032CM - 2026

PROGRAMMING FOR COMPUTATIONAL CHEMISTRY

Introduction to the course

Gianluca Levi

gianluca.levi@units.it, giale@hi.is

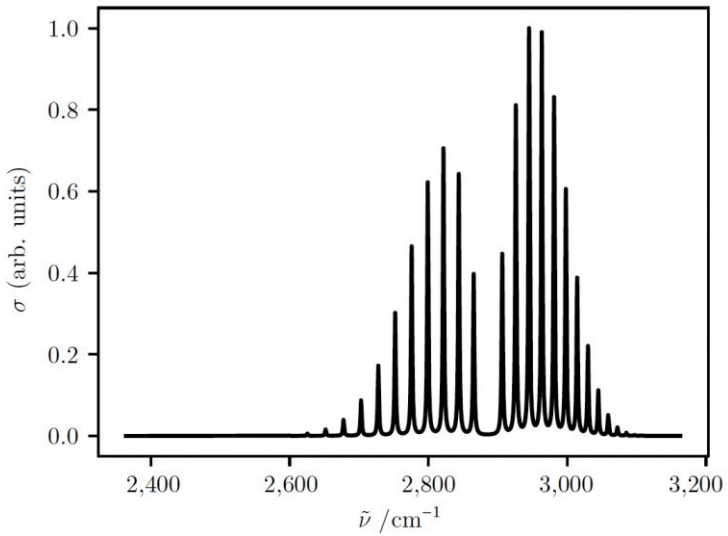
Office: Building C11, 3rd floor, Room 329

Fall 2026

Why learn programming for computational chemistry?

Why learn programming for computational chemistry?

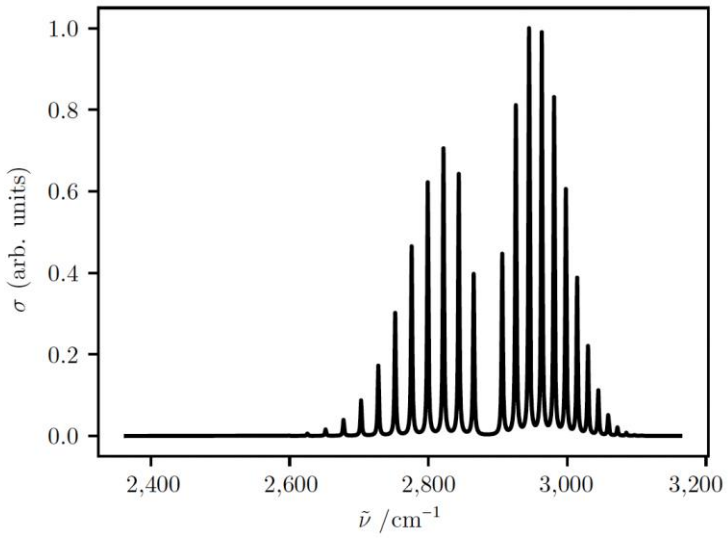
Create computer-based tools to address tasks in chemistry (and beyond) in an **efficient**, **automated**, and **reproducible** manner



Simulated IR spectrum of $^{1}H^{35}Cl$ from Python for Chemists, C. Hill

Why learn programming for computational chemistry?

Create computer-based tools to address tasks in chemistry (and beyond) in an **efficient**, **automated**, and **reproducible** manner



Simulated IR spectrum of $^{1}H^{35}Cl$ from Python for Chemists, C. Hill

Work with **large computational chemistry software packages** (e.g. atomic-scale simulation programs)

- **Understand** what happens under the hood
- **Adapt and modify** implementations to address specific research questions



Why learn programming in the AI era?

Why learn programming in the AI era?

Develop **algorithmic and critical thinking**

- Develop transferable **problem solving skills**

  Break down complex problems into logical, algorithmic steps

- **Understand** computational methods and theory

  Gain deeper insights into scientific problems

Use AI critically: define the scientific task, test and validate AI generated code

Course plan

Each class First half lecture, second half lab exercises

Assignments About one per week, we start each assignment together in class

---

UNIX and Fortran foundations — Weeks 1–2

UNIX/Linux shell, basics of programming and Fortran

Python and scientific computing — Weeks 3–6

Python fundamentals, JupyterLab, NumPy, Matplotlib
Numerical integration, root finding

Python scripting and molecular dynamics — Weeks 6–8

Scripts and modules, time propagation algorithms, Fourier series, normal modes

Molecular tools and Git — Weeks 9–10

ASE and NGLview, version control, project organisation

Assignments

Problem sets provided by the teacher (approx. one per week)

- We start working on the problems during the hands-on session in class, then complete at home
- **Submit report** to the teacher by email including input and output text
- You may collaborate on the assignments, but submissions must be individual

Submit at least 70% of the assignments to be eligible for the written exam

Assessment

Written exam (2 hours)	Write and execute Python code
Oral exam (2 days after the written exam)	Discussion of the written exam and topics covered in the lectures

Study material

Lecture slides (on Moodle and Teams)

Textbooks

- Fortran for Scientists and Engineers, 4th edition, S. J. Chapman, McGraw-Hill Education
- Programming for Computations - Python, 2nd edition, S. Linge and H. P. Langtangen, Springer (Open Access: https://link.springer.com/book/10.1007/978-3-030-16877-3)
- Python for Chemists, C. Hill, Cambridge University Press

Register on **Moodle**: https://moodle2.units.it/course/view.php?id=17203

Register on **Moodle**: https://moodle2.units.it/course/view.php?id=17203

Join the **Teams page** of the course

- Find the Teams code in the course catalogue: https://www.units.it/catalogo-della-didattica-a-distanza

CHIMICA (SM13A)

INSEGNAMENTO	CODICE TEAMS	PERIODO	DOCENTI
PROGRAMMING FOR COMPUTATIONAL CHEMISTRY (032CM - 2026 - [SM13A+4+ - Ord. 2025] analitica e ambiente - AC 2)	yvozlrq	S1	LEVI GIANLUCA

- Join the Teams page using the code
- **Course material and recordings of the lectures** in Shared section of General channel

