# Physique Numérique — Computational Physics (M1)

Lecturer: Tommaso Roscilde (tommaso.roscilde@ens-lyon.fr)
TDs: Filippo Caleca (filippo.caleca@ens-lyon.fr)

Web-page: https://sites.google.com/site/roscilde/home/teaching/computational-physics-m1

## Course description

Numerical techniques have become pervasive in any field of Physics, from theory to experiments. This course offers a survey of some of the most relevant numerical problems raised by modern Physics (eigenvalue problems, calculation of statistical sums, minimisation of many-variable functions, and solution of differential equations), and of some of the most common and efficient computer algorithms devised to solve these problems.
The lectures (held in English) will be accompanied by hands-on TD sessions, each aimed at the actual solution of a Physics problem using the techniques described in the lectures.

*Tentative course scheme:*
**Large eigenvalue problems**: power method; Lanczos method
(with links to: classical mechanics, quantum mechanics, etc.)
**Numerical integration**: random walks and Markov chains; Monte Carlo method
(with links to: statistical physics, solid-state physics, high-energy physics etc. )
**Numerical optimization**: gradient descent and its variants; simulated annealing
(with links to: quantum mechanics, machine learning, etc. )
**Differential equations:** linear equations and algebraic methods; nonlinear equations: Runge-Kutta schemes; spectral methods; Verlet scheme
(with links to: classical mechanics, quantum mechanics, hydrodynamics etc.)

## TD sessions

The TD sessions will be hands-on sessions, in which you will be required to *write a little computer code* - either alone or in a duo (binôme) - and do physics with it. You should bring with you your laptop (or have one shared with your duo partner) and choose a programming platform of your choice (Python, Matlab, Mathematica, Fortran, C, etc.), which is *operational* on your laptop (namely you should make sure that you have installed the software that you need, e.g. a C/Fortran compiler or a working Python distribution).

## Requirements

An introductory class in informatics (ex. Outils numériques (L3)); a minimal familiarity with computer coding (either Python, Matlab, Mathematica, Fortran, C, etc.); elementary analytical mechanics, quantum mechanics, statistical physics

## Exam mode

The exam will consist of a **take-home computational project,** involving the demonstration of a physical effect via a computer simulation; or the development of an algorithm for the efficient numerical solution of a physical problem.
You can work **in a duo (binôme) or alone** on the project. Ideally *you* propose the project on the basis of a subject of your own interest — it can be based on a physical problem you want to solve with a computational approach, or an algorithm you want to implement which is useful to solve known physical problems, or other things as well. If you are out of ideas, we can of course help you, suggesting a few potential problems. In any case, *you will have to consult us before starting to work on your project — you need your project to be "approved"* in terms of feasibility/ appropriateness in the context of the lecture subjects.
Your work should culminate in a ***report,*** **which should not exceed 10 pages** (including front page, bibliography and potential appendices), and which should ideally contain original numerical results that you obtained by yourself, as well as an introduction to the approach that allowed you to obtain them.

## Bibliography

The literature on computational physics is rather vast — you can find below a few representative examples of potentially useful books:

J.-M. Thijssen, *Computational Physics* (Cambridge, 2007);

T. Pang, *An Introduction to Computational Physics* (Cambridge, 2006);

R. H. Landau, M. J. Páez and C. C. Bordeianu, *Computational Physics - Problem Solving with Python*, Wiley-VCH (2015);

A. Iserles, *A First Course in the Numerical Analysis of Differential Equations* (Cambridge, 2009);J. D. Lambert, *Numerical Methods for Ordinary Differential Systems* (Wiley, 1993).