Introduction to Scientific Computing

Martino Andrea Scarpolini

December 2025

Dates

April 8-24 2026.

Course Overview

This short series of lectures introduces the fundamental concepts and practices of scientific computing, with a particular focus on the numerical solution of ordinary and partial differential equations (ODEs and PDEs). Students will explore the mathematical and computational tools used to model and simulate physical, biological, and engineering systems. Throughout the course, we will employ numerical calculus to solve problems ranging from simple ODEs to classical PDEs arising in continuum mechanics and related fields.

A central objective of the course is to develop both conceptual and practical understanding in contemporary scientific programming. We will examine standard compiled languages (C and Fortran), modern high-level languages (Python), and just-in-time compiled environments (Julia), highlighting their different roles in modern High Performance Computing (HPC). Emphasis will be placed on writing clear, efficient, and reproducible scientific code using modern programming paradigms.

Scientific computing is inherently collaborative. Accordingly, the course will introduce essential software-engineering practices, including version control with Git and strategies for effective collaborative code development. Students will gain hands-on experience collaborating on shared code repositories.

Because HPC systems are typically accessed through text-based interfaces, the course begins with a brief introduction to Unix-like operating systems and command-line shells (bash/zsh). Students are expected to have access to a Unix-like environment (e.g. Linux or macOS) with standard compilation tools installed.

Schedule of Lessons

- **Lesson 1** [Wednesday 08/04/2026]: Unix-like operating systems and the Bash shell: navigation, scripting, and workflow automation.
- **Lesson 2** [Friday 10/04/2026]: Introduction to scientific programming languages: Python, C/Fortran, and Julia. Interpreted vs. compiled vs. JIT-compiled paradigms.
- **Lesson 3** [Tuesday 14/04/2026]: Numerical methods for ODEs.
- **Lesson 4** [Friday 17/04/2026]: Introduction to Git.
- **Lesson 5** [Tuesday 21/04/2026]: Collaborative development with Git.
- **Lesson 6** [Wednesday 15/04/2026]: Numerical methods for PDEs I.
- **Lesson 7** [Wednesday 22/04/2026]: Numerical methods for PDEs II.
- **Lesson 8** [Friday 24/04/2026]: Discovery of chosen open source projects.