Computational methods: general information

Mario Galante, Almaz Khabibrakhmanov, Alexandre Tkatchenko

September 16, 2021

The main idea of the course is to provide knowledge and practical experience of the numerical techniques that constitute the basis of Computational Physics and Chemistry. Each lecture will be comprised of an introduction to the theory behind a given technique, followed by a practical session centered on its implementation and application for well-known problems. Some emphasis will be put on the analysis of the outcome of the variation of physical parameters for the given problem. The first part will consolidate simple notions of python3 programming and cover the basic algorithms necessary to solve simple equations. The second part will provide an introduction on more advanced methods such as Monte Carlo and Machine Learning, as well as cover the numerical solution of the time-dependent Schrödinger equation and molecular dynamics.

Learning outcomes:

- knowledge of standard algorithms adopted in computational Physics and Chemistry and their limitations
- capability of writing a program to find the numerical solution of simple physical problems and analyse the resulting data
- understanding the basics of Monte Carlo methods, molecular dynamics simulations and machine learning

Lessons:

1. **Introduction to python3** Variables, arrays, functions, for loops, if statements, function and data visualisation
2. **Numerical differentiation** centered, forward, backwards first derivatives; higher order derivatives, numerical error and increment dependence
3. **Minima of functions** bisection and Newton-Raphson methods; dependence of starting point and critical failures
4. **Numerical integration** midpoint method, Simpson formula, integration of oscillatory functions (as more advanced?)
5. **1D differential equation** solution of the Newton equation for a projectile including factors such as air resistance; Damped harmonic oscillator (?)
6. **Partial differential equations** solution of the heat equation (or 2D Poisson equation)
7. **Random numbers**: their generation and notions of probability; use of random number generators in numerical recipes. Simulation of a dice toss, simple Brownian motion

8. **Molecular dynamics on simple systems**: implementation of the dynamics of a system of hard spheres in a 2D box; checking the conservation of energy and choosing a timestep
9. **Simulations in a real package**: use of an external code (LAMMPS) to compare simulations in microcanonical and canonical ensemble; comparison of numerical results with known statistical mechanical formulas; to check the influence of physical parameters on the behaviour of a simple system (e. g. Lennard-Jones gas)
10. **Monte Carlo** basics, application to integrals, system optimization, solution of the Ising model
11. **Schrödinger equation** static solution for a particle in a box; dynamical solution for a particle passing through a potential energy barrier
12. **Intro to Machine Learning** e. g. basics of image recognition

Assessment:

- midterm exercise : solution and analysis of a simple physical problem using notions covered up to Lesson 6 (20%)
- final project : adaptation of the codes developed during the course to solve a problem within molecular dynamics, Schrodinger dynamics, machine learning and Monte Carlo methods (40%)
- oral exam (40%)

Comment: shall we give them small homework after each class?