

Physics 512 Syllabus, 2025

This course focuses on equipping students with the tools needed to solve problems in physics using computers. The course will be taught using python, and students are strongly encouraged to have at least some familiarity with basic python coding by the beginning of the course. Assessment is via hands-on problem sets (40%), a final coding project (30%), and a take-home exam (30%). There will be an option to not do the final coding project, in which case the grade will be 60% problem sets and 40% exam. *Numerical Recipes, 3rd edition* covers many of the course subjects and though not required, it is strongly recommended. Lectures will consist of a mix of blackboard derivations and live/interactive coding examples demonstrating the principles in action. Tutorial sessions will be carried out in-person, and will focus on helping students implement/debug class concepts into working code. Students are encouraged to provide examples of computing challenges they face that can be used as practical examples.

Lectures will be held in Rutherford 114 from 8:35 AM to 9:55 AM on Tuesdays and Thursdays. Instructor office hours are TBD in ERP 333. Tutorial/debug sessions are TBD. I will do my best to stream/record lectures via Zoom at <https://mcgill.zoom.us/j/89934871559?pwd=Y9hHbavdgbpiRMGec4y1AdMsqXGLwh.1> on a *best-effort* basis. Sometimes glitches happen so I cannot guarantee all lectures will be recorded. If you are watching online and notice the lecture is not recording, please speak up!

All course materials (other than zoom recordings) will be posted to github at <https://github.com/sievers/phys512-2025/>. Problem sets will be submitted via github as well. Each student should get a github account, and invite the TA group on github. We will have a slack space for students to talk among themselves as well. You can join at TBD Problem sets are due at 11:59 PM on TBD.

In the event of extraordinary circumstances beyond the University's control, the content and/or evaluation scheme in this course is subject to change

Instructor information: Jonathan Sievers (jonathan dot sievers at mcgill dot ca). Office: ERP333. Phone: x2156.

Teaching assistants: Valentin Boettcher (valentin dot boettcher at mail dot mcgill dot ca), Aditya Chugh (aditya dot chugh at mail dot mcgill dot ca), Han Gao (han dot gao3 at mail dot mcgill dot ca), and Kenny Moc (Kenny dot moc at mail dot mcgill dot ca).

AI Policy: The point of this course is for you to learn how to get the computer to do what you need it to, and crucially, to understand when something has gone wrong and how to fix it. To do this, you need to write and debug code yourselves. Consequently, no AI code is allowed unless specifically authorized

by the instructor. I hope you will thank me one day when you manage to fix an incorrect code chatGPT has written for you.

Topics covered:

- Brief introduction/review of python and coding practices.
- Floating point math/roundoff error. How to optimize *e.g.* numerical derivatives. Resource: “What every computer scientist should know about floating point arithmetic.” by David Goldberg.
- Function interpolation and numerical integration. Error analysis of various techniques.
- Integration of ODEs. Runge Kutta methods. Stiff equations, and implicit techniques.
- Linear algebra and linear least-squares fitting. Singular value decomposition and its application to numerically unstable models. Legendre and Chebyshev polynomials. Iterative solutions to large problems using conjugate gradient.
- Nonlinear least-squares fitting. Newton’s method, Levenberg-Marquardt, and Markov Chain Monte Carlo.
- Discrete Fourier transforms. Convolutions and applications to image processing. Aliasing and the Nyquist theorem. Stationary noise and matched filters.
- Random variables. Transformation method, rejection method, and ratio-of-uniforms.
- Partial differential equations. Eulerian and Lagrangian techniques. Advection equation, stability analysis, and the CFL condition. Inviscid fluid flow, numerical dissipation, and the Lax method.
- Discussion of performance considerations and profiling. Basics of GPU computing.
- Brief introduction to machine learning.

Language Policy:

In accord with McGill University’s Charter of Student Rights, students in this course have the right to submit in English or in French any written work that is to be graded. This does not apply to courses in which acquiring proficiency in a language is one of the objectives.

Conformément à la Charte des droits de l’étudiant de l’Université McGill, chaque étudiant a le droit de soumettre en français ou en anglais tout travail écrit devant être noté, sauf dans le cas des cours dont l’un des objets est la maîtrise d’une langue.

Academic Integrity:

McGill University values academic integrity. Therefore, all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures. (Approved by Senate on 29 January 2003) (See McGill's guide to academic honesty for more information.)

L'université McGill attache une haute importance à l'honnêteté académique. Il incombe par conséquent à tous les étudiants de comprendre ce que l'on entend par tricherie, plagiat et autres infractions académiques, ainsi que les conséquences que peuvent avoir de telles actions, selon le Code de conduite de l'étudiant et des procédures disciplinaires (pour de plus amples renseignements, veuillez consulter le guide pour l'honnêteté académique de McGill.