

PHYSICS 250: COMPUTATIONAL PHYSICS

RESPONSIBLE LLM USE

Instructor: David W. Miller, MCP 245,
773-702-7671, David.W.Miller@uchicago.edu

Introduction

Large Language Models (LLMs) such as ChatGPT, Copilot, and others are powerful new tools for coding, debugging, and learning. I use them in a number of ways – as *tools* – to augment and improve my (long established) approaches to doing science.

The use of these tools does not run counter to the goals of this course (see the *Learning Goals* companion document, and thus that are certainly not prohibited and you are welcome to use them. In fact, part of how this course is evolving is in trying to build in guidance for helping you *learn* how to use them. Like any tool, and especially new ones, you must do so intentionally and responsibly. You are not here simply to write some code here and there, but rather to build your own ability to *think computationally about physics*.

To that end, I have written this document (with a bit of back and forth with ChatGPT...I certainly did not simply copy and paste) which outlines my guidance for how to use these new tools to your advantage, while not compromising your growth and evolution as a thinker and a scientist. To my mind, and as described in the slides and the *Learning Goals*, Computational physics is about identifying the right set of (computational) tools for a problem (of which LLMs can be a member!), designing algorithms to solve those problems, establishing an understanding why a method works (or doesn't) and connecting those computations back to physical insights.

An LLM can provide examples or guidance, but it cannot replace the reasoning skills that you must develop as a physicist.

Guidelines for responsible usage

- Think before you ask
 - Write down your own plan, pseudocode, or approach before consulting an LLM.

- Use the LLM to refine or stress-test your thinking, not to generate it from scratch.
- Interrogate, don't copy/paste
 - Treat LLM responses as suggestions.
 - Ask yourself *“Does this make sense? Can I explain why? Would I have predicted this step?”*
 - Re-implement code in your own style rather than copying directly.
- Ask “Why?” and “How?” yourself
 - Prioritize questions that deepen understanding
 - * “Why is my Runge-Kutta solution unstable?”
 - * “What’s the trade-off between Euler and Verlet methods?”
 - Avoid *“give me the code for X”* without context and guidance from your own thought process
- Document your LLM usage
 - In your solutions, include a short note on *how* you used an LLM.
 - * *‘I asked ChatGPT why my finite-difference solver produced oscillations. It suggested checking boundary conditions, which helped me identify a missing condition in my code.’*
- Use LLMs as a critical “reviewer”
 - Try asking an LLM to critique your pseudocode or find potential bugs
 - Compare its output to your own implementation
- Maintain a Computational Logbook
 - Keep track of your problem-solving steps, including when you used AI tools.
 - This will help you explain your reasoning later, and is also a valuable professional skill

What counts as your work

- Your grade will reflect your ability to explain and justify what you did. We will have in-class dialogues and explanations of approaches. This quarter you should expect to be asked to:
- Annotate your algorithms with explanations
- Walk through your code verbally or in writing
- Be able to write your own pseudocode in quasi-real-time

Copy-pasted LLM output without understanding is not acceptable and will not prepare you for research, or even your final project.

Conclusions

LLMs can accelerate your learning, but only if you use them with awareness, intentionality, and openness. In this course, you are not being trained to be better at prompting; you are being trained to think computationally about physical problems. Use these tools to sharpen your skills, not to replace them.