

General Introduction

Context and Teaching Objectives

This course follows directly from the introductory programming course in Python taught during the previous semester. The aim is no longer to learn the syntax of the language, but to **employ it as an instrument of scientific computation**. The objectives are threefold: to deepen the use of Python’s scientific libraries (**NumPy**, **SciPy**, **Matplotlib**, **SymPy**, **Pandas**) for solving complex problems; to master advanced techniques of scientific computing—linear algebra, optimization, symbolic computation, and probabilistic simulation; and finally, to produce visualizations and data analyses of professional quality.

The course therefore presupposes two prerequisites: a **command of the fundamentals of Python** (syntax, functions, data structures), acquired in Semester 2, and a **university-level mathematical background** in analysis, linear algebra, and probability. It is precisely at the intersection of these two domains that the subject matter lies: the student will learn to translate a mathematical problem into an algorithm, to execute it, and to interpret the result obtained.

Pedagogical Progression

The course follows a deliberately **general-to-specific** progression. The first chapter establishes an overview of Python’s scientific ecosystem, presenting the role and complementarity of its main libraries, together with the Jupyter Notebook working environment. The four subsequent chapters then each deepen one library and its associated mathematical domain:

Chapter	Theme	Main Library
1	Python Scientific Ecosystem and Jupyter Notebook	Overview
2	Advanced Linear Algebra	SciPy
3	Differential Calculus and Integration	SymPy
4	Probability and Simulation	NumPy
5	Data Analysis	Pandas

This organization is not arbitrary. It reflects the very structure of contemporary scientific computing, in which each type of problem—solving a linear system, computing a derivative, estimating a probability, processing a dataset—calls for a specific tool, while resting on the same numerical foundations.

Targeted Competencies

By the end of this course, the student should be able to:

- **select** the appropriate library for a given scientific problem;
- **combine** several libraries within a single project;
- **perform** rigorous numerical and symbolic computations;

- **analyze** and **visualize** experimental or simulated data;
- **document** a scientific project in a reproducible and well-structured notebook.

The transversal competency pursued, however, remains singular: to move from the isolated use of scattered tools toward a genuine **integrated scientific methodology**, in which computation, analysis, visualization, and documentation form a coherent whole.