PHYS 598 CPA: Computational Physics
Fall 2024
222 Loomis Lab MW 13:30-14:50

Course web page URL
http://rainman.astro.illinois.edu/compphys
login: compphys   password: !ocow!

Professor: Charles F. Gammie
Loomis Lab Room 235, Astronomy Bldg. Room 202
gammie@illinois.edu
Office Hours: 2-3pm Tu and by appointment

Teaching Assistant: Abhishek Joshi
Loomis Lab Room 241
avjoshi2@illinois.edu
Office Hours: 5-6pm W and by appointment

Course Description

This is a computational methods in physics course, a computational analog of the mathematical methods in physics courses you may have taken in the past. We will derive and implement algorithms that produce approximate numerical solutions to analytically intractable physical problems. You should leave this course with a basic algorithmic toolkit and set of implementations that you can build on to do interesting physics.

We will cover: numerical differentiation, integration, and solution of ordinary differential equations, and convergence, truncation error, and roundoff error; methods for solution of hyperbolic, elliptic, and parabolic differential equations; Monte Carlo techniques; aspects of time series analysis, statistical models, and elementary aspects of machine learning. We will also cover a few topics of current interest and topics relevant to the research of enrolled students.

Prerequisites

The prerequisites for this course are knowledge of programming (CS 105 or equivalent) and knowledge of physics at the advanced undergraduate level (PHYS 325, PHYS 427, PHYS 435, PHYS 486, or equivalent), or permission of instructor.

Learning Objectives

1. Students should understand the origin of errors in numerical results, for example in numerical integration, and be able to estimate the magnitude and scaling of errors.
2. Students should be able to identify and implement tests of numerical methods using relevant, analytically tractable problems.
3. Students should leave with a basic numerical toolkit that they can describe physically, algorithmically, and programmatically.
4. Students should be able to estimate the computational resources required for a calculation.

Course Requirements

You are expected to attend lecture and participate actively.

There will be a problem set every week. The problem set workload is high, and is designed to expose you to the kind of computational problem solving that occurs in a research environment.

Final Project

There will also be a final project on a topic selected in consultation with Prof. Gammie. The topic should be related to your research or be related to a problem in your intended field of research. This is an opportunity to develop analysis tools in your field!

The final project should require a time investment similar to a problem set. It consists of a writeup that must include (1) a statement of the physical problem; (2) a description of the algorithm used to solve it; (3) an implementation of the algorithm (code, submitted separately); (4) a description of how you tested the code; (5) an estimate of the computational resources required. You must meet with Prof. Gammie to discuss the final project before the end of September, submit a proposal before the end of October, and submit the writeup on the last day of instruction in early December.

Required and Suggested Resources

There is no required text for the class.

Numerical Recipes, by Press et al., is a readable and reliable introduction to numerical methods. I recommend buying it. If you do not buy it, then you may want to consult the online second editions that are freely available at http://numerical.recipes/.

I will provide additional suggested resources in the lecture notes as the semester progresses.

You will need access to a computer with C/C++, Python, and Mathematica installed. Please come talk to me if you do not have access to a suitable machine. Please ask the TA or me for help if you do not understand what is required.

I recommend installing the gnu compiler collection on your machine, which is freely available at https://gcc.gnu.org.

Most machines come with a python installation. I recommend, however, installing anaconda in your home directory/folder. It is freely available at https://www.anaconda.com/.

As a UIUC student you can get a free copy of Mathematica for Students or use the online version. Instructions for obtaining Mathematica can be found at https://webstore.illinois.edu.

Programming and bug-finding can be frustrating, especially in your first programming-intensive course. You are encouraged to learn to use all the same tools that you would use for a research programming problem: ChatGPT (https://chatgpt.com/), copilot (https://copilot.microsoft.com/, you can log in with your Illinois ID and there are also plugins for most major editors), and whatever else you find useful.

Academic Integrity and Collaborative Work

Academic honesty is essential to this course and to the University. Any instance of academic dishonesty (including but not limited to cheating, plagiarism, falsification of data, and alteration of grade) will be documented in the student's academic file. In addition, the particular problem set will be given a zero.

Guidelines for collaborative work: Discussing course material with your classmates is encouraged, but each student is expected to do his or her own work. On problem sets, you may discuss the questions and issues behind them, but you are responsible for your own answers.

Guidelines for using machine learning: You are encouraged to use ML/AI to complete your problem sets unless you are explicitly asked not to. *Of course* you can you ML! It is a great tool to have in your toolbox.

Accessibility Statement

To insure that disability-related concerns are properly addressed from the beginning, students with disabilities who require reasonable accommodations to participate in this class are asked to see the instructor as soon as possible.

Course Grading

Course grades will be based **on problem sets**, each weighted equally, and on a **final project**, which will have twice the weight of a problem set.   There are no exams.

The 12 problem sets are each worth 100 points and the final project is worth 200 points, for a total of 1400 points.   Expressed as a percentage of available points the grade cutoffs will not exceed:

A+	97
A	93
A-	90
B+	87
B	83
B-	80
C+	77
C	73
C-	70
D+	67
D	63

Topics and Important Dates

This is a brief list of important dates and the tentative subject of each problem set.

Aug 26, 2024: First day of Instruction

Aug 26: Lecture 1

Aug 30 (Fri): PS 1 due (Numerical Differentiation)

Sep 6: PS 2 due (Root Finding and Minimization)

Sep 13: PS 3 due (Numerical Integration / ODEs I)

Sep 16: Gammie away

Sep 20: PS 4 due (ODEs II). Deadline for students meeting with Prof. Gammie to discuss final project.

Sep 27: PS 5 due (ODEs III)

Sep 30: Last day to discuss topic of final project with Prof. Gammie

Oct 4: PS 6 due (PDEs I; Hyperbolic Equations)

Oct 11: PS 7 due (PDEs II; Hyperbolic Equations)

Oct 18: PS 8 due (PDEs III, Multidimensional Hyperbolic Equations)
(warning: right after spring break!)

Oct 25: PS 9 due (PDEs IV: Multidimensional Elliptic Equations)

Oct 30: Gammie away

Nov 1: PS 10 due (PDEs V: Multidimensional Elliptic Equations, Monte Carlo methods). Deadline for final project outline.

Nov 8: PS 11 due (Monte Carlo methods; Ising Problem)

Nov 18: PS 12 due (Time Series Analysis; Fitting Models to Data)

Nov 23-Dec 1: Fall Break

Nov 31: PS 13 due (Stochastic PDEs)

Dec 6: PS 14 due (Deep Learning)

Dec 11: Last Lecture

Dec 11: Final day of instruction. Final project due.