MSE 305
Introduction to atomic-scale modeling

Prof. Michele Ceriotti

TAs:
Markus Fasching; Alessandro Forina; Sandra Saade

Shut up and calculate![1]

Revisiting basic concepts from materials science in terms of

1. An atomic-scale picture of matter
2. Simple models and explicit numerical calculations

What should you learn?

1. A more intuitive hands-on understanding of MS concepts
2. Basic practical skills in atomistic modeling (with Python!)
3. Some concrete examples of the use of modeling to predict materials properties

[1] https://doi.org/10.1063/1.1768652

Subjects covered

- Intro and Python re-cap
- Working with atomic-scale structures (2h, graded but not counted)
- Computational crystallography. (4h)
- Interatomic potentials (4h)
- Local stability of structures and defects (4h)
- Molecular dynamics (4h)
- Atomistic machine learning (4h)

Lecture format

- Hands-on, computational approach to modeling
- Entirely based on Jupyter notebooks based on custom Jupyter widgets (watch out for bugs…)
- Summary of the theory, exercises, self-evaluation, grading by the TAs
- Informal introduction to the topic (flipped class)
- Interactive Q&A during the lecture
- Aim: > 70% of the work accomplished during lecture hours
- Minimal programming skills expected. App-mode hides most of the coding overhead, and notebooks can be run from the noto.epfl.ch platform

Rules of the house

- Attendance is highly recommended
- Exercises should be done during lecture hours, or shortly after
- Submit your solutions by Thursday after the end of lecture (most lectures span two weeks)
- 75% of the grade: assessment of the solutions
  - scale from 1 (worst) to 10 (best). first assignment doesn't count, others all count the same
  - grades and feedback roughly within 1 week of submission
- 25% of the grade: 15' discussion on one exercise, chosen at random, at the end of the course (details to be arranged)

Collaboration vs plagiarism

- You have to write yourself and understand all answers and code you submit

- You can discuss among yourselves, look up documentation and stackoverflow, ask for help but you cannot copy verbatim the answer of another student or ChatGPT, or try to reverse-engineer it from the notebook source

- If you can’t get to the bottom of an answer, or are late for an assignment, it’s better to submit partial answers than to copy