

# PHYSICS 250: COMPUTATIONAL PHYSICS

**Instructor:** David W. Miller, MCP 245,  
773-702-7671, [David.W.Miller@uchicago.edu](mailto:David.W.Miller@uchicago.edu)

<b>Lecture Times:</b>	Tue–Thur 2:00pm–3:20pm
<b>Lecture Location:</b>	KPTC 101 (Kersten Physics Teaching Center)
<b>Textbook:</b>	None required, many suggested! (see Reading List)
<b>Canvas Course Site:</b>	<a href="https://canvas.uchicago.edu/courses/58627">https://canvas.uchicago.edu/courses/58627</a>
<b>Course GitHub Site:</b>	<a href="https://github.com/UChicagoPhysics/PHYS250">https://github.com/UChicagoPhysics/PHYS250</a>
<b>Course JupyterLab:</b>	<a href="https://binderhub.pile.uchicago.edu/">https://binderhub.pile.uchicago.edu/</a>
<b>Course Discussions:</b>	<a href="https://canvas.uchicago.edu/courses/58627/discussion_topics">https://canvas.uchicago.edu/courses/58627/discussion_topics</a>
<b>Prof. Office Hours:</b>	Tue. 3:30pm–4:30pm, and by appt.
<b>Computer Lab Hours:</b>	Help is available for PHYS 250 by TAs in the Computer Science Instructional Lab (CSIL), 1st floor of Crerar, on (TBC): Tue 7:00-9:00 pm in CSIL 1 & 2 Wed 2:30-3:50 pm in CSIL 1 & 2 Wed 7:00-9:00 pm in CSIL 1 & 2
<b>Teaching Assistants:</b>	See the <i>Additional Information</i> section.
<b>Description:</b>	Introduces the use of computers in physics. After an introduction to programming basics, we will cover numerical solutions to fundamental problems, techniques for manipulating large data sets, neural networks, and basic data analysis.
<b>Homework (80%):</b>	Homework & materials available on <a href="#">Canvas</a> and <a href="#">GitHub</a> . <b><i>Due Date: Thursdays.</i></b> Graded homework will be returned the following week. <b><i>Collaboration Policy:</i></b> Collaboration on issues, concepts, and approaches is encouraged, but the work <i>must be your own</i> .
<b>Final Project (20%):</b>	Dec 5, 2024. Poster presentation on approved topics (topic approval due Oct. 29). Reviewed by judges panel, final grade by instructor. (See Add'l Info)

## TENTATIVE COURSE OUTLINE:

The weekly coverage is subject to changes and adjustments as the course progresses.

Week	Week Of	Lecture Topics, Exams, Information
Week 1: Lec 1–2	Mon Sep 30	<ul style="list-style-type: none"> <li>Algorithmic thinking, programming structures</li> <li>Python, Jupyter, Unix, shell, git</li> <li><b>Relevant reading:</b> KN 1.1; LPB 1.5, 5.2,</li> </ul>
Week 2: Lec 3–4	Mon Oct 7	<ul style="list-style-type: none"> <li>Software design concepts, visualization</li> <li>Random number generators, errors</li> <li><b>Relevant reading:</b> KN 6.2; LPB 2.1</li> </ul>
Week 3: Lec 5–6	Mon Oct 14	<ul style="list-style-type: none"> <li>Ising model, Metropolis algorithm</li> <li><b>Relevant reading:</b> Sethna 8.1; KN 6.4; LPB 15.1–15.4</li> </ul>
Week 4: Lec 7–8	Mon Oct 21	<ul style="list-style-type: none"> <li>Minimization and the Monte Carlo method</li> <li><b>Relevant reading:</b> Franklin 12.1–12.6</li> </ul>
Week 5: Lec 9–10	Mon Oct 28	<ul style="list-style-type: none"> <li>Ordinary differential equations</li> <li><b>Relevant reading:</b> Franklin 2.1–2.6; KN 6.8; LPB 7.1–7.10, 9.1–9.8</li> </ul>
Week 6: Lec 11–12	Mon Nov 4	<ul style="list-style-type: none"> <li>Partial differential equations</li> <li><b>Relevant reading:</b> Franklin 4.1–4.4; LPB 17.1–17.19</li> </ul>
Week 7: Lec 13–14	Mon Nov 11	<ul style="list-style-type: none"> <li>Fourier transforms</li> <li>Data analysis techniques</li> <li><b>Relevant reading:</b> Franklin 7.1–7.7; KN 4.1–4.3; LPB 10.1–10.10</li> </ul>
Week 8: Lec 15–16	Mon Nov 18	<ul style="list-style-type: none"> <li>Data analysis techniques</li> <li>Neural networks</li> <li><b>Relevant reading:</b> Franklin 14.1–14.6</li> </ul>
Week 9: Holidays	Mon Nov 25	<ul style="list-style-type: none"> <li><b>Holiday:</b> No lectures for Thanksgiving Break</li> </ul>
Week 10: Lec 17	Mon Dec 2	<ul style="list-style-type: none"> <li>Neural networks</li> <li><b>Relevant reading:</b> Franklin 14.1–14.6</li> <li><b>Thur 5 Dec: Final project posters</b> 1:00–3:30pm in the KPTC (Lounge + Atrium)</li> </ul>

Recommended References:

- Press, *Numerical recipes : the art of scientific computing*
  - QA297.N866 2007
  - available in a limited form online [here](#)
  - python resources & exercises [here](#)
- Sethna, *Statistical Mechanics: Entropy, Order Parameters, and Complexity*
  - QC174.8.S48 2006eb
  - available as a [PDF here](#)
  - computational resources & exercises [here](#)
- Kinder & Nelson (KN), *A Student's Guide to Python for Physical Modeling*
  - ISBN: 9781400889426
  - computational resources & exercises [here](#)
- Franklin, *Computational Methods for Physics*
  - ISBN: 9781139525398
  - computational resources & exercises [here](#)
- Landau, Paez, Bordeianu (LPB), *Computational Physics, Problem Solving with Python*
  - QC20.82.L36 2007
  - computational resources & exercises [here](#)
  - The text book in the Library is actually, *Computational Physics, Problem Solving with Computers (2nd Ed.)* but the updated online version is more useful, I think
- Halterman, *Fundamentals of C++ Programming*
  - available as a [PDF here](#)
  - computational resources & exercises [here](#)

Supplementary Math Texts (can be helpful for algorithms):

- Arfken & Weber, *Mathematical Methods for Physicists* QA37.2.A740 1995
- Greenberg, *Advanced Engineering Mathematics* TA330.G725 1998

## ADDITIONAL INFORMATION FOR THE COURSE:

### Teaching Assistants (TA):

TA's will have office hours as well as be available in the CSIL lab for assistance.

#### **Office Hours:**

TA Name	Email	Office Hours	Location
Sravan Munagavalasa	<a href="mailto:sravan@uchicago.edu">sravan@uchicago.edu</a>	Tue 4:30-5:45 pm	KPTC 307
Max Topel	<a href="mailto:maxtopel@uchicago.edu">maxtopel@uchicago.edu</a>	Mon 3:00-4:00 pm	KPTC 307

#### **CSIL Lab Assistance**

TA Name	Email	Office Hours	CSIL Lab
Max Topel	<a href="mailto:maxtopel@uchicago.edu">maxtopel@uchicago.edu</a>	TUE 7:30-9:00 pm	CSIL 1&2
Sravan	<a href="mailto:sravan@uchicago.edu">sravan@uchicago.edu</a>	WED 2:30-3:50 pm	CSIL 1&2

### Schedule and Section Assignment Information:

- CSIL Lab Sections will begin meeting in Week 2 of Spring Quarter.

### Course Discussions

We will use Canvas Discussions for class discussions and questions. The goal is to be able to provide help fast and efficiently from classmates, the TA, and myself.

- Rather than emailing questions to the teaching staff, I encourage you to post your questions on Canvas.

### Final Poster Project

- **Topic:** The topic should fall within the class's theme of modeling and analyzing physical systems. A complete project will include software in a GitHub repository written in a language of your choice that can be run with minimal setup by the Instructor and TAs. This project should explore a phenomenon (or phenomena) that are only properly or fully analyzable with computational methods, or for which we learn significantly more than with analytic methods alone. The proposed topic must be selected by Tuesday of 5th Week (Oct 29, 2024) in consultation with the instructor and TA, although modifications and changes are certainly allowed, but are strongly encouraged to be discussed first. You are encouraged to include a sketch of the poster (see below), a description of the equations you plan to solve or analyze, and the primary results or even figures that you plan to include.

- **Structure:** The projects are individual and are not allowed to be done jointly with others. However, you are strongly encouraged to discuss approaches, methods, ideas, implementations, and results with any and everyone.
- **Poster Presentation:** The project will be presented in a poster session at the end of the term on the Friday of Reading Period. All posters will be hung simultaneously and audience members will be circulating to review the posters. Your poster will need to be printed the day **before** at the very latest. The project's poster will be evaluated by at least 2 judges (including the TAs and instructor). The criteria and rubric will be distributed in advance to the class. In addition, the GitHub repository must be indicated on the poster, and it will be tested separately for its ability to be executed, as well as for its design and implementation of the key computational methods used.

#### Technical resources for python, git, and jupyter

- **General:**
  - <https://wiki.uchicago.edu/display/phylabs/An+Introduction+to+Python>
- **Python:**
  - Installing: <https://www.anaconda.com/distribution/>
  - Tutorial: <https://docs.python.org/3/tutorial/index.html>
  - Tutorial: <https://www.w3schools.com/python/>
- **Jupyter:**
  - Installing: <https://jupyter.readthedocs.io/en/latest/install.html>
  - Tutorial: [https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/examples\\_index.html](https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/examples_index.html)
  - Tutorial: <https://www.dataquest.io/blog/jupyter-notebook-tutorial/>
  - Tutorial: <https://realpython.com/jupyter-notebook-introduction/>
- **Git:**
  - Installing: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
  - Tutorial: <https://guides.github.com/activities/hello-world/>
  - Tutorial: <https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners>
  - Tutorial: <http://marklodato.github.io/visual-git-guide/index-en.html>
  - Tutorial: <https://medium.com/@abhishekj/an-intro-to-git-and-github-1a0e2c7e3a2f>