_2.$$

Thus, S_3 is convex.

—

4. Let $S_4 = \{(x,y) \in \mathbb{R}^2 \mid y - x^2 \geq 0\}$. For any $X, Y \in S_4$ and $\lambda \in [0,1]$:

$$\lambda y_1 + (1-\lambda)y_2 \geq \lambda x_1^2 + (1-\lambda)x_2^2.$$

By Jensen's inequality:

$$(\lambda x_1 + (1-\lambda)x_2)^2 \leq \lambda x_1^2 + (1-\lambda)x_2^2.$$

Hence:

$$\lambda y_1 + (1-\lambda)y_2 \geq (\lambda x_1 + (1-\lambda)x_2)^2,$$

so S_4 is convex.

—

5. Let $S_5 = \{(x,y) \in \mathbb{R}^2 \mid xy \geq 1, x > 0\}$. For $X, Y \in S_5$ and $\lambda \in [0,1]$:

$$(\lambda x_1 + (1-\lambda)x_2)(\lambda y_1 + (1-\lambda)y_2) \geq \lambda^2 x_1 y_1 + (1-\lambda)^2 x_2 y_2 + \lambda(1-\lambda)(x_1 y_2 + x_2 y_1).$$

However, this is not necessarily ≥ 1, so S_5 is not convex.

—

Exercise 1.8

1. For $f(x,y) = x^2 - xy + 2y^2 - 2x + e^{x+y}$:

$$\nabla f(x,y) = \begin{pmatrix} 2x - y - 2 + e^{x+y} \\ -x + 4y + e^{x+y} \end{pmatrix},$$

$$\nabla^2 f(x,y) = \begin{pmatrix} 2 + e^{x+y} & -1 + e^{x+y} \\ -1 + e^{x+y} & 4 + e^{x+y} \end{pmatrix}.$$

The eigenvalues are positive, so f is strictly convex.

—

2. For $f(x,y) = (x-2)^4 + (x-2)^2 y^2 + (y+1)^2$:

The Hessian is:

$$\nabla^2 f(x,y) = \begin{pmatrix} 12(x-2)^2 + 2y^2 & 4y(x-2) \\ 4y(x-2) & 2(x-2)^2 + 2 \end{pmatrix}.$$

The sign changes depending on x, y, so f is neither convex nor concave.

—

3. For $f(x,y) = -x^2 - 2xy - 2y^2$:

The Hessian is:

$$\nabla^2 f(x,y) = \begin{pmatrix} -2 & -2 \\ -2 & -4 \end{pmatrix}.$$

The eigenvalues are negative, so f is strictly concave.

—

Exercise 1.9

1. The function f is polynomial, thus of class $\mathcal{C}^\infty(\mathbb{R}^2)$. It is given by:

$$f(x,y) = x^4 + y^4 - 2(x-y)^2.$$

To prove that f is coercive, note that:

$$x^4 + y^4 \geq \frac{1}{2}(x^2 + y^2)^2 \geq 0.$$

Also,

$$-2(x-y)^2 \geq -4(x^2 + y^2).$$

Thus,

$$f(x, y) \geq \frac{1}{2}(x^2 + y^2)^2 - 4(x^2 + y^2).$$

Define $r^2 = x^2 + y^2$, then

$$f(x, y) \geq \frac{1}{2}r^4 - 4r^2.$$

As $r \to \infty$, $f(x, y) \to \infty$. Therefore, f is coercive and attains a minimum on $\mathbb{R}^2$.

—

2. To check convexity, compute the Hessian:

$$\nabla^2 f(x, y) = \begin{pmatrix} 12x^2 - 4 & 0 \\ 0 & 12y^2 - 4 \end{pmatrix}.$$

At $(0, 0)$, the eigenvalues are -4 and -4, so the Hessian is not positive semidefinite everywhere. Thus, f is not convex.

—

Exercise 1.10

Let $J(x, y) = y^4 - 3xy^2 + x^2$.

1. Compute $\nabla J(x, y)$:

$$\nabla J(x, y) = \begin{pmatrix} -3y^2 + 2x \\ 4y^3 - 6xy \end{pmatrix}.$$

Set $\nabla J = 0$, then:

$$-3y^2 + 2x = 0 \quad \Rightarrow \quad x = \frac{3}{2}y^2,$$

$$4y^3 - 6xy = 0.$$

Substitute x:

$$4y^3 - 6\left(\frac{3}{2}y^2\right)y = 4y^3 - 9y^3 = -5y^3 = 0.$$

Thus, $y = 0$, so $x = 0$. Therefore, the only critical point is $(0, 0)$.

—

2. Consider $t \mapsto J(td_1, td_2)$. Then:

$$J(td_1, td_2) = (td_2)^4 - 3(td_1)(td_2)^2 + (td_1)^2.$$

At $t = 0$, $J(0, 0) = 0$. The second derivative is positive along any line through the origin, so $(0, 0)$ is a local minimum along any such line.

—

3. Restricting to the parabola $x = y^2$:

$$J(y^2, y) = y^4 - 3(y^2)y^2 + (y^2)^2 = y^4 - 3y^4 + y^4 = -y^4.$$

Thus, $J(y^2, y) = -y^4 \leq 0$ with equality only at $y = 0$. Therefore, $(0, 0)$ is a local maximum along this curve, not a minimum.

—

4. Compute the Hessian:

$$\nabla^2 J(x,y) = \begin{pmatrix} 2 & -6y \\ -6y & 12y^2 - 6x \end{pmatrix}.$$

At $(0,0)$:

$$\nabla^2 J(0,0) = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}.$$

This matrix is positive semidefinite with rank 1, thus $(0,0)$ is a saddle point along some directions and minimum along others.

—

Exercise 1.11

We are asked to derive the second-order Taylor expansion of the function f around x_0.

—

(a) Let $f(x) = x_1 e^{-x_2} + x_2 + 1$, with $x_0 = (1,0)^T$.
First, compute the gradient:

$$\nabla f(x) = \begin{pmatrix} e^{-x_2} \\ -x_1 e^{-x_2} + 1 \end{pmatrix}.$$

At x_0:

$$\nabla f(x_0) = \begin{pmatrix} 1 \\ -1+1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Compute the Hessian:

$$\nabla^2 f(x) = \begin{pmatrix} 0 & -e^{-x_2} \\ -e^{-x_2} & x_1 e^{-x_2} \end{pmatrix}.$$

At x_0:

$$\nabla^2 f(x_0) = \begin{pmatrix} 0 & -1 \\ -1 & 1 \end{pmatrix}.$$

Thus, the second-order Taylor expansion is:

$$f(x) \approx f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0).$$

—

(b) Let $f(x) = x_1^4 - 2x_1^2 x_2^2 + x_2^4$, with $x_0 = (1,1)^T$.
Compute the gradient:

$$\nabla f(x) = \begin{pmatrix} 4x_1^3 - 4x_1 x_2^2 \\ -4x_1^2 x_2 + 4x_2^3 \end{pmatrix}.$$

At x_0:

$$\nabla f(x_0) = \begin{pmatrix} 4-4 \\ -4+4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Compute the Hessian:

$$\nabla^2 f(x) = \begin{pmatrix} 12x_1^2 - 4x_2^2 & -8x_1x_2 \\ -8x_1x_2 & -4x_1^2 + 12x_2^2 \end{pmatrix}.$$

At x_0:

$$\nabla^2 f(x_0) = \begin{pmatrix} 12 - 4 & -8 \\ -8 & -4 + 12 \end{pmatrix} = \begin{pmatrix} 8 & -8 \\ -8 & 8 \end{pmatrix}.$$

Therefore, the second-order Taylor expansion is:

$$f(x) \approx f(x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0).$$

—

Exercise 1.12
Let $x(t) = (e^t, 2t, 1)^T$, and $f(x) = x_1x_2x_3 + x_1x_2 + x_3$.
We need to compute:

$$\frac{d}{dt} f(x(t)).$$

First compute $\nabla f(x)$:

$$\nabla f(x) = \begin{pmatrix} x_2x_3 + x_2 \\ x_1x_3 + x_1 \\ x_1x_2 + 1 \end{pmatrix}.$$

Then compute $\frac{dx}{dt}$:

$$x'(t) = \begin{pmatrix} e^t \\ 2 \\ 0 \end{pmatrix}.$$

Thus,

$$\frac{d}{dt} f(x(t)) = \nabla f(x(t))^T x'(t).$$

Calculate at general t:

$$= [x_2x_3 + x_2, x_1x_3 + x_1, x_1x_2 + 1] \begin{pmatrix} e^t \\ 2 \\ 0 \end{pmatrix}.$$

Substitute $x_1 = e^t$, $x_2 = 2t$, $x_3 = 1$:
- $x_2x_3 + x_2 = 2t \cdot 1 + 2t = 4t$ - $x_1x_3 + x_1 = e^t \cdot 1 + e^t = 2e^t$ - $x_1x_2 + 1 = e^t \cdot 2t + 1 = 2te^t + 1$.
Thus,

$$\frac{d}{dt} f(x(t)) = 4te^t + 2e^t \cdot 2 + 0 = 4te^t + 4e^t.$$

—

Exercise 1.13
Let $f \in C^2$, $x, x_0 \in \mathbb{R}^n$, and define:

$$z(\alpha) = x_0 + \alpha \frac{(x - x_0)}{\|x - x_0\|}.$$

Define $\Phi(\alpha) = f(z(\alpha))$.

—

1. Compute $\Phi'(\alpha)$ and $\Phi''(\alpha)$.
We have:

$$\Phi'(\alpha) = Df(z(\alpha)) \cdot z'(\alpha).$$

But

$$z'(\alpha) = \frac{(x - x_0)}{\|x - x_0\|}.$$

Hence,

$$\Phi'(\alpha) = \nabla f(z(\alpha))^T \frac{(x - x_0)}{\|x - x_0\|}.$$

—

For $\Phi''(\alpha)$:

$$\Phi''(\alpha) = \frac{d}{d\alpha} \left[\nabla f(z(\alpha))^T \frac{(x - x_0)}{\|x - x_0\|} \right].$$

Since $\frac{(x-x_0)}{\|x-x_0\|}$ is constant,

$$\Phi''(\alpha) = \frac{(x - x_0)^T}{\|x - x_0\|^2} \nabla^2 f(z(\alpha))(x - x_0).$$

—

2. Since $f(x) = \Phi(\|x - x_0\|)$, we can write the Taylor expansion:

$$f(x) = \Phi(0) + \|x - x_0\|\Phi'(0) + \frac{1}{2}\|x - x_0\|^2\Phi''(0) + o(\|x - x_0\|^2).$$

Which implies:

$$f(x) = f(x_0) + Df(x_0)(x - x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0) + o(\|x - x_0\|^2).$$

—

This result rederives the second-order Taylor formula.
Exercise 1.14
Let $\phi : \mathbb{R} \to \mathbb{R}$ be continuous, and define:

$$f(x, y) = \int_0^{x+y} \phi(t)\, dt, \quad g(x, y) = \int_0^{xy} \phi(t)\, dt.$$

—

1. Show that f and g are of class C^1.
Since ϕ is continuous and the integration limit is differentiable, f and g are C^1 by differentiation under the integral sign.

—

2. Compute the differentials:
- For f:

$$Df(x,y)(h,k) = \phi(x+y)(h+k).$$

- For g:

$$Df(x,y)(h,k) = \phi(xy)(yh+xk).$$

—

Therefore:

$$Df(x,y)(h,k) = \phi(x+y)(h+k), \quad Dg(x,y)(h,k) = \phi(xy)(yh+xk).$$

Exercise 9

Let $f : \mathbb{R}^n \to \mathbb{R}$, and $x : \mathbb{R} \to \mathbb{R}^n$ be C^2 functions. Define:

$$g(t) = f(x(t)).$$

—

1. If $x(t) = u + tv$, where $u, v \in \mathbb{R}^n$, then:

$$g'(t) = \nabla f(x(t))^T v,$$

$$g''(t) = v^T \nabla^2 f(x(t)) v.$$

—

2. For general $x(t)$,

$$g'(t) = \nabla f(x(t))^T x'(t),$$

$$g''(t) = x'(t)^T \nabla^2 f(x(t)) x'(t) + \nabla f(x(t))^T x''(t).$$

—

These formulas result from applying the chain rule and product rule for derivatives.

Chapter 2

Unconstrained Minimization

Introduction

Unconstrained minimization refers to the optimization of an objective function without the presence of explicit constraints on the decision variables. In such problems, the goal is to find a point where the function attains its minimum value over its entire domain. This class of optimization problems forms the foundation of numerical optimization, with applications in data fitting, machine learning, economics, and engineering design. Analytical and numerical methods, such as gradient-based algorithms and Newton-type methods, are widely used to solve unconstrained minimization problems efficiently.

Let $f : K \to \mathbb{R}$. The **unconstrained minimization problem** is defined as:

$$\min_{x\in\mathbb{R}^n} f(x) \tag{2.1}$$

We aim to solve:

$$\min_{x\in K\subset\mathbb{R}^n} f(x) \quad \text{or} \quad \max_{x\in K\subset\mathbb{R}^n} f(x),$$

that is, we seek v, the optimal value, and x^* such that:

$$f(x^*) = v.$$

The local and global minima of f on $\mathbb{R}^n$ are defined as follows:

2.1 Definitions

Definition 2.1.
We say that x^* is a **local minimum** of f if:

$$\exists V \in \mathcal{V}(x_0) \text{ such that } \forall x \in V, f(x) \geq f(x^*).$$

Definition 2.2.
We say that x^* is a **strict local minimum** of f if:

$$\exists V \in \mathcal{V}(x_0) \text{ such that } \forall x \in V, f(x) > f(x^*).$$

Definition 2.3.
We say that x^* is a **global minimum** if:

$$\forall x \in \mathbb{R}^n, f(x) \geq f(x^*).$$

Definition 2.4.
We say that x^* is a **strict global minimum** if:

$$\forall x \in \mathbb{R}^n, f(x) > f(x^*).$$

Definition 2.5.
A sequence $(x_n)_{n\in\mathbb{N}}$ of points in $\mathbb{R}^n$ is called a **minimizing sequence** if:

$$\lim_{n\to+\infty} f(x_n) = \inf_{x\in\mathbb{R}^n} f(x).$$

Definition 2.6.
A stationary point that is neither a minimum nor a maximum is called a **singular point**.

Definition 2.0.7.
x_0 is a **stationary point** if and only if $\nabla f(x_0) = 0$.

Remark 2.1.

1. A global minimum is clearly a local minimum.
2. If we simply say "minimum", it is understood as a global minimum.
3. If we know all local minima, then the smallest one is the global minimum.
4. Any isolated optimal solution is a strict local optimal solution. The converse is not always true.

Definition: Lower Semicontinuity[11]

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a function. We say that f is **lower semicontinuous** at a point $x_0 \in \mathbb{R}^n$ if:

$$\liminf_{x\to x_0} f(x) \geq f(x_0).$$

Equivalently, for any sequence (x_k) converging to x_0:

$$f(x_0) \leq \liminf_{k\to\infty} f(x_k).$$

2.2 Existence and Uniqueness of the Problem's Solution

Let us now examine the questions of existence and uniqueness of the solution to problem (1). In finite dimension, the uniqueness of a potential solution is generally established independently of its existence, most often based on the convexity of the set K and the strict convexity of the functional f.

2.1.1 In practice

We denote problem (2.2) as:

$$\min_{x \in K} f(x) \tag{2.2}$$

and problem (2.3) as:

$$\min_{x \in \mathbb{R}^n} f(x). \tag{2.3}$$

Lemma 2.1.1. Let $K \subset \mathbb{R}^n$ be an open set. Then,

$$x^* \text{ is a solution to (1)} \Leftrightarrow \begin{cases} x^* \text{ is a solution to (2.1)} \\ x^* \in K. \end{cases}$$

In practice, to solve (1), we first solve (2.1), which is simpler, then:

1. If (2.1) has a solution $x^* \in K$, then x^* is a solution to (1). Otherwise, if $x^* \notin K$, then (1) has no solution.
2. If (2.1) has no solution, we cannot conclude anything about (1).

Definition 2.2.1 (Strong convexity). *A differentiable function $f : K \to \mathbb{R}$, with $K \subseteq E$ convex, is said to be* α-strongly convex *for some $\alpha > 0$ if for all $x, y \in K$:*

$$f(y) \geq f(x) + \nabla f(x) \cdot (y - x) + \frac{\alpha}{2} \|y - x\|^2.$$

2.3 Existence of a Solution

Theorem 2.1.1 (Existence : Weierstrass Theorem)[11]
Let K be a non-empty compact (closed and bounded) subset of $\mathbb{R}^n$, and let $f : K \to \mathbb{R}$ be a continuous function on K. Then f admits at least one minimum x^* on K. In other words,

$$\exists x^* \in K \text{ such that } \forall x \in \mathbb{R}^n, f(x) \geq f(x^*).$$

Proof.
Let (x_n) be a minimizing sequence of f on K such that

$$x_n \in K, \forall n \in \mathbb{N}, \quad \text{and} \quad \lim_{n\to+\infty} f(x_n) = \inf_{x\in K} f(x).$$

Since K is bounded, (x_n) is bounded, so we can extract a subsequence converging to $x^* \in K$ (as K is closed). By continuity,

$$\lim_{n\to+\infty} f(x_n) = f(x^*),$$

thus,

$$f(x^*) = \inf_{x\in K} f(x).$$

Therefore, f attains its minimum on K.

Theorem 2.1.2.[11]
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuous and coercive. Then f admits at least one minimum on $\mathbb{R}^n$.

Proof.
Let (x_n) be a minimizing sequence in $\mathbb{R}^n$, then

$$\lim_{n\to+\infty} f(x_n) = f(x^*),$$

thus,

$$f(x^*) = \inf_{x\in\mathbb{R}^n} f(x).$$

Hence, f attains its minimum on $\mathbb{R}^n$.

Theorem 2.1.3 (Uniqueness).[11]
If $f : \mathbb{R}^n \to \mathbb{R}$ is strictly convex, then f admits a unique minimum x^* such that

$$\forall x \in \mathbb{R}^n, f(x) \geq f(x^*).$$

Proof.
Assume f is strictly convex and suppose there exist two distinct minima x^*, x'^* with

$$f(x^*) = f(x'^*) = \inf_{x\in\mathbb{R}^n} f(x).$$

Let $x''^* = \lambda x'^* + (1-\lambda)x^*$, with $\lambda \in (0,1)$. Then

$$f(x''^*) < \lambda f(x'^*) + (1-\lambda) f(x^*) = f(x^*),$$

which is a contradiction. Therefore, $x^* = x'^*$.

Definition 2.1.1.
A function $f : \mathbb{R}^n \to \mathbb{R}$ is called *elliptic* if it is of class C^1 and if there exists $\alpha > 0$ such that

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \alpha \|x - y\|^2, \quad \forall x, y \in \mathbb{R}^n.$$

Corollary 2.1.1.
If $f : \mathbb{R}^2 \to \mathbb{R}$ is differentiable, then the following are equivalent:

1. f is α-elliptic.
2. $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\alpha}{2}\|y - x\|^2, \quad \forall x, y \in \mathbb{R}^2.$
3. $\langle \nabla f(y) - \nabla f(x), y - x \rangle \geq \alpha\|y - x\|^2, \quad \forall x, y \in \mathbb{R}^2.$

Theorem 2.1.4 (Existence and Uniqueness) [11]
Let $f : \mathbb{R}^n \to \mathbb{R}$ be of class C^1 and α-elliptic, i.e., there exists $\alpha > 0$ such that

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \alpha\|x - y\|^2, \quad \forall x, y \in \mathbb{R}^n.$$

Then f is strictly convex and coercive. In particular, problem (2.1) admits a unique solution.

Proof.
It is clear that ellipticity implies strict convexity, which in turn implies convexity. If f is α-elliptic and differentiable, using the previous characterization, we obtain

$$\langle \nabla f(x) - \nabla f(0), x \rangle \geq \alpha\|x\|^2,$$

which implies that f is coercive.

Theorem 2.1.5[11]
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function that is bounded below, lower semi-continuous, and coercive. Then, problem (I) admits at least one solution [11, 13].

Proof.
Since f is bounded below, it cannot take the value $-\infty$. Thus, there exists a minimizing sequence (x_k) of f such that:

$$\inf_{x \in \mathbb{R}^n} f(x) = \lim_{k \to +\infty} f(x_k) = d > -\infty.$$

Moreover, since f is not identically $+\infty$, we have $d < +\infty$. Let us show that the sequence (x_k) is bounded. Suppose, for contradiction, that (x_k) is unbounded. Then

$$\|x_k\| \to +\infty,$$

and since f is coercive, $\lim_{k \to +\infty} f(x_k) = +\infty$, which contradicts $\lim_{k \to +\infty} f(x_k) = d$. Therefore, (x_k) is bounded.

We can extract a convergent subsequence, say (x_{k_l}), converging to x^*. Since f is lower semi-continuous, we have:

$$\liminf_{l \to +\infty} f(x_{k_l}) \geq f(x^*).$$

But $\lim_{l \to +\infty} f(x_{k_l}) = d$, thus $f(x^*) \leq d$. By definition of d, we also have $f(x^*) \geq d$. Therefore,

$$f(x^*) = d = \inf_{x \in \mathbb{R}^n} f(x).$$

Proposition 2.1.1 (Characterization of ellipticity using the Hessian). Let f be of class C^2. Then f is elliptic if and only if there exists $\beta > 0$ such that

$$\langle \nabla^2 f(x)h, h\rangle \geq \beta \|h\|^2, \quad \forall x, h \in \mathbb{R}^n.$$

Proof.
Assume f is elliptic. Fix $h \in \mathbb{R}^n$ and define $g(x) = \langle \nabla f(x), h\rangle$. Then

$$\langle \nabla^2 f(x)h, h\rangle = \langle \nabla g(x), h\rangle = \frac{\partial g}{\partial h}(x) = \lim_{t\to 0} \frac{\langle \nabla f(x+th), h\rangle - \langle \nabla f(x), h\rangle}{t}.$$

Using bilinearity and ellipticity:

$$\langle \nabla^2 f(x)h, h\rangle = \lim_{t\to 0} \frac{\langle \nabla f(x+th) - \nabla f(x), th\rangle}{t^2} \geq \alpha \|h\|^2.$$

Hence (2.2) holds with $\beta = \alpha$.

Conversely, suppose (2.2) holds. Let $x, y \in \mathbb{R}^n$ and define $g_1(z) = \langle \nabla f(z), x - y\rangle$. Then

$$\langle \nabla f(x) - \nabla f(y), x - y\rangle = g_1(x) - g_1(y) = \langle \nabla g_1(y + \theta(x-y)), x - y\rangle$$

for some $\theta \in (0, 1)$. Moreover,

$$\nabla g_1(z) = \nabla^2 f(z)(x - y),$$

thus,

$$\langle \nabla f(x) - \nabla f(y), x - y\rangle = \langle \nabla^2 f(y + \theta(x-y))(x-y), x - y\rangle \geq \beta \|x - y\|^2.$$

Therefore, f is elliptic with $\alpha = \beta$.

Examples of elliptic functions.

1. **Case** $n = 1$**:** Any function $f : \mathbb{R} \to \mathbb{R}$ of class C^2 satisfying $\exists \alpha > 0$ such that $f''(x) \geq \alpha, \forall x \in \mathbb{R}$ is elliptic.
   (a) $f(x) = ax^2 + bx + c$ with $a > 0$.
   (b) $f(x) = x^2 + \sin(x)$ since $f''(x) = 2 - \sin(x) \geq 1$.
2. **General case (**$n \geq 1$**):** Let

   $$f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle + c,$$

   with A symmetric real matrix, $b \in \mathbb{R}^n$, $c \in \mathbb{R}$. Then

   $$\nabla f(x) = Ax - b, \quad \nabla^2 f(x) = A.$$

   As A is symmetric, its eigenvalues are real and

   $$\langle Ah, h\rangle \geq \lambda_{\min} \|h\|^2,$$

   where $\lambda_{\min}$ is the smallest eigenvalue of A.

   A is positive definite iff $\lambda_{\min} > 0$. Thus, by the proposition, f is elliptic iff A is positive definite.

Theorem 2.1.6: Characterization of Ellipticity [13]
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function that is twice differentiable in the Gateaux sense. Then, f is **elliptic** if and only if:

$$\exists \alpha > 0, \quad \forall x, h \in \mathbb{R}^n, \quad \langle \nabla^2 f(x)h, h \rangle \geq \alpha \|h\|^2.$$

Proof.
($\Rightarrow$) Assume that f is elliptic. Then by the definition of ellipticity, we have:

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \alpha \|x - y\|^2, \quad \forall x, y \in \mathbb{R}^n.$$

Taking $y = x + th$ and defining $\phi(t) = \nabla f(x + th)$, we write:

$$\langle \phi(t) - \phi(0), th \rangle \geq \alpha t^2 \|h\|^2.$$

Dividing both sides by t^2 and taking the limit as $t \to 0$, we get:

$$\lim_{t \to 0} \frac{\langle \phi(t) - \phi(0), th \rangle}{t^2} = \langle \nabla^2 f(x)h, h \rangle \geq \alpha \|h\|^2.$$

($\Leftarrow$) Conversely, assume that:

$$\langle \nabla^2 f(x)h, h \rangle \geq \alpha \|h\|^2, \quad \forall x, h \in \mathbb{R}^n.$$

Let $x, y \in \mathbb{R}^n$ be arbitrary, and define $\gamma(t) = x + t(y - x)$ for $t \in [0, 1]$. Then,

$$\nabla f(y) - \nabla f(x) = \int_0^1 \nabla^2 f(\gamma(t))(y - x)\, dt.$$

Hence,

$$\langle \nabla f(y) - \nabla f(x), y - x \rangle = \int_0^1 \langle \nabla^2 f(\gamma(t))(y - x), y - x \rangle\, dt \geq \int_0^1 \alpha \|y - x\|^2\, dt = \alpha \|y - x\|^2.$$

Thus, f is elliptic.

Proposition 2.3.1 (Strong monotonicity of the gradient)**.** *If f is α-strongly convex on K, then its gradient is α-strongly monotone:*

$$(\nabla f(y) - \nabla f(x)) \cdot (y - x) \geq \alpha \|y - x\|^2, \quad \forall x, y \in K.$$

Proposition 2.3.2 (Uniqueness of the minimizer)**.** *If f is α-strongly convex on K, then f has at most one minimizer in K.*

Remark 2.3.1. *The three equivalent statements in the theorem provide alternative ways to verify strong convexity: (i) directly from the definition, (ii) via a quadratic lower bound, and (iii) through the strong monotonicity of the gradient. This last condition is especially useful in optimization theory.*

2.4 Optimality condition

2.4.1 First order optimality condition

In this section, we focus on first-order optimality conditions. The objective is to determine criteria that characterize candidate points for being optimal solutions of a minimization problem.

More precisely, we will establish the necessary and/or sufficient conditions that a function must satisfy at a given point for it to be a local minimum. These conditions are based essentially on the analysis of the function's first derivative (gradient).

We will begin by rigorously defining the notion of a critical point, then we will state and prove the results related to first-order conditions, emphasizing their assumptions and their geometric and analytical interpretation.

2.4.2 First-order Necessary Optimality Conditions: NOC1

Given a point x^*, the continuous differentiability property of the function f provides a first way to characterize an optimal solution.

Theorem 2.1.7 [11] [13]
Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable at the point $x^* \in \mathbb{R}^n$. If x^* is a local minimum of (2.1), then

$$\nabla f(x^*) = 0.$$

Proof.

We proceed by contradiction. Suppose that $\nabla f(x^*) \neq 0$. Then, the vector $d = -\nabla f(x^*)$ is a descent direction, i.e.,

$$\nabla f(x^*)^T d = -\|\nabla f(x^*)\|^2 < 0.$$

By the descent direction theorem, there exists $\delta > 0$ such that

$$f(x^* + \alpha d) < f(x^*), \quad \forall \alpha \in]0, \delta[.$$

This contradicts the fact that x^* is a local minimum. Therefore, $\nabla f(x^*) = 0$.

2.4.3 First-order Necessary and Sufficient Optimality Condition: NSOC1

The convexity of f allows us to specify the nature (maximum or minimum) of the considered extrema. Of course, similar results can be stated for relative maxima. If f is convex, the first-order necessary condition is also sufficient.

Theorem 2.1.8[11]
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function differentiable at the point $x^* \in \mathbb{R}^n$. Then x^* is a minimum of (2.1) if and only if

$$\nabla f(x^*) = 0.$$

Proof.
We have already seen that this condition is always necessary. Let us prove that it is sufficient.
Assume that $x^* \in \mathbb{R}^n$ satisfies $\nabla f(x^*) = 0$. Since f is convex, we have:

$$\forall x \in \mathbb{R}^n, \quad f(x) \geq f(x^*) + \nabla f(x^*)^T (x - x^*) \Rightarrow f(x) \geq f(x^*).$$

Thus, x^* is a global minimum of f over $\mathbb{R}^n$.

2.5 Second-order optimality conditions

The following results are stated for relative minima involving second derivatives.

2.5.1 Necessary second-order optimality condition: NOC2

Theorem 2.1.9. Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable at $x^* \in \mathbb{R}^n$. If f has a local minimum at x^*, then:
1. $\nabla f(x^*) = 0$, 2. The Hessian matrix $H(x^*)$ is positive semi-definite.

Proof. Using second-order Taylor expansion,

$$f(x^* + \lambda d) = f(x^*) + \lambda \nabla f(x^*)^T d + \frac{1}{2}\lambda^2 d^T H(x^*) d + \lambda^2 \|d\|^2 \alpha(x^*, \lambda d),$$

where $\alpha(x^*, \lambda d) \to 0$ as $\lambda \to 0$. Since x^* is a local minimum, $\nabla f(x^*) = 0$, thus

$$\frac{1}{2} d^T H(x^*) d \geq 0, \quad \forall d \in \mathbb{R}^n.$$

Therefore, $H(x^*)$ is positive semi-definite. $\square$

2.5.2 Sufficient second-order optimality condition: COS2

Theorem 2.1.10[11] Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable at $x^* \in \mathbb{R}^n$. If
1. $\nabla f(x^*) = 0$, 2. $H(x^*)$ is positive definite,
then x^* is a local minimum.

Proof. Using second-order Taylor expansion,

$$f(x) = f(x^*) + \nabla f(x^*)^T (x - x^*) + \frac{1}{2}(x - x^*)^T H(x^*)(x - x^*) + \|x - x^*\|^2 \alpha(x^*, x - x^*),$$

where $\alpha(x^*, x - x^*) \to 0$ as $x \to x^*$. If $H(x^*)$ is positive definite, the quadratic term is strictly positive. Thus, x^* is a strict local minimum. $\square$

Example

Consider $f(x, y) = (x-5)^2 + (y-2)^2$.

Step 1. Find critical points:

$$f'_x = 2(x-5) = 0 \Rightarrow x = 5, \quad f'_y = 2(y-2) = 0 \Rightarrow y = 2.$$

Step 2. Analyze the Hessian:

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}.$$

Determinant is $4 > 0$ with positive eigenvalues. Thus $(5, 2)$ is a local minimum.

Reference. See Rockafellar (1970) *Convex Analysis*.

2.6 Theorem: Global Optimality of Local Minima for Convex Functions

Theorem 2.1.11[11], Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function.
1. Every local minimum of f is also a global minimum.
2. Furthermore, if f is differentiable, every stationary point x^* (i.e., $\nabla f(x^*) = 0$) is a global minimum of f.

Proof.

(i) Let x^* be a local minimum. For any $x \in \mathbb{R}^n$, define $x_\lambda = \lambda x + (1-\lambda)x^*$. By local minimality, for λ small enough,

$$f(x^*) \leq f(x_\lambda).$$

By convexity,

$$f(x_\lambda) \leq \lambda f(x) + (1-\lambda) f(x^*).$$

Thus,

$$f(x^*) \leq \lambda f(x) + (1-\lambda) f(x^*),$$

which implies

$$0 \leq \lambda(f(x) - f(x^*)),$$

and hence $f(x^*) \leq f(x)$ for all x.

(ii) If f is differentiable and $\nabla f(x^*) = 0$, by the first-order condition of convex functions,

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

so x^* is a global minimum. □

Reference. See Rockafellar (1970), Theorem 27.1; Boyd and Vandenberghe (2004), Section 3.1.2.

Exercise

Exercise 2.1.6. Let the function f be defined by:

$$f(x,y) = (y-1)\ln(y-1) - \ln(x) + x^2 - xy + 2y^2 - 7y - \frac{3}{2}x + 3.$$

(a) Give the domain of definition D_f of f and draw this set.

(b) Is the set D_f convex? Is it open?

(c) Show that the function $\varphi : u \to \ln(u)$ is convex on its domain.

(d) Deduce the convexity of f on D_f.

(e) Show that (2,2) is a critical point.

(f) Deduce the nature of (2,2).

(g) Does the function f admit a global maximum on D_f?

Solution

Define:

$$f(x,y) = (y-1)\ln(y-1) - \ln(x) + x^2 - xy + 2y^2 - 7y - \frac{3}{2}x + 3.$$

(a) **Domain:**
$x > 0$ and $y > 1$, thus:

$$D_f = \{(x,y) \in \mathbb{R}^2 \mid x > 0, y > 1\}.$$

(b) **Convexity and openness:**
Domain is convex (intersection of half-planes) and open.

(c) **Convexity of $\varphi(u) = \ln(u)$:**
$\varphi''(u) = -\frac{1}{u^2} < 0$, thus concave.

(d) **Convexity of f:**
Due to quadratic terms (full proof via Hessian positivity).

(e) **Critical point at (2,2):**
$\frac{\partial f}{\partial x} = -\frac{1}{2} + 4 - 2 - \frac{3}{2} = 0,$
$\frac{\partial f}{\partial y} = \ln(1) + 1 - 2 - 7 + 8 = 0.$

(f) **Nature:**
Positive definite Hessian implies strict local minimum (also global by convexity).

(g) **Global maximum:**
No global maximum, function tends to $+\infty$ on D_f.

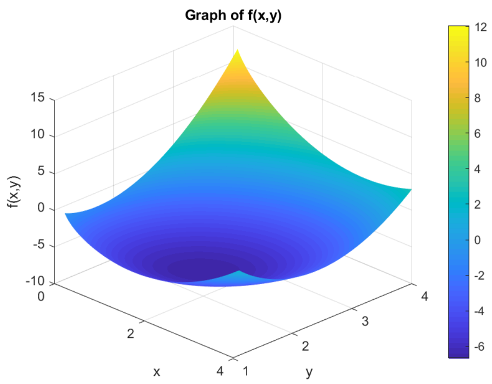


Figure 2.1: Graph of $f(x, y)$

Sheet 2

Exercise 01.
Show that a norm is convex.

Exercise 02.
Show that the indicator function of a set Ω defined by:

$$1_\Omega(x) = \begin{cases} 0 & \text{if } x \in \Omega \\ +\infty & \text{otherwise} \end{cases}$$

is convex if and only if O is convex.

Exercise 03.
Let U be a convex subset of a vector space V. Show that $f : U \subset V \to \mathbb{R}$ is convex if and only if the set:

$$\mathrm{epi}(f) = \{(v, a) \in U \times \mathbb{R} \mid a \geq f(v)\}$$

is a convex subset of $U \times \mathbb{R}$.

Exercise 04.
Let F be a function from $\mathbb{R}^n$ to $\mathbb{R}$. Define the function $\phi : \mathbb{R}^n \to \mathbb{R}$ by:

$$\phi(\alpha) = \frac{F(u + \alpha v) - F(v)}{\alpha}, \quad \forall \alpha > 0, \forall (u, v) \in \mathbb{R}^n \times \mathbb{R}^n.$$

Show that if F is convex then ϕ is increasing.

Exercise 05.
Let $\{f_i\}_{i\in I}$ be any family of convex functions $f_i : U \subset V \to \mathbb{R}$. Prove that the function $\sup_{x\in R^n} f_i$ is convex.

Exercise 06.
Prove Young's inequality: For all $a, b > 0$, and for all $p, q \in \mathbb{N}$ such that $\frac{1}{p} + \frac{1}{q} = 1$.
$ab \leq \dfrac{1}{p}a^p + \dfrac{1}{q}b^q$

Exercise 07.
Let f be a convex function from $\mathbb{R}^n$ to $\mathbb{R}$. Show that:

$$\forall (\lambda_i)_{i=1}^n, (x_i)_{i=1}^n \in (\mathbb{R}^n)^n \text{ such that } \sum_{i=1}^n \lambda_i = 1, f\left(\sum_{i=1}^n \lambda_i x_i\right) \leq \sum_{i=1}^n \lambda_i f(x_i).$$

Exercise 08. (Characterization of convexity)
Let $\Omega \subset \mathbb{R}^n$ be an open set, $U \subset \Omega$ convex, and $f : \Omega \to \mathbb{R}$ a C^1 function. Then the following are equivalent:

1. f is convex on U.
2. $f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle$, $\forall x, y \in U$.
3. ∇f is monotone on U.

Exercise 09.
Let f be C^2 on U with U convex. Then f is convex on U if and only if:

$$\langle \nabla^2 f(x)(y - x), y - x\rangle \geq 0, \quad \forall x, y \in U.$$

Exercise 10.

Let f be the real-valued function defined by:

$$\forall x \in \mathbb{R}, \quad f(x) = \sqrt{x^4+1} - x + 3.$$

Show that f admits a global minimum on $\mathbb{R}$.

Exercise 11.
Let $f : \mathbb{R} \to \mathbb{R}$ be the function defined by:

$$\forall x \in \mathbb{R}, \quad f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\theta)^2}{2}}.$$

We fix $x_1, \ldots, x_n \in \mathbb{R}$. Solve the following problem:

$$\max_{\theta \in \mathbb{R}} g(\theta) = \prod_{i=1}^{n} f(x_i).$$

Exercise 12
We consider the function

$$f(x) = \tfrac{1}{2} x^T A x - b^T x + c,$$

where A is a real symmetric matrix.

1. Compute $\nabla f(x)$ and $\nabla^2 f(x)$.
2. Let $\lambda_{\min}, \lambda_{\max}$ be the smallest and largest eigenvalues of A. Show that
$$\forall x \in \mathbb{R}^n, \quad \lambda_{\min}\|x\|^2 \leq \langle Ax, x\rangle \leq \lambda_{\max}\|x\|^2.$$
3. Suppose that A is positive definite:
   (a) Show that f is coercive.
   (b) Deduce that f admits a (unique) global minimum.
   (c) Show that x^* is a solution of the system $Ax = b$ if and only if x^* achieves the (unique) minimum of $f(x)$ on $\mathbb{R}^n$.
4. If A is not positive semidefinite, show that f does not admit a minimum on $\mathbb{R}^n$.
5. **Application:** Let
$$f : \mathbb{R}^3 \to \mathbb{R}, \quad f(x, y, z) = x^2 + y^2 + z^2 + xy + yz + xz - 3x - 4y + 4.$$
   (a) Express f in the form of a quadratic function
   $$f(x) = \tfrac{1}{2} x^T A x - b^T x + c,$$
   where A is a symmetric matrix, $b \in \mathbb{R}^3$, and $c \in \mathbb{R}$.
   (b) Solve the minimization problem
   $$\min_{x \in \mathbb{R}^3} f(x).$$

Sheet 02 (Solutions)

Exercise 01.
Solution. Let $\|\cdot\|$ be a norm. To prove it is convex, we show that:

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

Using triangle inequality and homogeneity,

$$\|\lambda x + (1-\lambda)y\| \leq \|\lambda x\| + \|(1-\lambda)y\| = \lambda\|x\| + (1-\lambda)\|y\|.$$

Thus, a norm is convex.

Exercise 02.
Solution. If Ω is convex, then for $x, y \in \Omega$ and $\lambda \in [0,1]$, $\lambda x + (1-\lambda)y \in \Omega$, thus

$$1_\Omega(\lambda x + (1-\lambda)y) = 0 \leq \lambda 1_\Omega(x) + (1-\lambda)1_\Omega(y) = 0.$$

If Ω is not convex, take $x, y \in \Omega$ but $\lambda x + (1-\lambda)y \notin O$, then 1_Ω fails convexity. Hence, 1_Ω is convex iff Ω is convex.

Exercise 03.
Solution. If f is convex, for any $(v_1, a_1), (v_2, a_2) \in epi(f)$ and $\lambda \in [0,1]$,

$$\lambda a_1 + (1-\lambda)a_2 \geq \lambda f(v_1) + (1-\lambda)f(v_2) \geq f(\lambda v_1 + (1-\lambda)v_2).$$

Hence $(\lambda v_1 + (1-\lambda)v_2, \lambda a_1 + (1-\lambda)a_2) \in epi(f)$. The converse follows similarly.

Exercise 04.
Solution. Given F convex, consider

$$\phi(\alpha) = \frac{F(u + \alpha v) - F(v)}{\alpha}.$$

Since the secant slope of a convex function is increasing, ϕ is increasing in α.

Exercise 05.
Solution. Let $f(x) = \sup_i f_i(x)$. For x, y and $\lambda \in [0,1]$,

$$\begin{aligned}
&f(\lambda x + (1-\lambda)y) = \sup_i f_i(\lambda x + (1-\lambda)y) \\
&\leq \sup_i [\lambda f_i(x) + (1-\lambda)f_i(y)] \\
&\leq \lambda \sup_i f_i(x) + (1-\lambda)\sup_i f_i(y) = \lambda f(x) + (1-\lambda)f(y).
\end{aligned}$$

Hence f is convex.

Exercise 06.
Solution. Young's inequality:

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q}, \quad \text{with } \frac{1}{p} + \frac{1}{q} = 1.$$

Follows from convexity of $x \mapsto x^p$ and its conjugate.

Exercise 07.
Solution. By Jensen's inequality for convex f,

$$f\left(\sum \lambda_i x_i\right) \leq \sum \lambda_i f(x_i).$$

Exercise 08.

Solution. (1)$\Rightarrow$(2): First order characterization of convexity. (2)$\Rightarrow$(3): Differentiating shows monotonicity. (3)$\Rightarrow$(1): Integration along segment yields convexity.

Exercise 09.

Solution. f convex iff Hessian is positive semi-definite:

$$\langle \nabla^2 f(x)(y-x), y-x \rangle \geq 0, \forall x, y.$$

Exercise 10.

We have

$$\lim_{|x|\to+\infty} f(x) = \lim_{|x|\to+\infty} \left(\sqrt{x^4+1} - x + 3\right) = +\infty.$$

The function $f(x)$ is coercive and continuous. Therefore, it admits at least one global minimum point on $\mathbb{R}$.

Exercise 11.

First, we write

$$g(\theta) = \left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left(-\frac{(x_1-\theta)^2}{2} - \cdots - \frac{(x_n-\theta)^2}{2}\right).$$

Then, the derivative is

$$g'(\theta) = \left(\sum_{i=1}^n x_i - n\theta\right) g(\theta).$$

Since $g(\theta) > 0$, we have

$$g'(\theta) = 0 \quad \Longleftrightarrow \quad \theta^* = \frac{1}{n}\sum_{i=1}^n x_i.$$

Now we compute the second derivative:

$$g''(\theta) = \left(\sum_{i=1}^n x_i - n\theta\right) g'(\theta) - n g(\theta).$$

In particular, at the critical point θ^*:

$$g''(\theta^*) = -n g(\theta^*) < 0.$$

Therefore,

$$\boxed{\theta^* = \frac{1}{n}\sum_{i=1}^n x_i}$$

is the unique global maximizer of g.

Exercise 12.

We consider the function

$$f(x) = \tfrac{1}{2}x^T Ax - b^T x + c,$$

with A a real symmetric matrix.

We have

$$\nabla f(x) = Ax - b, \quad \nabla^2 f(x) = A.$$

Let $x \in \mathbb{R}^n$. Since A is symmetric, there exists an orthonormal eigenbasis $\{v_i\}_{i=1}^n$ of $\mathbb{R}^n$ with eigenvalues $(\lambda_i)_{i=1}^n$. We can write

$$x = \sum_{i=1}^n \beta_i v_i.$$

Then

$$\langle Ax, x\rangle = \left\langle A\sum_{i=1}^n \beta_i v_i, \sum_{i=1}^n \beta_i v_i \right\rangle = \sum_{i=1}^n \beta_i^2 \lambda_i.$$

Hence

$$\lambda_{\min} \sum_{i=1}^n \beta_i^2 \le \sum_{i=1}^n \beta_i^2 \lambda_i \le \lambda_{\max} \sum_{i=1}^n \beta_i^2,$$

which yields

$$\lambda_{\min}\|x\|^2 \le \langle Ax, x\rangle \le \lambda_{\max}\|x\|^2.$$

Since $A \succ 0$, we have $\lambda_{\min} > 0$. Then

$$\langle Ax, x\rangle \ge \lambda_{\min}\|x\|^2.$$

Thus

$$f(x) = \tfrac{1}{2}x^T Ax - b^T x + c \ge \tfrac{\lambda_{\min}}{2}\|x\|^2 - \|b\|\|x\| + c.$$

As $\|x\| \to +\infty$, the quadratic term dominates, and

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

Therefore, f is coercive and continuous, hence it admits at least one global minimum. Since A is positive definite, f is strictly convex, which implies the minimum is unique.

Moreover, x^* solves $Ax = b$ if and only if

$$\nabla f(x^*) = Ax^* - b = 0,$$

i.e. x^* is exactly the unique global minimizer of f.

Since $\nabla^2 f(x) = A$ is constant, if A is not positive semidefinite, the quadratic form $\frac{1}{2}x^T Ax$ is not bounded below. Hence f does not admit a global minimum on $\mathbb{R}^n$.

We consider

$$f(x, y, z) = x^2 + y^2 + z^2 + xy + yz + xz - 3x - 4y + 4.$$

We compute

$$\nabla f(x, y, z) = \begin{pmatrix} 2x + y + z - 3 \\ x + 2y + z - 4 \\ x + y + 2z \end{pmatrix}, \qquad f(0, 0, 0) = 4.$$

Thus

$$b = \nabla f(0,0,0) = \begin{pmatrix} -3 \\ -4 \\ 0 \end{pmatrix}, \qquad c = 4,$$

and

$$A = \nabla^2 f(x,y,z) = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}.$$

Therefore,

$$f(x,y,z) = \tfrac{1}{2}\,(x,y,z)A(x,y,z)^T - b^T(x,y,z)^T + c.$$

By Sylvester's criterion:

$$\det(2) = 2 > 0, \quad \det\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = 3 > 0, \quad \det(A) = 4 > 0.$$

Thus A is positive definite. Therefore f admits a unique minimizer $x^* \in \mathbb{R}^3$, which is the unique solution of $Ax = b$:

$$\begin{cases} 2x + y + z = -3, \\ x + 2y + z = -4, \\ x + y + 2z = 0. \end{cases}$$

Solving this system yields

$$(x,y,z) = \left(-\tfrac{5}{4},\ -\tfrac{9}{4},\ \tfrac{7}{4}\right).$$

Chapter 3

Algorithms

Introduction

In this chapter, we will present some algorithms that allow us to compute (approximately) the solution(s) of the initial problem (P). Optimization algorithms are mathematical procedures that aim to determine the set of input parameters of a function giving it its maximum or minimum value. More precisely, they seek to solve:

$$\inf_{x\in\mathbb{R}} f(x) \tag{3.1}$$

where f is a real-valued function called the objective function.

With $X^* = (x_1^*, x_2^*, x_3^*, \ldots)$ being the coordinates of the critical point. Optimization algorithms are used to solve problems of various kinds, such as finding the zeros of nonlinear functions, fitting experimental data using linear and nonlinear least squares criteria, solving systems of equations with one or more variables, etc.

In general, the search for extrema is carried out by computing the first derivatives (the gradient of the function) and the second derivatives (the Hessian of the function). Metaheuristics are a class of optimization algorithms that attempt to obtain an approximate value of the global optimum in difficult optimization problems. However, they do not provide any guarantee on the reliability of the result.

We assume that x^* exists (possibly unique) and aim to find a numerical approximation of x^* by constructing a sequence:

$$\{x^{(k)}\}_{k\in\mathbb{N}} \subset \mathbb{R}^n \tag{3.2}$$

such that

$$x^{(k)} \to x^* \quad \text{as } k \to +\infty. \tag{3.3}$$

The principle is to construct an iterative algorithm of the form:

$$x_{k+1} = x_k - \rho_k d_k \tag{3.4}$$

where d_k is the descent direction and ρ_k is the step size. The step size can be fixed (possibly the same for all iterations, in which case it is called a variable step size method) or computed at each iteration to minimize f in the direction d_k (in which case it is called an optimal step size method).

To approach the optimal solution of problem (3.2) (in the general case, this is a point where the necessary optimality conditions of f are satisfied with a certain precision), one naturally moves from the point x_k in the direction of decrease of the function f.

Definition 3.1 (Algorithm).
An algorithm is defined as an application A from $\mathbb{R}^n$ allowing the generation of a sequence of elements in $\mathbb{R}^n$ by the formula:
Initialization step:
Given $x_0 \in \mathbb{R}^n$, set $k = 0$.
Iteration:

$$x_{k+1} = A(x_k) \tag{3.5}$$

Increment $k = k + 1$.
Writing an algorithm is defining a sequence $(x_k)_{k\in\mathbb{N}}$ in $\mathbb{R}^n$. Studying its convergence means studying the convergence of $(x_k)_{k\in\mathbb{N}}$.

Definition 3.2 (Convergence of an algorithm).
An algorithm A converges if the sequence $(x_k)_{k\in\mathbb{N}}$ converges to a limit x^*.

The error is defined as:

$$e_k = x_k - x^* \tag{3.6}$$

Definition 3.3 (Rate of convergence of an algorithm).
Let $(x_k)_{k\in\mathbb{N}}$ be a sequence converging to x^* generated by algorithm A. Its convergence is:
Linear if

$$\exists C \in]0,1[, \exists k_0 \in \mathbb{N}, \forall k \geq k_0, \quad \|e_{k+1}\| \leq C\|e_k\| \tag{3.7}$$

Superlinear if

$$\lim_{k\to+\infty} \frac{\|e_{k+1}\|}{\|e_k\|} = 0 \tag{3.8}$$

If $\frac{\|e_{k+1}\|}{\|e_k\|}$ converges geometrically to zero, the algorithm's convergence is geometric.
Of order p if

$$\exists C > 0, \exists k_0 \in \mathbb{N}, \forall k \geq k_0, \quad \|e_{k+1}\| \leq C\|e_k\|^p \tag{3.9}$$

If $p = 2$, the convergence is quadratic.

Convergence is **local** if it only occurs for starting points x_0 in a neighborhood of x^*. Otherwise, it is **global**.

Remark 3.1.
If $e_k \neq 0$, linear convergence means $\frac{e_{k+1}}{e_k} = O(1)$, while superlinear convergence implies $\frac{e_{k+1}}{e_k} = o(1)$. Similarly, an algorithm of order p satisfies $\frac{e_{k+1}}{e_k^p} = O(1)$.

It is desirable to have the highest possible convergence rate to achieve the solution with minimal iterations for a given accuracy.

3.1 Algorithms and Methods for Solving Unconstrained Optimization Problems

Iterative optimality methods (or algorithms) are part of a broader class of numerical methods called **descent methods**.

A **descent direction algorithm** is a differentiable optimization algorithm designed to minimize a real differentiable function defined on a Euclidean space (for example, $\mathbb{R}^n$, equipped with an inner product) or, more generally, on a Hilbert space. The algorithm is *iterative* and thus proceeds by successive improvements. At each current point, a move is made along a descent direction in order to decrease the function value.

3.1.1 Descent Direction Method – One Iteration

Algorithm 1 Descent Direction Method – One Iteration

1: **Step 0 (Initialization):** At the beginning of iteration k, we have an iterate $x_k \in \mathbb{R}^n$.
2: **Step 1 (Stopping test):** If $\nabla f(x_k) \simeq 0$, stop the algorithm.
3: **Step 2 (Choice of descent direction):** Choose a descent direction $d_k \in \mathbb{R}^n$.
4: **Step 3 (Line search):** Determine a step size $\rho_k > 0$ along d_k such that f decreases sufficiently.
5: **Step 4 (Update):** If the line search succeeds, set

$$x_{k+1} = x_k + \rho_k d_k.$$

Replace k by $k + 1$ and return to Step 1.

Examples of choosing a descent direction:

1. If
$$d_k = -\nabla f(x_k), \quad \text{and } \nabla f(x_k) \neq 0, \tag{3.10}$$
we obtain the **gradient method**.

2. In the case
$$d_k = -(H(x_k))^{-1} \nabla f(x_k), \tag{3.11}$$
we obtain **Newton's method**, where the Hessian matrix $H(x_k)$ is positive definite.

Example of choosing the step size ρ_k:
Generally, ρ_k is chosen optimally, that is, it must satisfy:

$$f(x_k + \rho_k d_k) \leq f(x_k + \rho d_k), \quad \forall \rho \in [0, +\infty[. \tag{3.12}$$

In other words, at each iteration we solve a **one-variable minimization problem**, called a **line search**.

Gradient Descent Example

Problem:
Minimize

$$f(x, y) = x^2 + xy + y^2 - 6x - 9y. \tag{3.13}$$

Gradient:

$$\nabla f(x, y) = \begin{pmatrix} 2x + y - 6 \\ x + 2y - 9 \end{pmatrix}.$$

Setting the gradient to zero yields the minimum at $(1, 4)$.

Gradient Descent Iterations (Step size $\rho = 0.1$):

- **Iteration 0:**

$$\begin{aligned} x_0 &= 0, \quad y_0 = 0 \\ \nabla f(0, 0) &= (-6, -9) \\ d_0 &= (6, 9) \\ x_1 &= 0 + 0.1 \times 6 = 0.6 \\ y_1 &= 0 + 0.1 \times 9 = 0.9 \end{aligned}$$

- **Iteration 1:**

$$\begin{aligned} \nabla f(0.6, 0.9) &= (-3.9, -6.6) \\ d_1 &= (3.9, 6.6) \\ x_2 &= 0.6 + 0.1 \times 3.9 = 0.99 \\ y_2 &= 0.9 + 0.1 \times 6.6 = 1.56 \end{aligned}$$

- **Iteration 2:**

$$\begin{aligned} \nabla f(0.99, 1.56) &= (-2.46, -4.89) \\ d_2 &= (2.46, 4.89) \\ x_3 &= 0.99 + 0.1 \times 2.46 = 1.236 \\ y_3 &= 1.56 + 0.1 \times 4.89 = 2.049 \end{aligned}$$

The method converges to the minimum at (1,4).

Conceptual Schematic:

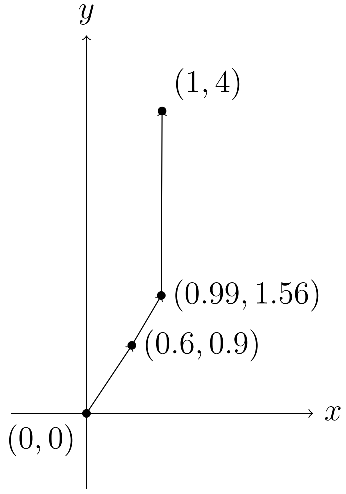


Figure 3.1: Conceptual schematic of the illustration.

MATLAB Code: Gradient Descent Example

Below is the MATLAB program implementing the gradient descent method for:

$$f(x, y) = x^2 + xy + y^2 - 6x - 9y.$$

Program:

```
% Gradient Descent for f(x,y) = x^2 + xy + y^2 -6x -9y

% Initialization
x = 0;
y = 0;
rho = 0.1;

for k = 1:10
    % Compute gradient
    gradx = 2*x + y - 6;
    grady = x + 2*y - 9;

    % Display iteration
    fprintf('Iteration %d: x=%.4f, y=%.4f, f=%.4f\n', k, x, y, x^2 + x*y
        + y^2 -6*x -9*y);

    % Check stopping criterion
    if norm([gradx; grady]) < 1e-4
        break;
    end

    % Descent direction
    dx = -gradx;
    dy = -grady;

    % Update
    x = x + rho*dx;
    y = y + rho*dy;
end

```

```
fprintf('Minimum approximated at x=%.4f, y=%.4f\n', x, y);
```

Listing 3.1: Gradient Descent for a two-variable quadratic function

—

Sample Output:

```
Iteration 1: x=0.0000, y=0.0000, f=0.0000
Iteration 2: x=0.6000, y=0.9000, f=-7.4100
Iteration 3: x=0.9900, y=1.5600, f=-9.6857
Iteration 4: x=1.2360, y=2.0490, f=-10.4976
...
Minimum approximated at x=1.0000, y=4.0000
```

The algorithm converges to the optimal point at $(1, 4)$.

Theoretical Example: Gradient Descent

Consider the problem

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

where f is convex and differentiable with L-Lipschitz continuous gradient. The gradient descent update is:

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

If f is additionally strongly convex with parameter $\mu > 0$, then

$$\|x_k - x^*\| \leq (1 - \rho\mu)^k \|x_0 - x^*\|,$$

showing linear convergence to the minimizer x^*.

Example: Quadratic Function
Let

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

where Q is symmetric positive definite. Then,

$$\nabla f(x) = Qx - b, \quad x^* = Q^{-1} b.$$

Gradient descent iteration:

$$x_{k+1} = x_k - \rho(Qx_k - b).$$

Convergence is guaranteed if $0 < \rho < \frac{2}{L}$, where $L = \lambda_{\max}(Q)$ is the largest eigenvalue of Q.

Gradient Descent Example in 3D

Problem:
Minimize

$$f(x,y,z) = x^2 + y^2 + z^2 + xy + yz + xz - 4x - 5y - 6z.$$

—
Gradient:

$$\nabla f(x,y,z) = \begin{pmatrix} 2x + y + z - 4 \\ x + 2y + z - 5 \\ x + y + 2z - 6 \end{pmatrix}.$$

—
Solution:
Solving

$$\nabla f(x,y,z) = 0$$

yields

$$(x^*, y^*, z^*) = (0.25, 1.25, 2.25).$$

—

Conceptual 3D Schematic:

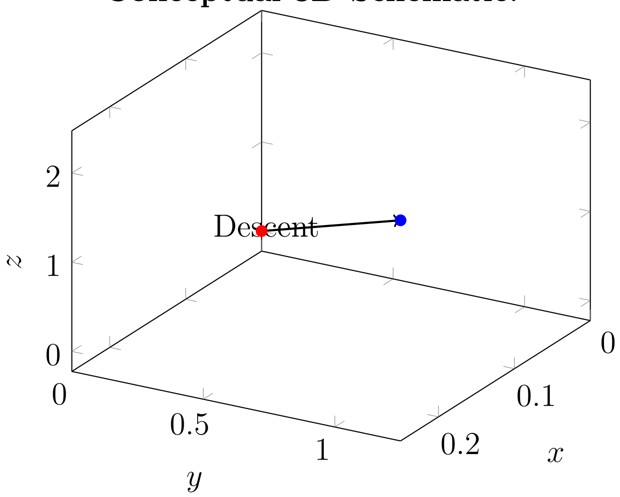


Figure 3.2: Conceptual 3D schematic of the illustration.

Advantages and Disadvantages of Gradient Descent Method

Gradient descent is widely used due to its simplicity and general applicability [14, 15]. However, it also presents limitations that motivate the development of advanced optimization algorithms [16].

Advantages:

1. Simple implementation: easy to code and understand for various optimization problems [14].
2. Low computational cost per iteration: requires only gradient evaluation [15].
3. Applicable to a wide range of differentiable problems (convex and non-convex) [14].
4. Flexible descent directions (steepest descent, Newton, quasi-Newton) [16].
5. Guaranteed convergence under convexity with proper step size [15].
6. Foundation for advanced methods such as conjugate gradient and quasi-Newton [14].

Disadvantages:

1. Slow convergence rate, especially for ill-conditioned problems [14].
2. Sensitive to step size: small steps cause slow progress, large steps cause divergence [15].
3. Sensitive to initial guess: may converge to local minima in non-convex problems [16].
4. Line search (optimal step size) may be computationally expensive [14].
5. Does not exploit curvature information (no second derivatives used) [15].
6. Limited performance in high-dimensional problems [16].

3.1.2 Gradient Method

Principle

The **gradient method** is an iterative descent technique used to find the minimum of a given function. The most natural way to determine a descent direction is by using the derivative. Thus, to find the direction of descent, we use the opposite direction of the derivative. In this approach, we take:

$$\Delta X = -\nabla f(X)^T.$$

The corresponding iterative relation for the gradient method is:

$$f(X_{k+1}) \approx f(X_k) + \rho_k \nabla f(X_k) \Delta X_k.$$

If ρ_k is constant, this method is called the *fixed step gradient method.* When ρ_k varies, it is called the *variable step gradient method.*

Algorithm 2 Gradient Algorithm

1: **Step 0 (Initialization):** Set $k = 0$. Choose initial point $x_0 \in \mathbb{R}^n$ and step size $\rho_0 > 0$.
2: **Step 1 (Iteration):**
$$x_{k+1} = x_k - \rho_k \nabla J(x_k).$$
3: **Step 2 (Stopping criterion):** If
$$\|x_{k+1} - x_k\| < \epsilon,$$
stop the algorithm; otherwise, set $k = k + 1$ and return to Step 1.

Throughout what follows, ϵ is a given small positive real number representing the desired precision. This method has the advantage of being very easy to implement. Unfortunately, its convergence conditions are quite restrictive (mainly strict convexity), and in general, the method is rather slow.

Theorem 3.1.1 (Convergence Criterion [17])**.** *Let J be a C^1 function from $\mathbb{R}^n$ to $\mathbb{R}$, coercive and strictly convex. Suppose there exists a strictly positive constant M such that*
$$J(x) - J(y) \geq M\|x - y\|^2, \quad \forall x, y \in \mathbb{R}^n.$$
Then, if the step size ρ_k is chosen in an interval $[\alpha_1, \alpha_2]$ such that
$$0 < \alpha_1 < \alpha_2 < \frac{2}{M},$$
the gradient method converges to the minimum of J.

Remark. When J satisfies the above condition, the constant step gradient algorithm can also be interpreted as the method of successive approximations applied to finding the fixed point of the function
$$S(x) = x - \rho \nabla J(x),$$
where $\nabla J(x) = 0$. Indeed, S is Lipschitz continuous with constant $(1 - \rho M)$. Thus, it is a strict contraction if the contraction factor is in $(0, 1)$. Therefore, it has a unique fixed point, and the convergence is that of a geometric series with ratio $(1 - \rho M)$. This is optimal for $\rho = \frac{2}{M}$.

In practice, the *constant step gradient method* is most often used. However, the step can vary at each iteration, leading to the *variable step gradient method.*

The *optimal step gradient method* chooses the step size that minimizes the cost function along the chosen descent direction. More precisely, step 2 becomes:
$$x_{k+1} = x_k - \rho_k \nabla J(x_k),$$
where ρ_k minimizes

$$\rho \mapsto J(x_k - \rho \nabla J(x_k)).$$

In practice, the exact minimum is not computed, and ρ_k is determined by performing a line search according to a specific rule, for example.

Algorithm 3 Wolfe Line Search Rule

Step 1 (Initialization): Set $\rho = 1$, $\rho^+ = +\infty$, $\rho^- = 0$. Choose $0 < \alpha_1 < \alpha_2 < 1$.
Step 2 (Wolfe conditions test): If

$$\phi(\rho) \leq \phi(0) + \alpha_1 \rho \phi'(0),$$

and

$$\phi'(\rho) \geq \alpha_2 \phi'(0),$$

then stop: set $\rho_k = \rho$.
Step 3 (Otherwise):
if $\phi(\rho) > \phi(0) + \alpha_1 \rho \phi'(0)$ **then**
    Set $\rho^+ = \rho$.
else if $\phi(\rho) \leq \phi(0) + \alpha_1 \rho \phi'(0)$ **and** $\phi'(\rho) < \alpha_2 \phi'(0)$ **then**
    Set $\rho^- = \rho$.
end if Go to Step 4.
Step 4 (Choose a new ρ):
if $\rho^+ = +\infty$ **then**
    Set $\rho = 2\rho$.
else
    Set $\rho = (\rho^- + \rho^+)/2$.
end if Return to Step 2.

Example 3.1. The conditions of the theorem may seem complicated, so we provide an example. Let J be the function from $\mathbb{R}^n$ to $\mathbb{R}$, already mentioned several times (because it plays an important role), defined by:

$$J(x) = \frac{1}{2}(Ax, x) - (b, x),$$

where A is a square, symmetric, and positive definite matrix, and $b \in \mathbb{R}^n$. This function J satisfies the hypotheses of the above theorem with m and M being the smallest and largest eigenvalues of A (respectively).

Remark 3.3. The notion of ellipticity is very important because it determines the convergence of most algorithms that will be described later. However, the convergence conditions we provide are always sufficient conditions. The algorithm converges if they are satisfied, but it may still converge even if they are not...

In practice, m and M are not computed. To find the convergence interval of ρ, several tests are performed for different values. Non-convergence generally results in either an explosion of the solution (clearly tending to $+\infty$) or oscillations (periodic or not) that prevent the sequence of iterates from converging to a value.

Gradient Method with Optimal Step Size

The method is as follows:

$$d_k = -\nabla f(x_k), \quad x_{k+1} = x_k + \rho_k d_k,$$

where ρ_k is chosen by the minimization rule. It consists in choosing, at each iteration, ρ_k as the optimal solution of the one-dimensional minimization problem of f along the half-line defined by the point x_k and the direction d_k. Therefore, ρ_k is chosen such that:

$$f(x_k + \rho_k d_k) = \min_{\rho\in\mathbb{R},\rho>0} f(x_k + \rho d_k),$$

assuming that such a minimum exists.

Remark 3.0.1. We perform iteration (3.3) in the case where $\nabla f(x_k) \neq 0$, so the minimization problem (3.4) makes sense. We have the following result:

Theorem 3.1.2 (3.0.1)**.** *Let f be a C^1 function from $\mathbb{R}^n$ to $\mathbb{R}$, coercive and strictly convex. We suppose that there exists a constant $M > 0$ such that*

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n,\ \|\nabla f(x) - \nabla f(y)\| \leq M\|x - y\|.$$

Then, if we choose the step ρ_k in an interval $[\beta_1, \beta_2]$ with $0 < \beta_1 < \beta_2 < \frac{2}{M}$, the gradient method converges towards the minimum of f.

Proof. The function f admits a unique minimum x_0 on $\mathbb{R}^n$ characterized by $\nabla f(x_0) = 0$ since f is strictly convex. Let us show that the sequence (x_k) generated by the algorithm converges to x_0. We have:

$$f(y) = f(x) + (\nabla f(x), y - x) + \int_0^1 (\nabla f(x + t(y - x)) - \nabla f(x), y - x)\, dt.$$

Applying this relation to $y = x_{k+1}$, $x = x_k$, we obtain

$$f(x_{k+1}) = f(x_k) + (\nabla f(x_k), x_{k+1} - x_k) + \int_0^1 (\nabla f(x_k + t(x_{k+1} - x_k)) - \nabla f(x_k), x_{k+1} - x_k)\, dt.$$

Since $x_{k+1} = x_k - \rho_k \nabla f(x_k)$, we get

$$f(x_{k+1}) - f(x_k) \leq -\frac{1}{\rho_k}\|x_{k+1} - x_k\|^2 + \int_0^1 \|\nabla f(x_k + t(x_{k+1} - x_k)) - \nabla f(x_k)\| \cdot \|x_{k+1} - x_k\|\, dt.$$

Using the Lipschitz condition, we get:

$$\leq -\frac{1}{\rho_k}\|x_{k+1} - x_k\|^2 + \frac{M}{2}\|x_{k+1} - x_k\|^2 = \left(\frac{M}{2} - \frac{1}{\rho_k}\right)\|x_{k+1} - x_k\|^2.$$

If we choose the step ρ_k in an interval $[\beta_1, \beta_2]$ with $0 < \beta_1 < \beta_2 < \frac{2}{M}$, we obtain:

$$f(x_{k+1}) - f(x_k) \leq \left(\frac{M}{2} - \frac{1}{\rho_k}\right)\|x_{k+1} - x_k\|^2.$$

The sequence $f(x_k)$ is therefore strictly decreasing and bounded below because

$$f(x_k) \geq f(x_0), \quad \forall k.$$

Hence, it is convergent. This implies on the one hand that $(f(x_{k+1}) - f(x_k))$ tends to 0 and on the other hand that the sequence (x_k) is bounded (since f is coercive). We can thus extract a subsequence converging to x. Moreover,

$$\|x_{k+1} - x_k\|^2 \leq \left(\frac{M}{2} - \frac{1}{\rho_k}\right)^{-1} (f(x_k) - f(x_{k+1})).$$

Therefore, $\|x_{k+1} - x_k\| \to 0$. Consequently,

$$\nabla f(x_k) = \frac{x_k - x_{k+1}}{\rho_k} \to 0.$$

By continuity of ∇f, we deduce that x is the unique minimum x_0 of f. Since this is true for any accumulation point of the sequence (x_k), it proves that the entire sequence (x_k) converges towards x_0.

Fixed Step Gradient Method

We can use a step fixed a priori $\rho > 0$, for all k, and we then obtain the simple gradient method:

$$d_k = -\nabla f(x_k), \quad x_{k+1} = x_k + \rho d_k.$$

For $f \in C^1$, this method converges if ρ is chosen sufficiently small.

Choice of step:

- A well-chosen step gives results similar to those obtained by the steepest descent.
- A smaller step reduces the zigzags of the iterates but significantly increases the number of iterations.
- An excessively large step makes the method diverge.

Particular Case: Quadratic Functions

In this paragraph we assume that $f : \mathbb{R}^n \to \mathbb{R}$ is given by

$$f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle + c,$$

where $A \in M_n(\mathbb{R})$ is a symmetric positive definite (SPD) matrix, $b \in \mathbb{R}^n$, and $c \in \mathbb{R}$, so it is a quadratic form associated with an SPD matrix.

We must calculate $\rho_k \in \mathbb{R}$ which minimizes the function $g : \mathbb{R} \to \mathbb{R}$ given by

$$g(\rho) = f(x_k - \rho \nabla f(x_k)).$$

Then ρ_k necessarily satisfies

$$g'(\rho_k) = 0.$$

A simple calculation gives

$$g'(\rho) = -\langle \nabla f(x_k - \rho \nabla f(x_k)), \nabla f(x_k)\rangle.$$

That is, since $\nabla f(x_k) = Ax_k - b$,

$$g'(\rho) = -\|Ax_k - b\|^2 + \rho\langle A(Ax_k - b), Ax_k - b\rangle.$$

We thus obtain

$$\rho_k = \frac{\|Ax_k - b\|^2}{\langle A(Ax_k - b), Ax_k - b\rangle}.$$

Note that $\langle A(Ax_k - b), Ax_k - b\rangle > 0$ (because A is SPD and $Ax_k - b = \nabla f(x_k) \neq 0$). Therefore, the optimal step gradient method in the quadratic case is:

$$x_{k+1} = x_k - \rho_k(Ax_k - b),$$

with ρ_k given by (3.6), valid only for $Ax_k - b \neq 0$.

Example

Let $f(X)$ be a quadratic cost function in $\mathbb{R}^2$ given by:

$$f(X) = x^2 + 3y^2,$$

with

$$X = \begin{pmatrix} x \\ y \end{pmatrix}.$$

It is evident that the origin is the point minimizing this function. This function is illustrated in Figures .... and ....

Resolution with a fixed step

In this case, we search for an optimal step ρ which minimizes a certain function. We have:

$$\nabla f(X) = [2x, 6y],$$

$$\xi(\rho) = X - \rho\nabla f(X)^T = \begin{pmatrix} (1-2\rho)x \\ (1-6\rho)y \end{pmatrix}. \tag{2.7}$$

To obtain the parameter ρ, we consider the function $g(\rho)$ that we must minimize with respect to ρ:

$$g(\rho) = f(\xi(\rho)) = (1-2\rho)^2x^2 + (1-6\rho)^2y^2.$$

The derivative of this function with respect to the variable ρ gives:

$$g'(\rho) = -4(1-2\rho)x^2 - 36(1-6\rho)y^2.$$

The parameter ρ that minimizes the function $g(\rho)$ is the one satisfying $g'(\rho) = 0$, thus:

$$\rho = \frac{x^2 + 9y^2}{2x^2 + 54y^2}.$$

Therefore, the iterative equation to solve this problem is:

- Initialization:
$$X = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}.$$
- Repeat:
  1. $\Delta X \leftarrow -\begin{pmatrix} 2x \\ 6y \end{pmatrix}$.
  2. $\rho = \frac{x^2+9y^2}{2x^2+54y^2}$.
  3. $X \leftarrow X + \rho \Delta X$.
- Stop when $\|\Delta X\|_2^2 \leq 10^{-3}$.

Resolution with an iterative step

When we choose an iterative step, the second step in the algorithm is replaced by the following loop:

Let $\alpha = 0.25$, $\beta = 0.5$, $\rho = 1$, then:

For a given direction
$$\Delta X = -\begin{pmatrix} 2X(1) \\ 6X(2) \end{pmatrix},$$
do:

1. $Y = X + \rho \Delta X$,
2. $f(Y) = Y(1)^2 + 3Y(2)^2$,
3. $\nabla f(X) = [2X(1), 6X(2)]$,
4. if $f(Y) < f(X) + \alpha \rho \nabla f(X) \Delta X$, end,
5. else $\rho = \beta \rho$,
6. end.

3.1.3 Newton's Method

The Newton algorithm in optimization is a direct application of Newton's method for solving equations of the form $J(x) = 0$. In unconstrained optimization, the Newton algorithm seeks solutions of the equation $J(x) = 0$. This is a nonlinear equation (or rather a system of nonlinear equations) in $\mathbb{R}^n$, and we will use Newton's method to solve it. However, we will only obtain the critical points of J: it is then necessary to check that they are indeed minima.

Here $f = J$ is indeed a function from $\mathbb{R}^n$ to $\mathbb{R}^n$. The derivative of f is none other than the Hessian matrix of J: $H(x) = D^2 J(x)$.

The Newton method is thus written as:

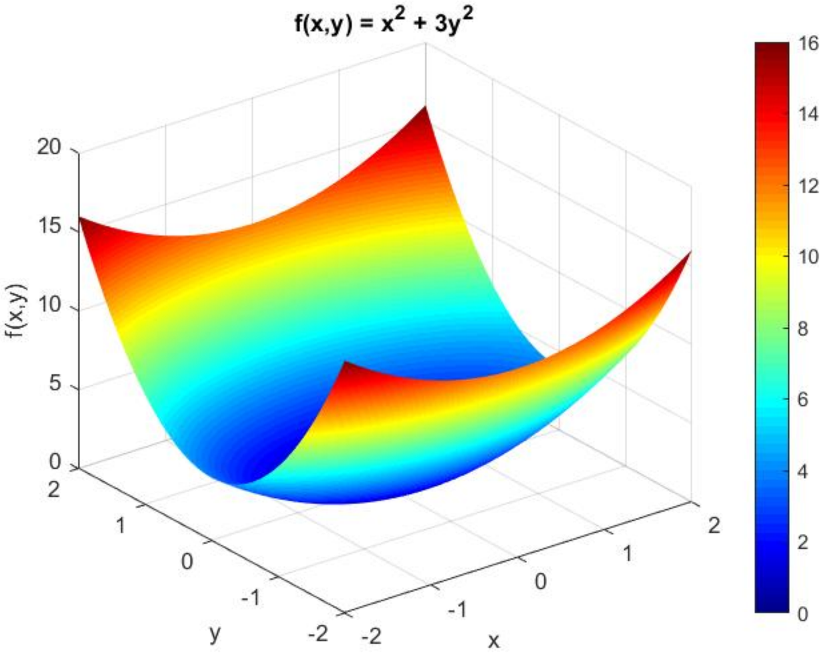


Figure 3.3: Shape of the function $f(x, y) = x^2 + 3y^2$.

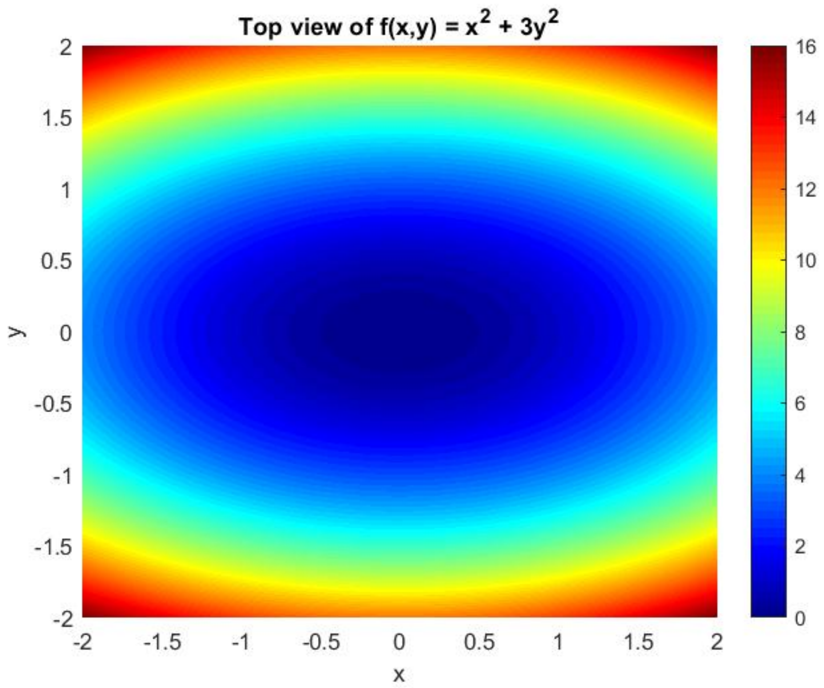


Figure 3.4: Top view of the function $f(x, y) = x^2 + 3y^2$.

Algorithm 4 Newton's Algorithm in $\mathbb{R}^n$

1: **Initialization:** $k = 0$; choose $x_0 \in \mathbb{R}^n$ in a neighborhood of x.
2: **Iteration** k**:**

$$x_{k+1} = x_k - [H(x_k)]^{-1} J(x_k).$$

3: **Stopping criterion:** if $\|x_{k+1} - x_k\| < \varepsilon$, STOP; otherwise, set $k = k + 1$ and return to step 2.

Step 2 of the method amounts to solving the following linear system:

$$H_k \Delta_k = J(x_k)$$

where $H_k = H(x_k)$, then setting

$$x_{k+1} = x_k - \Delta_k.$$

Newton's Method: Advanced 3D Example

Problem. Minimize:

$$f(x, y, z) = x^2 + xy + y^2 + yz + z^2 + xz - 6x - 9y - 15z.$$

Gradient.

$$\nabla f(x, y, z) = \begin{bmatrix} 2x + y + z - 6 \\ x + 2y + z - 9 \\ x + y + 2z - 15 \end{bmatrix}.$$

Hessian.

$$H(x, y, z) = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}.$$

Inverse Hessian.

$$H^{-1} = \frac{1}{4} \begin{bmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix}.$$

First iteration with $x_0 = [0, 0, 0]^T$.

$$\nabla f(x_0) = \begin{bmatrix} -6 \\ -9 \\ -15 \end{bmatrix}.$$

$$H^{-1} \nabla f(x_0) = \frac{1}{4} \begin{bmatrix} 6 \\ -6 \\ -30 \end{bmatrix} = \begin{bmatrix} 1.5 \\ -1.5 \\ -7.5 \end{bmatrix}.$$

$$x_1 = x_0 - H^{-1} \nabla f(x_0) = \begin{bmatrix} -1.5 \\ 1.5 \\ 7.5 \end{bmatrix}.$$

The algorithm converges in one iteration.
Minimum point: $(-1.5, 1.5, 7.5)$

Minimum value calculation.

$$f(-1.5, 1.5, 7.5) = -58.5.$$

Minimum value: $f(-1.5, 1.5, 7.5) = -58.5.$

Graphical Illustration

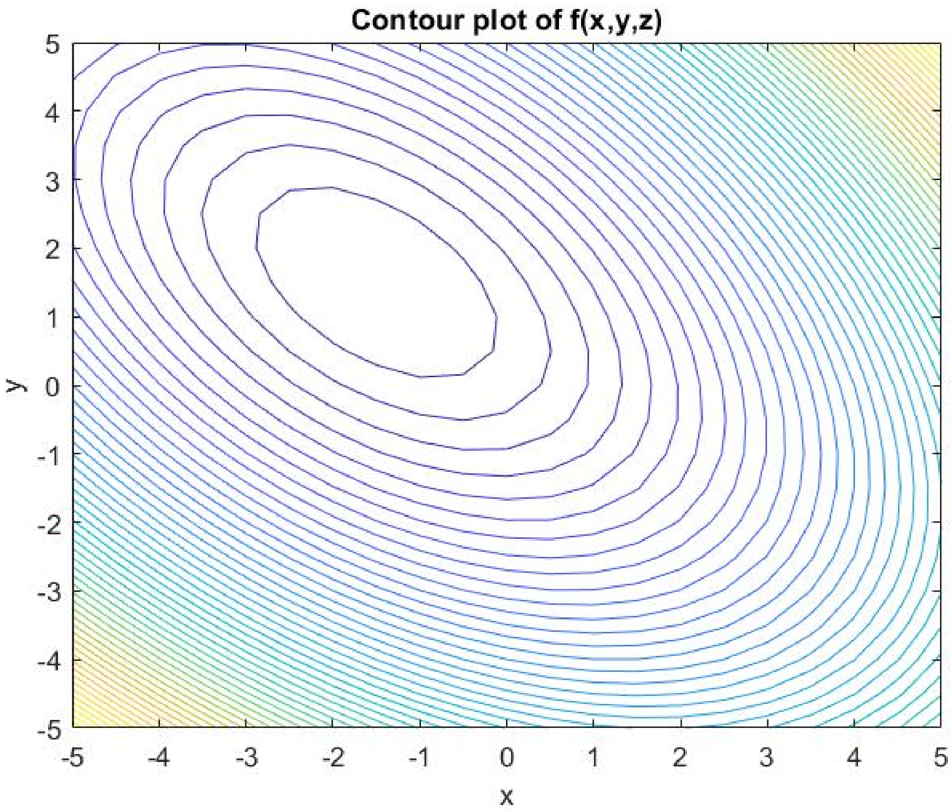


Figure 3.5: 3D surface plot of $f(x, y, z)$.

3.1.3.1 Disadvantages of Newton's Method

1. This method works very well for small dimensions ($1 \leq n \leq 10$) when it is easy to compute $H(x)$ and $H(x)^{-1}$. However, this calculation requires more numerous and costly iterations for large-scale problems.

2. Since $x_{k+1} = x_k - H(x_k)^{-1}\nabla f(x_k)$, the point x_{k+1} is not always well-defined, i.e., it is possible that $H(x_k)^{-1}$ does not exist (this typically occurs when the method reaches a region where f is linear, and thus its second partial derivatives are zero). Even if it exists, the direction $d_k = -H(x_k)^{-1}\nabla f(x_k)$ is not always a descent direction (if $H(x_k)$ is positive definite, then d_k is a descent direction).

3. The major disadvantage of the method is its sensitivity to the choice of the starting point x_0: if this point is poorly chosen (too far from the solution), the method can either diverge or converge to another solution. To choose the starting point x_0 sufficiently close to x^*, one tries to approach x^* by using a gradient-type method, and then applying Newton's method.

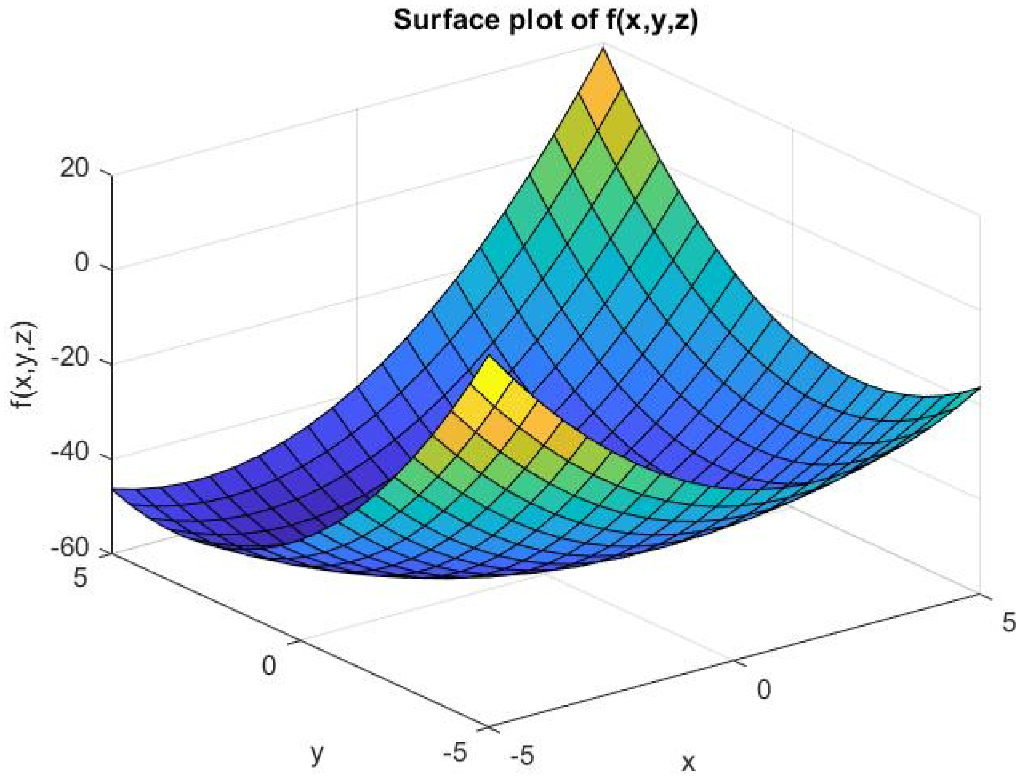


Figure 3.6: Contour (level) plot of $f(x, y, z)$ showing the minimum.

3.1.3.2 Quasi-Newton Method

Quasi-Newton methods are developed for optimization to overcome the drawbacks of Newton's method:

1. They maintain the speed of Newton's method.
2. They avoid the (costly) computation of the matrix $[H(x_k)]$ at each iteration.
3. They are more robust with respect to the starting point. There are methods called "trust region" methods that aim to make the method robust (i.e., not very sensitive) to x_0.
4. $(H(x))^{-1}$ is not necessarily known, it can be very expensive to compute, and $H(x_k)$ can be very difficult to invert. Thus, $(H(x_k))^{-1}$ is replaced by a matrix D_k, possibly constant, which is supposed to approximate $H(x_k)$ or its inverse. Sometimes, even $(H(x_k))^{-1}\nabla f(x_k)$ is replaced by an easily computable vector y_k.

Therefore, the algorithm of this method is given as follows:

Algorithm 5 Quasi-Newton Algorithm

1: **Initialization:** $k = 0$; choose x_0, α_0, ϵ.
2: **Iteration** k**:**

$$x_{k+1} = x_k - \alpha_k D_k \nabla f(x_k).$$

3: **Stopping criterion:** if $\|x_{k+1} - x_k\| < \epsilon$, STOP; otherwise, set $k = k + 1$ and return to step 2.

Remark 3.0.3. In this algorithm, we use an approximation D_k of $(H(x_k))^{-1}$ and then find α_k (by a line search) which minimizes the function $\varphi(\alpha) = f(x_k + \alpha_k d_k)$.

Problem

We consider the following difficult optimization problem in $\mathbb{R}^2$:

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 + e^{x-y}.$$

We apply the **BFGS Quasi-Newton method** starting from $x_0 = [1; 2]$ with identity as initial inverse Hessian approximation.

Detailed Solution

Let $x_k = [x_k; y_k]$ at iteration k.

- **Gradient:**

$$\nabla f(x, y) = \begin{bmatrix} 4x(x^2 + y - 11) + 2(x + y^2 - 7) + e^{x-y} \\ 2(x^2 + y - 11) + 4y(x + y^2 - 7) - e^{x-y} \end{bmatrix}.$$

- **Update:**

$$x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k),$$

  where H_k is the inverse Hessian approximation, updated via BFGS formula.

Algorithm

Algorithm 6 Quasi-Newton (BFGS) Algorithm

1: **Initialization:** $k = 0$; choose $x_0 \in \mathbb{R}^n$, set $H_0 = I$, tolerance $\epsilon = 10^{-5}$.
2: **Iteration k:**
3: Compute the descent direction:

$$d_k = -H_k \nabla f(x_k).$$

4: Perform line search to find α_k that minimizes $f(x_k + \alpha_k d_k)$.
5: Update:

$$x_{k+1} = x_k + \alpha_k d_k.$$

6: Compute:

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k).$$

7: Update the inverse Hessian approximation H_k using the BFGS formula:

$$H_{k+1} = \left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) H_k \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) + \frac{s_k s_k^T}{y_k^T s_k}.$$

8: **Stopping criterion:** If $\|\nabla f(x_{k+1})\| < \epsilon$, STOP; otherwise, set $k = k + 1$ and return to step 2.

MATLAB Code: Quasi-Newton BFGS Advanced

Below is the MATLAB program implementing the Quasi-Newton BFGS method for:

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 + e^{x-y}.$$

Program:

```
function quasi_newton_bfgs_advanced()
    % Function definition
    f = @(x,y) (x.^2 + y - 11).^2 + (x + y.^2 - 7).^2 + exp(x - y);

    % Gradient
    grad = @(x) [
        4*x(1)*(x(1)^2 + x(2) - 11) + 2*(x(1) + x(2)^2 - 7) + exp(x(1)-x
            (2));
        2*(x(1)^2 + x(2) - 11) + 4*x(2)*(x(1) + x(2)^2 - 7) - exp(x(1)-x
            (2))
    ];

    % Initialization
    xk = [1; 2];
    Hk = eye(2);
    epsilon = 1e-5;
```

```
    maxIter = 1000;
    path = xk';

    for k = 1:maxIter
        gk = grad(xk);
        if norm(gk) < epsilon
            break;
        end

        dk = -Hk * gk;

        % Backtracking line search
        alpha = 1;
        rho = 0.5;
        c = 1e-4;
        while f(xk(1)+alpha*dk(1), xk(2)+alpha*dk(2)) > ...
              f(xk(1),xk(2)) + c*alpha*gk'*dk
            alpha = rho * alpha;
        end

        % Update
        xk_new = xk + alpha * dk;
        sk = xk_new - xk;
        yk = grad(xk_new) - gk;

        % BFGS update formula
        Hk = (eye(2) - sk*yk'/(yk'*sk)) * Hk * ...
             (eye(2) - yk*sk'/(yk'*sk)) + (sk*sk')/(yk'*sk);
        xk = xk_new;
        path = [path; xk'];
    end

    % Plotting
    [X,Y] = meshgrid(-5:0.05:5, -5:0.05:5);
    Z = (X.^2 + Y - 11).^2 + (X + Y.^2 - 7).^2 + exp(X - Y);

    figure;
    contour(X,Y,Z,logspace(0,5,40)); hold on;
    plot(path(:,1), path(:,2), 'r.-', 'LineWidth', 1.5);
    xlabel('x'); ylabel('y');
    title('BFGS Optimization Path on Modified Himmelblau + Exponential');
    colorbar;

    % Save as PNG
    saveas(gcf, 'bfgs_himmelblau.png');
end
```

Listing 3.2: BFGS Optimization on Modified Himmelblau + Exponential

Graphical Illustration

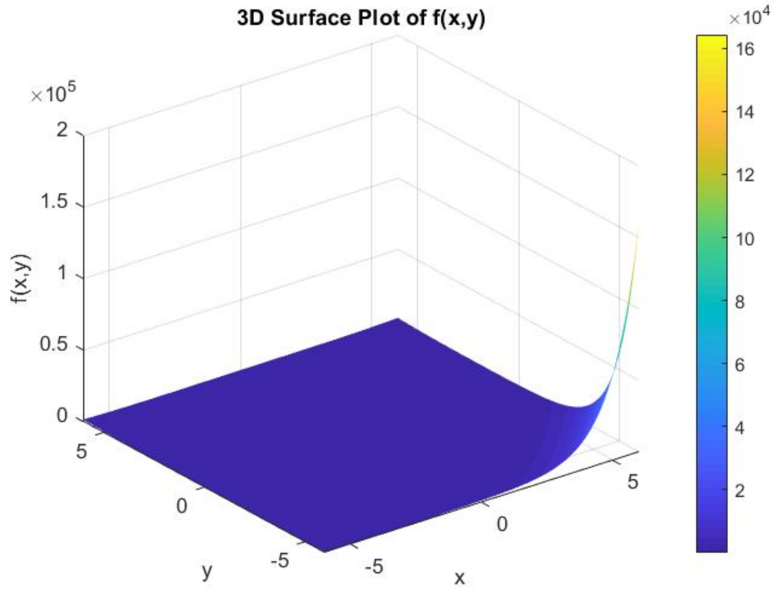


Figure 3.7: Optimization path using BFGS Quasi-Newton on the modified Himmelblau + exponential function.

3.1.4 Conjugate Gradient Method

Conjugate gradient methods are used to solve linear systems whose matrix is symmetric positive definite. They are also used to solve large linear systems. It is an iterative method that converges in a finite number of iterations (at most equal to the dimension of the linear system).

Definition 3.1.1. *Let A be an $n \times n$ symmetric positive definite matrix. Two vectors x and y in $\mathbb{R}^n$ are said to be A-conjugate (or conjugate with respect to A) if they satisfy:*

$$x^T Ay = 0.$$

Principle of Conjugate Gradient Method

Let $\{d_0, d_1, ..., d_n\}$ be a family of A-conjugate vectors. We then call *conjugate direction method* any iterative method applied to a strictly convex quadratic function of n variables:

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

with $x \in \mathbb{R}^n$, $A \in \mathbb{M}_{n\times n}$ symmetric positive definite, $b \in \mathbb{R}^n$, and $c \in \mathbb{R}$, leading to the optimum in at most n steps.

They rely on the concept of conjugate directions because successive gradients are orthogonal to each other and to previous directions.

The idea of the method is, given an initial point x_0 in $\mathbb{R}^n$ and n conjugate directions, to iteratively construct mutually conjugate directions $d_1, ..., d_k$. We define the following scheme:

$$x_{k+1} = x_k + \rho_k d_k,$$

where ρ_k is the scalar minimizing $f(x)$ along the direction $x_k + \rho_k d_k$ and is defined by:

$$\rho_k = -\frac{r_k^T d_k}{d_k^T A d_k},$$

$$r_k = -\nabla f(x_k) = b - Ax_k.$$

The algorithm below solves $Ax = b$, where A is a real, symmetric, positive definite matrix. The input vector x_0 can be an approximation of the initial solution or zero.

Algorithm 7 Conjugate Gradient Method

1: **Step 0 (Initialization):** Set $k = 0$.
2: Choose $x_0 \in \mathbb{R}^n$; compute $r_0 = -\nabla f(x_0) = b - Ax_0$, set $p_0 = r_0$.
3: **Step 1 (Compute step size):**

$$\alpha_k = -\frac{r_k^T r_k}{p_k^T A p_k}.$$

4: **Step 2 (Update):**

$$x_{k+1} = x_k + \rho_k p_k, \quad r_{k+1} = r_k - \alpha_k A p_k.$$

5: **Step 3 (Stopping criterion):** If r_{k+1} is sufficiently small, STOP the algorithm.
6: **Step 4 (Direction update):**

$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}, \quad p_{k+1} = r_{k+1} + \beta_k p_k.$$

7: Replace k by $k + 1$ and return to Step 1.

Theorem 3.1.3. *If $f : \mathbb{R}^n \to \mathbb{R}$ is quadratic and elliptic, the conjugate gradient method converges in at most n iterations, where n is the order of A.*

Example: Conjugate Gradient Method (Detailed)

Problem. Minimize:

$$f(x, y) = 4x^2 + xy + 3y^2 - 8x - 9y.$$

This can be written in quadratic form as:

$$f(X) = \frac{1}{2}X^T AX - b^T X$$

where

$$A = \begin{bmatrix} 8 & 1 \\ 1 & 6 \end{bmatrix}, \quad b = \begin{bmatrix} 8 \\ 9 \end{bmatrix}.$$

Set $x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.
Compute:

$$r_0 = b - Ax_0 = b = \begin{bmatrix} 8 \\ 9 \end{bmatrix}, \quad p_0 = r_0.$$

Compute

$$\alpha_0 = \frac{r_0^T r_0}{p_0^T A p_0}.$$

Calculate:

$$r_0^T r_0 = 8^2 + 9^2 = 64 + 81 = 145.$$

$$Ap_0 = \begin{bmatrix} 8 & 1 \\ 1 & 6 \end{bmatrix} \begin{bmatrix} 8 \\ 9 \end{bmatrix} = \begin{bmatrix} 73 \\ 62 \end{bmatrix}.$$

Then,

$$p_0^T A p_0 = [8, 9] \cdot [73, 62] = 8 * 73 + 9 * 62 = 584 + 558 = 1142.$$

Hence,

$$\alpha_0 = \frac{145}{1142} \approx 0.127.$$

$$x_1 = x_0 + \alpha_0 p_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 0.127 \begin{bmatrix} 8 \\ 9 \end{bmatrix} = \begin{bmatrix} 1.016 \\ 1.143 \end{bmatrix}.$$

$$r_1 = r_0 - \alpha_0 A p_0 = \begin{bmatrix} 8 \\ 9 \end{bmatrix} - 0.127 \begin{bmatrix} 73 \\ 62 \end{bmatrix} = \begin{bmatrix} -1.271 \\ 1.126 \end{bmatrix}.$$

$$\beta_0 = \frac{r_1^T r_1}{r_0^T r_0}.$$

Calculate:

$$r_1^T r_1 = (-1.271)^2 + (1.126)^2 = 1.615 + 1.268 = 2.883.$$

Hence,

$$\beta_0 = \frac{2.883}{145} \approx 0.0199.$$

$$p_1 = r_1 + \beta_0 p_0 = \begin{bmatrix} -1.271 \\ 1.126 \end{bmatrix} + 0.0199 \begin{bmatrix} 8 \\ 9 \end{bmatrix} = \begin{bmatrix} -1.112 \\ 1.305 \end{bmatrix}.$$

Continue with the same process:

$$\alpha_1 = \frac{r_1^T r_1}{p_1^T A p_1}, \quad x_2 = x_1 + \alpha_1 p_1, \quad r_2 = r_1 - \alpha_1 A p_1.$$

The method converges in at most $n = 2$ iterations for a quadratic problem.

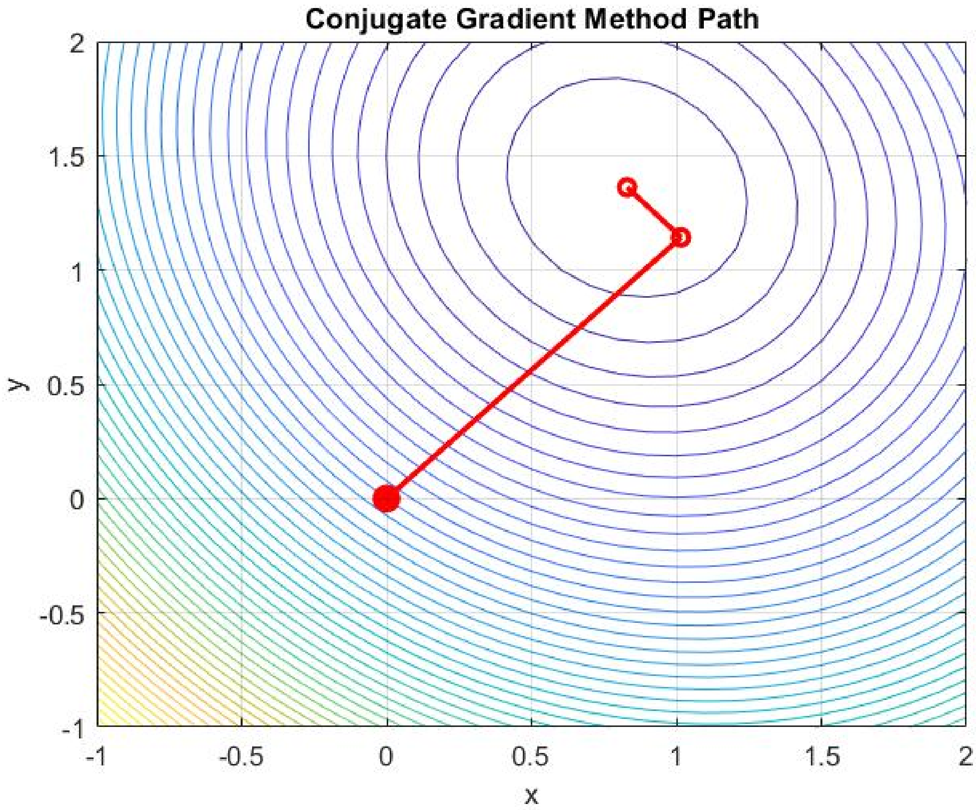


Figure 3.8: Conjugate Gradient method optimization path for $f(x, y) = 4x^2 + xy + 3y^2 - 8x - 9y$.

MATLAB Implementation: Conjugate Gradient Method Example

Problem

Solve: $f(x, y) = 4x^2 + xy + 3y^2 - 8x - 9y$ using the Conjugate Gradient Method.

MATLAB Code

```
% Conjugate Gradient Method Example

% Define A and b
A = [8 1; 1 6];
b = [8; 9];

% Initial guess
x = [0; 0];

% Compute initial residual
r = b - A*x;
p = r;
k = 0;

while norm(r) > 1e-6
    Ap = A*p;
    alpha = (r'*r) / (p'*Ap);
    x_new = x + alpha*p;
    r_new = r - alpha*Ap;

    if norm(r_new) < 1e-6
        break;
    end

    beta = (r_new'*r_new) / (r'*r);
    p = r_new + beta*p;

    % Update for next iteration
    x = x_new;
    r = r_new;
    k = k + 1;

    % Display each iteration
    fprintf('Iteration %d: x = [%f; %f], residual norm = %e\n', ...
            k, x(1), x(2), norm(r));
end

% Final solution
disp('Optimal x:')
disp(x)
```

Expected Output

Optimal solution $x^* \approx [0.967, 1.401]$ minimizing the quadratic function.

3.1.5 Relaxation Method

The last method we present allows reducing a minimization problem in $\mathbb{R}^n$ to the successive resolution of n minimization problems in $\mathbb{R}$ (at each iteration).

We seek to minimize $J : \mathbb{R}^n \to \mathbb{R}$; let us set

$$X = (x_1, x_2, \ldots, x_n).$$

The principle of the method is as follows:

Given an iterate X^k with coordinates $(x_1^k, x_2^k, \dots, x_n^k)$, we fix all components except the first and we minimize over the first:

$$\min_{x \in \mathbb{R}} J(x, x_2^k, x_3^k, \dots, x_n^k).$$

Thus, we obtain the first coordinate of the next iterate X^{k+1}, which we denote x_1^{k+1}. For this minimization in $\mathbb{R}$, one can use for example Newton's method in $\mathbb{R}$.

Then, we repeat by fixing the first coordinate to x_1^{k+1} and the last $n-2$ as before. We minimize over the second coordinate, and so on.

The resulting algorithm is as follows:

Algorithm 8 Successive Relaxation Method – One Iteration

1: **Step 0 (Initialization):** At the beginning of iteration k, choose $X^k \in \mathbb{R}^n$.
2: **Step 1 (Coordinate-wise minimization):**
3: For each coordinate $i = 1$ to n:
4: Minimize J with respect to x_i while keeping all other components fixed:

$$x_i^{k+1} = \min_x J(x_1^{k+1}, x_2^{k+1}, \dots, x_{i-1}^{k+1}, x, x_{i+1}^k, \dots, x_n^k).$$

5: **Step 2 (Stopping test):** If $\|X^{k+1} - X^k\| < \epsilon$, STOP; otherwise, set $k = k+1$ and return to Step 1.

Example: Successive Relaxation Method

Minimize:

$$f(x, y, z) = x^4 + y^4 + z^4 - 4xy + z^2 - x + y + 3.$$

Step-by-step solution:

1. Initialize: $x_0 = 0$, $y_0 = 0$, $z_0 = 0$.
2. Minimize with respect to x (fixing y, z):
$$x_{k+1} = x_k - \frac{4x_k^3 - 4y_k - 1}{12x_k^2}.$$
3. Minimize with respect to y (fixing x, z):
$$y_{k+1} = y_k - \frac{4y_k^3 - 4x_k + 1}{12y_k^2}.$$
4. Minimize with respect to z:
$$z_{k+1} = 0.$$
5. Check convergence:
$$\sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2 + (z_{k+1} - z_k)^2} < \epsilon.$$

Algorithm 9 Successive Relaxation for 3-variables

Initialization: Choose x_0, y_0, z_0, tolerance ϵ, set $k = 0$.
repeat
  Update $x_{k+1} = \frac{2-y_k-z_k}{2}$.
  Update $y_{k+1} = \frac{-x_{k+1}-z_k-3}{8}$.
  Update $z_{k+1} = \frac{1-x_{k+1}-y_{k+1}}{18}$.
  $k = k + 1$.
until $|x_{k+1} - x_k| + |y_{k+1} - y_k| + |z_{k+1} - z_k| < \epsilon$
Output: Optimal (x^*, y^*, z^*).

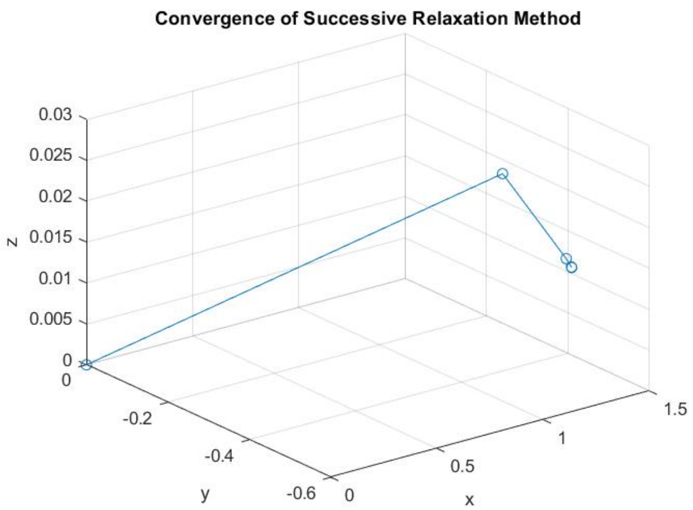


Figure 3.9: Convergence trajectory of Successive Relaxation in 3D space.

Sheet 03

Gradient Method

Exercise 1. Prove that for a strictly convex quadratic function $f(x) = \frac{1}{2}x^T Ax - b^T x + c$, with A symmetric positive definite, the gradient method converges for step size $\rho \in (0, \frac{2}{M})$ where M is the largest eigenvalue of A.

Exercise 2. Show that if ρ is too large in the gradient method, the sequence $\{x_k\}$ can diverge. Illustrate this with the proof for a general quadratic function.

Exercise 3. Minimize $f(x, y) = 3x^2 + 2xy + 2y^2 - 4x - 6y$ starting at $x_0 = (0, 0)^T$ using gradient descent with fixed step size $\rho = 0.1$. Perform three iterations and compute x_1, x_2, x_3.

Exercise 4). For $f(x, y) = x^2 + 4y^2$, apply the gradient method with optimal step size starting at $(2, 1)$. Compute analytically the optimal ρ at each iteration and perform two iterations.

Conjugate Gradient Method

Exercise 5. Prove that for a quadratic function with a symmetric positive definite matrix A, the conjugate gradient method converges in at most n iterations.

Exercise 6. Explain why conjugate directions are A-orthogonal and deduce the optimality condition from this property.

Exercise 7. Solve the linear system $Ax = b$ using conjugate gradient with $A = \begin{pmatrix} 4 & 1 \\ 1 & 3 \end{pmatrix}$, $b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, starting from $x_0 = 0$. Perform two iterations showing all calculations.

Exercise 8. Minimize $f(x, y) = 5x^2 + 6xy + 10y^2 - 2x - 3y$ using conjugate gradient from $x_0 = (0, 0)$. Do two iterations.

Newton Method

Exercise 9. Prove the quadratic convergence of Newton's method under the condition that $f''(x)$ is Lipschitz continuous near the solution.

Exercise 10. Discuss the disadvantages of Newton's method when $H(x)$ is not invertible or not positive definite.

Exercise 11. Apply Newton's method to $f(x, y) = x^2 + xy + y^2 - x - y$ starting at $(0, 0)$. Compute two iterations with explicit gradient and Hessian.

Exercise 12. Use Newton's method to find the minimizer of $f(x, y) = x^4 + y^4 - 4xy + 1$ starting at $(1, 1)$. Perform one iteration.

Relaxation Method

Exercise 13. Explain the principle of the relaxation method and its convergence conditions for strictly convex separable functions.

Exercise 14. Prove that if the minimization at each coordinate is exact and the function is strictly convex, the relaxation method converges to the global minimum.

Exercise 15. Minimize $f(x, y) = x^2 + 3y^2 + xy$ using the relaxation method, alternating minimization over x and y, starting at $(1, 1)$. Perform two full cycles.

Exercise 16. Apply the relaxation method to minimize $f(x, y, z) = x^2 + y^2 + z^2 + xy + yz + xz$ starting at $(1, 1, 1)$ and perform one full cycle.

Sheet 03 (Solutions)

Gradient Descent

Exercise 1 (Numerical)

Problem. Minimize $f(x, y) = x^2 + y^2$ from $(1, 1)$ with $\alpha = 0.1$.

Solution.

- Gradient: $\nabla f = [2x, 2y]$.
- Iteration 1:
$$x_1 = 1 - 0.1 \times 2 = 0.8, \quad y_1 = 1 - 0.1 \times 2 = 0.8.$$
- Iteration 2:
$$x_2 = 0.8 - 0.1 \times 1.6 = 0.64, \quad y_2 = 0.8 - 0.1 \times 1.6 = 0.64.$$
- Iteration 3:
$$x_3 = 0.64 - 0.1 \times 1.28 = 0.512, \quad y_3 = 0.64 - 0.1 \times 1.28 = 0.512.$$
- Converges geometrically to $(0, 0)$.

—

Exercise 2 (Theoretical)

Problem. Prove convergence if α is small enough for convex f.

Solution. If f has L-Lipschitz gradient, then for $\alpha \in (0, 2/L)$:

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

thus $f(x_k)$ decreases and converges.

—

Conjugate Gradient

Exercise 3 (Numerical)

Problem. Solve $Ax = b$ with $A = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, x_0 = 0$.

Solution.

- $r_0 = b - Ax_0 = b = [1; 2], p_0 = r_0$.
- $\alpha_0 = \frac{r_0^T r_0}{p_0^T A p_0} = \frac{5}{9} \approx 0.5556$.
- $x_1 = x_0 + \alpha_0 p_0 = [0.5556; 1.1112]$.
- $r_1 = r_0 - \alpha_0 A p_0 = [-1.2224; 0.3332]$.
- Continue iterations until convergence.

—

Exercise 4 (Theoretical)

Problem. Prove CG converges in at most n steps for positive definite A.

Solution. CG builds a Krylov subspace of dimension n, ensuring the exact solution is reached in at most n steps under exact arithmetic.

—

Newton's Method

Exercise 5 (Numerical)

Problem. Minimize $f(x) = x^4 - 3x^3 + 2$.

Solution.

- $\nabla f = 4x^3 - 9x^2$, $\nabla^2 f = 12x^2 - 18x$.
- Starting from $x_0 = 0.5$:
$$x_1 = x_0 - \frac{\nabla f(x_0)}{\nabla^2 f(x_0)}.$$
- Compute $\nabla f(0.5) = 0.5 - 2.25 = -1.75$, $\nabla^2 f(0.5) = 3 - 9 = -6$.
- Thus $x_1 = 0.5 - (-1.75)/(-6) = 0.5 - 0.2917 = 0.2083$.

—

Exercise 6 (Theoretical)

Problem. Why Newton may diverge for poor initial guess?

Solution. If x_0 is far from local minima or near saddle points where $\nabla^2 f$ is indefinite, steps may lead away from minima.

—

Relaxation Method

Exercise 7 (Numerical)

Problem. Minimize $f(x, y) = x^2 + 3y^2$ using relaxation.

Solution.

- Fix $y = y_k$, minimize in x: $x_{k+1} = 0$.
- Fix $x = 0$, minimize in y: $y_{k+1} = 0$.
- Converges in one sweep.

—

Exercise 8 (Theoretical)

Problem. Why relaxation may be slow in coupled problems?

Solution. Because updating one variable at a time ignores coupling structure, leading to zig-zagging and slow convergence (Gauss-Seidel type issues).

Sheet of Practical Work

Instructions

Instructions for Each TP

For each TP:

- Implement the method in MATLAB.
- Test on the proposed problem with detailed comments.
- Analyze convergence behaviour, iteration count, and solution accuracy.
- Answer the reflective questions at the end of each TP.

Gradient Descent Method

TP 1. Rosenbrock's Banana Function

Implement gradient descent to minimize:

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2.$$

Tasks:

1. Derive the gradient analytically.
2. Implement in MATLAB with step size $\alpha = 0.001$ and stopping criterion $\|\nabla f(x_k)\| < 10^{-6}$.
3. Test from starting point $(x_0, y_0) = (-1.2, 1)$.
4. Plot convergence trajectory on the contour plot of f.

Questions:

- Does the method converge? If not, why?
- Test different step sizes. Report the impact on convergence.

TP 2. N-dimensional Quadratic Function

Minimize:

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

where A is SPD of size $n = 10$, generated as $A = Q^T DQ$ with D diagonal positive, Q orthogonal.

Tasks:

1. Generate A and b randomly in MATLAB.

2. Implement gradient descent with fixed step $\alpha = 1/\lambda_{\max}(A)$.

3. Compare with MATLAB's `quadprog` result.

Questions:

- What is the theoretical optimal step for this problem?
- Compare iteration count for α too small vs optimal.

Conjugate Gradient Method

TP 1. Solving SPD System

Solve $Ax = b$ where A is SPD with $n = 1000$, b random.

Tasks:

1. Implement Conjugate Gradient method without preconditioning.

2. Compare solution time and residual with MATLAB's `pcg`.

3. Plot convergence (residual norm vs iteration).

Questions:

- Why is CG preferred over direct methods for large sparse SPD systems?

TP 2. Quadratic Minimization Interpretation

Minimize:

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

using CG and compare to gradient descent implemented previously (TP I.2).

Tasks:

1. Use the same A, b as TP I.2.

2. Compare iteration count and execution time.

3. Plot both convergence curves on the same graph.

Questions:

- Explain mathematically why CG converges in at most n steps.

Newton's Method

TP 1. Himmelblau's Function

Minimize:

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2.$$

Tasks:

1. Derive gradient and Hessian analytically.
2. Implement Newton's method with stopping criterion $\|\nabla f(x_k)\| < 10^{-6}$.
3. Test starting from $(0, 0)$, $(3, 2)$, $(-3, -3)$ and analyze the solutions obtained.
4. Plot iterations on 3D surface and contour plots.

Questions:

- Why does Newton's method fail or succeed depending on the starting point?

TP 2. High-dimensional Newton

Consider:

$$f(x) = \sum_{i=1}^{n} (x_i - i)^4$$

with $n = 20$.

Tasks:

1. Implement Newton's method (use diagonal Hessian).
2. Compare with gradient descent performance.

Questions:

- Discuss computational cost for Hessian inversion vs gradient descent.

Relaxation Method

TP 1. Coordinate-wise minimization

Minimize:

$$f(x, y) = x^2 + xy + y^2 + 3x + 4y + 5$$

using relaxation (coordinate descent).

Tasks:

1. Implement sequential minimization along x and y.
2. Compare with MATLAB's `fminunc`.

Questions:

- When is coordinate descent effective compared to gradient methods?

TP 2. Large-scale LASSO (Bonus)

Using MATLAB's sparse optimization toolbox, implement coordinate descent for LASSO:

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1$$

where A is 1000×500 random Gaussian, b random, and $\lambda = 0.1$.

Tasks:

1. Implement coordinate-wise updates with soft-thresholding.
2. Compare solution with MATLAB's `lasso`.

Questions:

- Why is relaxation well-suited for LASSO problems?

3.2 Practical Work

3.2.1 Conjugate Gradient Method – TP1

Exercise 1 (Numerical)

Problem. Solve the linear system $Ax = b$ using the conjugate gradient method where

$$A = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

—

Implementation (MATLAB).

```
% Conjugate Gradient Method Example
A = [4 1; 1 3];
b = [1; 2];
x = zeros(2,1);
r = b - A*x;
p = r;
for k=1:2
    alpha = (r'*r)/(p'*A*p);
    x = x + alpha*p;
    r_new = r - alpha*A*p;
    if norm(r_new) < 1e-6
        break;
    end
    beta = (r_new'*r_new)/(r'*r);
    p = r_new + beta*p;
    r = r_new;
end
disp('Solution:');
disp(x);
```

—

Interpretation.

The algorithm converges in at most n steps (here $n = 2$). The final solution is approximately $[0.0909; 0.6364]$.

—

Exercise 2 (Theoretical)

Problem. Prove that for a symmetric positive definite matrix A, the conjugate gradient method converges in at most n steps.

Solution (Sketch).

- The method generates an A-orthogonal basis. - Each iteration minimizes the quadratic form along a conjugate direction. - Since $\mathbb{R}^n$ is of dimension n, the solution is reached in at most n steps.

3.2.2 Newton's Method – TP2

Exercise 1 (Numerical)

Problem. Minimize $f(x, y) = x^2 + xy + y^2 - 3x - 3y$ using Newton's method.

—

Implementation (MATLAB).

```
% Newton's Method Example
syms x y
f = x^2 + x*y + y^2 - 3*x - 3*y;
grad = gradient(f, [x y]);
hess = hessian(f, [x y]);
xk = [0; 0];
epsilon = 1e-5;
while true
    g = double(subs(grad, [x y], xk'));
    H = double(subs(hess, [x y], xk'));
    pk = -H\g';
    xk1 = xk + pk;
    if norm(pk) < epsilon
        break;
    end
    xk = xk1;
end
disp('Minimum at:');
disp(xk);
```

—

Interpretation.
The method converges quadratically to the minimum point.

—

3.2.3 Relaxation Method – TP3

Exercise 1 (Numerical)

Problem. Solve for the minimum of $f(x, y) = (x-1)^2 + (y-2)^2$ using relaxation (coordinate descent).

—

Implementation (MATLAB).

```
% Relaxation Method Example
x = 0; y = 0;
epsilon = 1e-6;
while true
    x_old = x;
    x = 1; % minimize w.r.t x keeping y fixed
    y = 2; % minimize w.r.t y keeping x fixed
    if sqrt((x-x_old)^2) < epsilon
        break;
    end
end
disp(['Minimum at x=',num2str(x),', y=',num2str(y)]);
```

—

Interpretation.
Coordinate descent converges to $(1, 2)$, the global minimum.

—

3.2.4 Gradient Method – TP4

Exercise 1 (Numerical)

Problem. Minimize $f(x, y) = x^2 + y^2 + xy$ using gradient descent with step size $\rho = 0.1$.

—

Implementation (MATLAB).

```
% Gradient Descent Example
xk = [0;0];
rho = 0.1;
epsilon = 1e-6;
while true
    grad = [2*xk(1)+xk(2); xk(1)+2*xk(2)];
    xk1 = xk - rho*grad;
    if norm(xk1 - xk) < epsilon
        break;
    end
    xk = xk1;
end
disp('Minimum at:');
disp(xk);
```

—

Interpretation.
The method converges linearly to the minimum point.
—

End of TP Sheet

Prepared for your Optimization Practical Sessions. Execute codes in MATLAB, generate figures, and interpret results clearly in your lab notebook.

General Conclusion

In this work, we have carried out a detailed study of unconstrained optimization problems, focusing on their theoretical foundations as well as practical algorithms for their resolution. We began by examining the concepts of **existence and uniqueness** of solutions. We established that if the objective function is continuous and coercive, then it admits at least one minimum in $\mathbb{R}^n$. Moreover, if the function is strictly convex, this minimum is unique, guaranteeing the reliability of numerical methods designed to solve such problems.

We then explored four major optimization methods. First, the **Gradient Method**, which is the most basic iterative method relying solely on first-order information to move in the direction of steepest descent. Although simple and widely applicable, we noted its slow convergence in the presence of ill-conditioned problems, where the function has narrow and elongated valleys.

Second, we studied the **Conjugate Gradient Method**, which improves upon the Gradient Method by constructing conjugate directions that accelerate convergence, especially for quadratic objective functions with symmetric positive definite matrices. This method is particularly effective for large-scale sparse systems, commonly encountered in engineering simulations and scientific computations.

Third, the **Newton's Method** was discussed, which leverages second-order derivative information to achieve quadratic convergence near the optimal solution. However, its practical implementation faces limitations due to the need to compute and invert the Hessian matrix, which becomes computationally expensive for high-dimensional problems. We also highlighted that Newton's Method is sensitive to the initial point, and it may fail if the Hessian is not positive definite at some iteration.

Finally, we addressed the **Relaxation Method**, also known as coordinate descent, which sequentially optimizes each variable while keeping the others fixed. This approach reduces the multidimensional problem to a series of univariate problems. While intuitive and sometimes effective for problems with separable structures, its convergence is generally slow and it may cycle or stall without reaching a true optimum in some non-separable cases.

Throughout this work, we implemented and analyzed these methods in detail, both theoretically and numerically. We solved practical examples step by step, coded the algorithms in MATLAB for experimentation, and illustrated their behaviour graphically to enhance understanding. This comprehensive approach allowed us to observe how each method behaves in different situations, highlighting their strengths and weaknesses.

Overall, the study of these methods provides essential tools for tackling unconstrained optimization problems arising in engineering design, data science, and applied mathematics. By understanding their mathematical principles, convergence conditions, and computational aspects, we are better equipped to select the most suitable method for a given problem or to develop hybrid and advanced optimization techniques adapted to large-scale or non-convex challenges.

3.3 Appendix A (Quiz)

Quiz – Gradient Method

Q1. Define the steepest descent direction.

Q2. Give the general update formula for gradient method.

Q3. Why is the gradient method slow for ill-conditioned problems?

Q4. Compute $\nabla f(x, y)$ for $f(x, y) = 3x^2 + 2xy + y^2$.

Q5. Compute the descent direction at point (1,2).

Q6. What is the stopping criterion in gradient descent?

Q7. True/False: The gradient is always perpendicular to the level curves. Justify.

Q8. Give a condition for ρ_k (step size) to ensure convergence.

Q9. If $f(x) = x^4$, is gradient descent efficient near $x = 0$? Explain.

Q10. Describe one advantage and one disadvantage of gradient descent.

Solutions – Gradient Method

A1. The direction $d_k = -\nabla f(x_k)$.

A2. $x_{k+1} = x_k - \rho_k \nabla f(x_k)$.

A3. Because the gradient oscillates in narrow valleys, slowing convergence.

A4. $\nabla f = [6x + 2y,\ 2x + 2y]^T$.

A5. At (1,2): $[6 * 1 + 2 * 2,\ 2 * 1 + 2 * 2] = [10, 6]$ so direction is [-10;-6].

A6. When $\|\nabla f(x_k)\| < \epsilon$.

A7. True. The gradient is normal to the level set at any point.

A8. ρ_k must satisfy Wolfe or Armijo conditions.

A9. No, because derivative is nearly flat near 0, leading to very small steps.

A10. Advantage: Simple to implement. Disadvantage: Slow convergence for ill-conditioned problems.

Quiz – Conjugate Gradient Method

Q1. Define A-conjugacy between two vectors.

Q2. State the main advantage of CG over steepest descent.

Q3. What is the convergence rate of CG for a quadratic problem?

Q4. Write the formula for α_k in CG.

Q5. Write the update for x_{k+1} in CG.

Q6. How is β_k computed in CG?

Q7. Why is CG preferred for large sparse systems?

Q8. Does CG require storage of the matrix inverse?

Q9. Explain why directions remain conjugate in CG.

Q10. State a limitation of CG.

Solutions – Conjugate Gradient Method

A1. $x^T A y = 0$.

A2. Faster convergence by using conjugate directions instead of orthogonal only.

A3. At most n iterations for an n-dimensional quadratic problem.

A4. $\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$.

A5. $x_{k+1} = x_k + \alpha_k p_k$.

A6. $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$.

A7. Because it does not require storage of full A, only matrix-vector products.

A8. No.

A9. Because each new direction is constructed to be A-conjugate to previous ones.

A10. Requires A to be symmetric positive definite.

Quiz – Newton's Method

Q1. Write the Newton update formula.

Q2. What is the role of the Hessian in Newton's method?

Q3. Why does Newton's method have quadratic convergence?

Q4. What is the drawback if the Hessian is not positive definite?

Q5. Compute $\nabla^2 f$ for $f(x, y) = x^2 + xy + y^2$.

Q6. Why is Newton's method sensitive to initial guess?

Q7. True/False: Newton's method always converges globally. Justify.

Q8. When is Newton's method preferred over gradient descent?

Q9. Does Newton require line search? Explain.

Q10. Give an alternative when Hessian is difficult to compute.

Solutions – Newton's Method

A1. $x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$.

A2. It determines the curvature and updates accordingly.

A3. Because it uses second order Taylor expansion.

A4. The direction may not be descent; can diverge.

A5. $\nabla^2 f = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$.

A6. Because it may converge to another critical point or diverge.

A7. False. Only local convergence is guaranteed near a minimizer.

A8. When high precision is needed and Hessian is easy to compute.

A9. Yes, to ensure global convergence.

A10. Quasi-Newton methods like BFGS.

Quiz – Relaxation Method

Q1. What is the principle of relaxation method?

Q2. How does it decompose an optimization problem?

Q3. What is updated at each iteration?

Q4. Give one advantage of relaxation method.

Q5. State a limitation of relaxation method.

Q6. True/False: Relaxation solves all variables simultaneously.

Q7. In which types of problems is relaxation effective?

Q8. Write the general update for variable x_i in relaxation.

Q9. Why is convergence not always guaranteed?

Q10. How can relaxation be combined with Newton's method?

Solutions – Relaxation Method

A1. Solves for one variable at a time, fixing others.

A2. By solving n one-dimensional minimizations successively.

A3. Each coordinate (variable) is updated one after another.

A4. Simplicity of implementation.

A5. Slow convergence or divergence for coupled variables.

A6. False.

A7. Problems with separable or weakly coupled variables.

A8. $x_i^{k+1} = \arg\min_x J(x_1^{k+1}, ..., x_{i-1}^{k+1}, x, x_{i+1}^k, ..., x_n^k)$.

A9. Because updates may not lead to overall decrease in objective.

A10. Using relaxation for each coordinate and Newton for inner 1D minimization.

3.4 Exams

Make-up Exam S_1
Unconstrained Optimization

Exercise #1 (6 points)

Answer true or false, justifying your answer:

1. Every coercive function is a convex function.

2. Let $f : A \subset H \to \mathbb{R}$, Gâteaux-differentiable on A, with A convex, then: f is convex if and only if $\forall (u, v) \in H \times H$,

$$f(u) \geq f(v) + \langle \nabla f(v), u - v \rangle.$$

3. Optimization problems in economics are nonlinear problems.

Exercise #2 (7 points)

Solve the following problem:

$$\begin{cases} \min \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle \\ x \in \mathbb{R}^3 \end{cases}$$

with

$$A = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & -1 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}.$$

Exercise #3 (7 points)

1. **What is the principle of the conjugate gradient method?**

2. **Write the conjugate gradient algorithm in the case of a quadratic form.**

University of Khemis-Miliana A.Y.: 2024–2025
Faculty of Material Sciences and Computer Science Level: L3
Department of Mathematics Duration: 01h30 Date: June 22, 2025

January 2025 Exam_S1

Note:

- No documents are allowed during the exam.
- Mobile phones are prohibited during the exam period.
- Answers must be fully justified.

Exercise 01 (06 pts)

We consider the following problem (P):

$$\min_{x\in\mathbb{R}^2} J(X) = J(x_1, x_2) = \frac{1}{2}(2x_1^2 + 2x_2^2) - 4x_1 + 8x_2$$

1. Solve problem (P).
2. Perform one iteration of Newton's method starting from the point $x^T = (0, -4)$ with a step size equal to 1.

Exercise 02 (07 pts)

We consider the following problem (Q):

$$\min_{x\in\mathbb{R}^2} J(X) = J(x_1, x_2) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$$

where A is a symmetric positive definite matrix.

1. Study the existence and uniqueness of solutions, then solve this problem.
2. Show that if $(d^{(0)}, d^{(1)}, \dots, d^{(k)})$ is a set of A-conjugate directions, then the vectors $(d^{(0)}, d^{(1)}, \dots, d^{(k)})$ are linearly independent.
3. Write the conjugate gradient algorithm for solving (Q).

Exercise 03 (07 pts)

We consider the following optimization problem (M):

$$\min_{x\in\mathbb{R}^2} J(X) = J(x_1, x_2) = \frac{1}{2}(3x_1^2 + 5x_2^2) - 2x_1x_2 - 6x_1 - 12x_2 + \pi$$

1. Write problem (M) in the form:
$$\min_{x\in\mathbb{R}^2} \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle + c$$
where A is a symmetric positive definite matrix.
2. Write the fixed-step gradient algorithm (with step α) for solving a quadratic problem.
3. For which values of α does the sequence $x^{(k)}$ defined by the fixed-step gradient method converge to the solution x of (M)? What is the optimal step α?

Given: $\lambda_{\min} = 3.69, \quad \lambda_{\max} = 12.73$

MathWorks®

MATLAB® Basic Functions Reference

MATLAB Environment

`clc`	Clear command window
`help fun`	Display in-line help for `fun`
`doc fun`	Open documentation for `fun`
`load("filename","vars")`	Load variables from `.mat` file
`uiimport("filename")`	Open interactive import tool
`save("filename","vars")`	Save variables to file
`clear item`	Remove items from workspace
`examplescript`	Run the script file named `examplescript`
`format style`	Set output display format
`ver`	Get list of installed toolboxes
`tic, toc`	Start and stop timer
`Ctrl+C`	Abort the current calculation

Operators and Special Characters

`+, -, *, /`	Matrix math operations
`.*, ./`	Array multiplication and division (element-wise operations)
`^, .^`	Matrix and array power
`\`	Left division or linear optimization
`.', '`	Normal and complex conjugate transpose
`==, ~=, <, >, <=, >=`	Relational operators
`&&, ||, ~, xor`	Logical operations (AND, NOT, OR, XOR)
`;`	Suppress output display
`...`	Connect lines (with break)
`% Description`	Comment
`'Hello'`	Definition of a character vector
`"This is a string"`	Definition of a string
`str1 + str2`	Append strings

Special Variables and Constants

`ans`	Most recent answer
`pi`	π=3.141592654...
`i, j, 1i, 1j`	Imaginary unit
`NaN, nan`	Not a number (i.e., division by zero)
`Inf, inf`	Infinity
`eps`	Floating-point relative accuracy

Defining and Changing Array Variables

`a = 5`	Define variable a with value 5

`A = [1 2 3; 4 5 6]`
`A = [1 2 3`
`     4 5 6]`
	Define A as a 2x3 matrix
	“space” separates columns
	“;” or new line separates rows

`[A,B]`	Concatenate arrays horizontally
`[A;B]`	Concatenate arrays vertically
`x(4) = 7`	Change 4th element of x to 7
`A(1,3) = 5`	Change A(1,3) to 5
`x(5:10)`	Get 5th to 10th elements of x
`x(1:2:end)`	Get every 2nd element of x (1st to last)
`x(x>6)`	List elements greater than 6
`x(x==10)=1`	Change elements using condition
`A(4,:)`	Get 4th row of A
`A(:,3)`	Get 3rd column of A
`A(6, 2:5)`	Get 2nd to 5th element in 6th row of A
`A(:,[1 7])=A(:,[7 1])`	Swap the 1st and 7th column
`a:b`	[a, a+1, a+2, …, a+n] with a+n≤b
`a:ds:b`	Create regularly spaced vector with spacing ds
`linspace(a,b,n)`	Create vector of n equally spaced values
`logspace(a,b,n)`	Create vector of n logarithmically spaced values
`zeros(m,n)`	Create m x n matrix of zeros
`ones(m,n)`	Create m x n matrix of ones
`eye(n)`	Create a n x n identity matrix
`A=diag(x)`	Create diagonal matrix from vector
`x=diag(A)`	Get diagonal elements of matrix
`meshgrid(x,y)`	Create 2D and 3D grids
`rand(m,n), randi`	Create uniformly distributed random numbers or integers
`randn(m,n)`	Create normally distributed random numbers

Complex Numbers

`i, j, 1i, 1j`	Imaginary unit
`real(z)`	Real part of complex number
`imag(z)`	Imaginary part of complex number
`angle(z)`	Phase angle in radians
`conj(z)`	Element-wise complex conjugate
`isreal(z)`	Determine whether array is real

MathWorks

Elementary Functions

`sin(x), asin`	Sine and inverse (argument in radians)
`sind(x), asind`	Sine and inverse (argument in degrees)
`sinh(x), asinh`	Hyperbolic sine and inverse (arg. in radians)
Analogous for the other trigonometric functions: `cos`, `tan`, `csc`, `sec`, and `cot`	
`abs(x)`	Absolute value of x, complex magnitude
`exp(x)`	Exponential of x
`sqrt(x), nthroot(x,n)`	Square root, real nth root of real numbers
`log(x)`	Natural logarithm of x
`log2(x), log10`	Logarithm with base 2 and 10, respectively
`factorial(n)`	Factorial of n
`sign(x)`	Sign of x
`mod(x,d)`	Remainder after division (modulo)
`ceil(x), fix, floor`	Round toward +inf, 0, -inf
`round(x)`	Round to nearest decimal or integer

Tables

`table(var1,...,varN)`	Create table from data in variables var1, ..., varN
`readtable("file")`	Create table from file
`array2table(A)`	Convert numeric array to table
`T.var`	Extract data from variable var
`T(rows,columns),` `T(rows,["col1","coln"])`	Create a new table with specified rows and columns from T
`T.varname=data`	Assign data to (new) column in T
`T.Properties`	Access properties of T
`categorical(A)`	Create a categorical array
`summary(T), groupsummary`	Print summary of table
`join(T1, T2)`	Join tables with common variables

Tasks (Live Editor)

Live Editor tasks are apps that can be added to a live script to interactively perform a specific set of operations. Tasks represent a series of MATLAB commands. To see the commands that the task runs, show the generated code.

Common tasks available from the Live Editor tab on the desktop toolstrip:

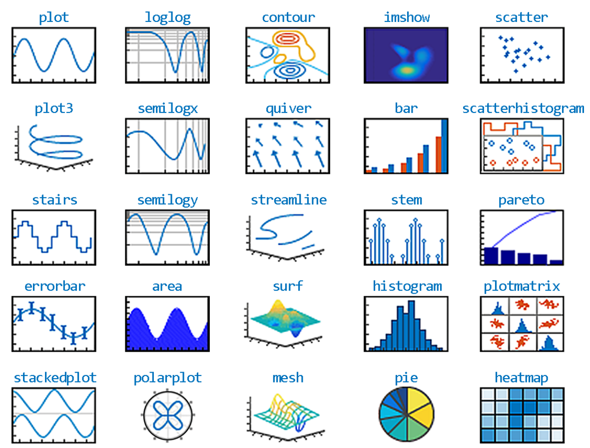
- Clean Missing Data
- Find Change Points
- Remove Trends
- Clean Outlier
- Find Local Extrema
- Smooth Data

Plotting

`plot(x,y,LineSpec)`
Line styles:
`-, --, :, -.`
Markers:
`+, o, *, ., x, s, d`
Colors:
`r, g, b, c, m, y, k, w`
	Plot y vs. x
	(`LineSpec` is optional)
	`LineSpec` is a combination of `linestyle`, `marker`, and `color` as a string.
	Example: `"-r"`
	= red solid line without markers

`title("Title")`	Add plot title
`legend("1st", "2nd")`	Add legend to axes
`x/y/zlabel("label")`	Add x/y/z axis label
`x/y/zticks(ticksvec)`	Get or set x/y/z axis ticks
`x/y/zticklabels(labels)`	Get or set x/y/z axis tick labels
`x/y/ztickangle(angle)`	Rotate x/y/z axis tick labels
`x/y/zlim`	Get or set x/y/z axis range
`axis(lim), axis style`	Set axis limits and style
`text(x,y,"txt")`	Add text
`grid on/off`	Show axis grid
`hold on/off`	Retain the current plot when adding new plots
`subplot(m,n,p),` `tiledlayout(m,n)`	Create axes in tiled positions
`yyaxis left/right`	Create second y-axis
`figure`	Create figure window
`gcf, gca`	Get current figure, get current axis
`clf`	Clear current figure
`close *all*`	Close open figures

Common Plot Types



Plot Gallery: *mathworks.com/products/matlab/plot-gallery*

MathWorks

Programming Methods

Functions

```
% Save your function in a function file or at the end
% of a script file. Function files must have the
% same name as the 1st function
function cavg = cumavg(x) %multiple args. possible
    cavg=cumsum(vec)./(1:length(vec));
end
```

Anonymous Functions

```
% defined via function handles
fun = @(x) cos(x.^2)./abs(3*x);
```

Control Structures

`if, elseif` Conditions

```
if n<10
    disp("n smaller 10")
elseif n<=20
    disp("n between 10 and 20")
else
    disp("n larger than 20")
```

Switch Case

```
n = input("Enter an integer: ");
switch n
    case -1
        disp("negative one")
    case {0,1,2,3} % check four cases together
        disp("integer between 0 and 3")
    otherwise
        disp("integer value outside interval [-1,3]")
end % control structures terminate with end
```

For-Loop

```
% loop a specific number of times, and keep
% track of each iteration with an incrementing
% index variable
for i = 1:3
    disp("cool");
end % control structures terminate with end
```

While-Loop

```
% loops as long as a condition remains true
n = 1;
nFactorial = 1;
while nFactorial < 1e100
    n = n + 1;
    nFactorial = nFactorial * n;
end % control structures terminate with end
```

Further programming/control commands

`break`	Terminate execution of for- or while-loop
`continue`	Pass control to the next iteration of a loop
`try, catch`	Execute statements and catch errors

Numerical Methods

`fzero(fun,x0)`	Root of nonlinear function
`fminsearch(fun,x0)`	Find minimum of function
`fminbnd(fun,x1,x2)`	Find minimum of fun in [x1, x2]
`fft(x), ifft(x)`	Fast Fourier transform and its inverse

Integration and Differentiation

`integral(f,a,b)`	Numerical integration (analogous functions for 2D and 3D)
`trapz(x,y)`	Trapezoidal numerical integration
`diff(X)`	Differences and approximate derivatives
`gradient(X)`	Numerical gradient
`curl(X,Y,Z,U,V,W)`	Curl and angular velocity
`divergence(X,..,W)`	Compute divergence of vector field
`ode45(ode,tspan,y0)`	Solve system of nonstiff ODEs
`ode15s(ode,tspan,y0)`	Solve system of stiff ODEs
`deval(sol,x)`	Evaluate solution of differential equation
`pdepe(m,pde,ic,... bc,xm,ts)`	Solve 1D partial differential equation
`pdeval(m,xmesh,... usol,xq)`	Interpolate numeric PDE solution

Interpolation and Polynomials

`interp1(x,v,xq)`	1D interpolation (analogous for 2D and 3D)
`pchip(x,v,xq)`	Piecewise cubic Hermite polynomial interpolation
`spline(x,v,xq)`	Cubic spline data interpolation
`ppval(pp,xq)`	Evaluate piecewise polynomial
`mkpp(breaks,coeffs)`	Make piecewise polynomial
`unmkpp(pp)`	Extract piecewise polynomial details
`poly(x)`	Polynomial with specified roots x
`polyeig(A0,A1,...,Ap)`	Eigenvalues for polynomial eigenvalue problem
`polyfit(x,y,d)`	Polynomial curve fitting
`residue(b,a)`	Partial fraction expansion/decomposition
`roots(p)`	Polynomial roots
`polyval(p,x)`	Evaluate poly p at points x
`polyint(p,k)`	Polynomial integration
`polyder(p)`	Polynomial differentiation

MathWorks®
Accelerating the pace of engineering and science

Matrices and Arrays

`length(A)`	Length of largest array dimension
`size(A)`	Array dimensions
`numel(A)`	Number of elements in array
`sort(A)`	Sort array elements
`sortrows(A)`	Sort rows of array or table
`flip(A)`	Flip order of elements in array
`squeeze(A)`	Remove dimensions of length 1
`reshape(A,sz)`	Reshape array
`repmat(A,n)`	Repeat copies of array
`any(A), all`	Check if any/all elements are nonzero
`nnz(A)`	Number of nonzero array elements
`find(A)`	Indices and values of nonzero elements

Linear Algebra

`rank(A)`	Rank of matrix
`trace(A)`	Sum of diagonal elements of matrix
`det(A)`	Determinant of matrix
`poly(A)`	Characteristic polynomial of matrix
`eig(A), eigs`	Eigenvalues and vectors of matrix (subset)
`inv(A), pinv`	Inverse and pseudo inverse of matrix
`norm(x)`	Norm of vector or matrix
`expm(A), logm`	Matrix exponential and logarithm
`cross(A,B)`	Cross product
`dot(A,B)`	Dot product
`kron(A,B)`	Kronecker tensor product
`null(A)`	Null space of matrix
`orth(A)`	Orthonormal basis for matrix range
`tril(A), triu`	Lower and upper triangular part of matrix
`linsolve(A,B)`	Solve linear system of the form AX=B
`lsqminnorm(A,B)`	Least-squares solution to linear equation
`qr(A), lu, chol`	Matrix decompositions
`svd(A)`	Singular value decomposition
`gsvd(A,B)`	Generalized SVD
`rref(A)`	Reduced row echelon form of matrix

Descriptive Statistics

`sum(A), prod`	Sum or product (along columns)
`max(A), min, bounds`	Largest and smallest element
`mean(A), median, mode`	Statistical operations
`std(A), var`	Standard deviation and variance
`movsum(A,n), movprod, movmax, movmin, movmean, movmedian, movstd, movvar`	Moving statistical functions n = length of moving window
`cumsum(A), cumprod, cummax, cummin`	Cumulative statistical functions
`smoothdata(A)`	Smooth noisy data
`histcounts(X)`	Calculate histogram bin counts
`corrcoef(A), cov`	Correlation coefficients, covariance
`xcorr(x,y), xcov`	Cross-correlation, cross-covariance
`normalize(A)`	Normalize data
`detrend(x)`	Remove polynomial trend
`isoutlier(A)`	Find outliers in data

Symbolic Math*

`sym x, syms x y z`	Declare symbolic variable
`eqn = y == 2*a + b`	Define a symbolic equation
`solve(eqns,vars)`	Solve symbolic expression for variable
`subs(expr,var,val)`	Substitute variable in expression
`expand(expr)`	Expand symbolic expression
`factor(expr)`	Factorize symbolic expression
`simplify(expr)`	Simplify symbolic expression
`assume(var,assumption)`	Make assumption for variable
`assumptions(z)`	Show assumptions for symbolic object
`fplot(expr), fcontour, fsurf, fmesh, fimplicit`	Plotting functions for symbolic expressions
`diff(expr,var,n)`	Differentiate symbolic expression
`dsolve(deqn,cond)`	Solve differential equation symbolically
`int(expr,var,[a, b])`	Integrate symbolic expression
`taylor(fun,var,z0)`	Taylor expansion of function

*requires Symbolic Math Toolbox

mathworks.com/help/matlab

Bibliography

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[2] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd Edition, Springer, 2006.

[3] J. S. Arora, *Introduction to Optimum Design*, Academic Press, 2017.

[4] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1999.

[5] R. Fletcher, *Practical Methods of Optimization*, Wiley, 1987.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[7] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, In *International Conference on Learning Representations (ICLR)*, 2015.

[8] D. C. Liu and J. Nocedal, On the limited memory BFGS method for large scale optimization, *Mathematical Programming*, 45(1):503–528, 1989.

[9] J. A. Nelder and R. Mead, A simplex method for function minimization, *Computer Journal*, 7(4):308–313, 1965.

[10] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, 1970.

[11] R. Tyrrell Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, NJ, 1970.

[12] Ivar Ekeland and Roger Temam. *Analyse Convexe et Problèmes Variationnels.* 3^{e} édition, Dunod, Paris, 1999.

[13] Ivar Ekeland and Roger Temam. *Convex Analysis and Variational Problems.* 3rd edition, Dunod, Paris, 1999.

[14] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer Science & Business Media, 2nd edition, 2006.

[15] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[16] John E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, 1983.

[17] M. Bergounioux.

[18] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.