Computational Physics I (PHY4140/PHY5340)

Instructor: Mykyta Chubynsky
Email: `chubynsky@gmail.com`
Office: Macdonald 316
Office hours: Monday 4:15 – 5:00 pm and by appointment

TA: Tyler Shendruk
Email: `tshendruk@gmail.com`

Schedule

Lectures: Monday & Wednesday, 2:30 – 4:00 pm, Morisset 256
Lab: Tuesday, 2:30 – 4:30 pm, Cube 104

Goals of the course:

- Give some knowledge about basic numerical methods used to solve mathematical problems arising in physics (integration, solution of algebraic and differential equations, optimization), their strengths and weaknesses and suitability for particular problems. Show how to derive the methods (without too much mathematical rigour), analyze their stability and associated numerical error;
- Give a general overview of the field of computational physics (examples of problems and methods, current trends), some (very basic) information about developing scientific software (programming languages, optimization, etc.), as well as using available software;
- Provide the opportunity to practice writing code modeling different physical systems using the methods studied in the course;
- Students should be able to learn some interesting physics by using the code developed by them in the course to explore various systems and phenomena (nonlinear dynamics, waves, quantum systems, etc.) that are difficult or impossible to study analytically.

What this course is not

- Not a programming or software development/engineering course — only minimal information will be given;
- Not a course on using specific software packages and libraries;
- Will not cover traditional discrete algorithms that computer scientists like so much (sorting, searching, data structures, graphs, etc.), even though some are occasionally useful in computational physics;
- Not a course on processing experimental data. Data fitting will be covered in Computational Physics II;
- Stochastic methods (e.g., Monte Carlo) and their applications will be covered in Computational Physics II.

Tentative outline

(subject to change based on time constraints and student preferences)

- Introduction: overview of computational physics, programming languages, basics of developing scientific software (debugging, optimization);
- Roundoff and truncation (discretization) errors;
- Numerical differentiation, interpolation, integration;
- Fourier transforms;
- Algebraic equations (root finding);
- Ordinary differential equations (ODEs): initial value problems;
- Molecular dynamics;
- Nonlinear dynamics;
- ODEs: boundary value problems;
- Linear algebra (solving sets of linear equations, matrix inversion, eigenvalues);
- Finding maxima and minima (steepest descent, conjugate gradient methods);

- Partial differential equations: elliptic (e.g., Poisson equation), parabolic (e.g., diffusion equation, time-dependent Schrödinger equation), hyperbolic (e.g., wave equation);
- Introduction to the finite-element method.

Evaluation

Labs (1-week homeworks):	**30%**
Projects (2-week homeworks):	**40%**
Final exam (take-home):	**30%**

Textbooks

There is no required textbook.

An extremely useful series of reference books is
W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes*, Cambridge University Press.
There are different publication years and different languages (i.e., Numerical Recipes in C, Numerical Recipes in Fortran 77, etc.), but otherwise the books are similar. The website for the books is `http://www.nr.com/` . Older editions are available for free by chapter. New editions are available by paid subscription, but you can access a limited number of pages per month for free. This book covers nearly all methods I will be talking about and much more. It explains the methods, but without "unnecessary" mathematical rigour.

Academic fraud regulations

Students need to be aware of the university regulations on academic fraud that can be found at
`http://web5.uottawa.ca/admingov/regulation_13.html`