Computational Methods

Physics 367

September 1st, 2026

Joel Franklin
Physics 232
(503) 777-7249
jfrankli@reed.edu

Lecture and Office Hours
Lecture meets on Tu-Th from 10:30–11:50 a.m. in Physics 240a.

Office hours will be held on M 3–4 p.m., Tu, W 1–2 p.m., Th 3–5 p.m.

Text
Joel Franklin, "Computational Methods for Physics," Cambridge University Press, 2013.

additional texts for reference (these can be found in course reserves at the library):

Nicholas Giordano, "Computational Physics," Prentice Hall, 1997.

William H. Press, Saul A. Teukolsky, William T. Vetterling & Brian P. Flannery, "Numerical Recipes in C++: The Art of Scientific Computing," 2nd edition, Cambridge University Press, 2002.

Website: (course)
`http://people.reed.edu/~jfrankli/Courses/P367.F26`

(examples and data)
`https://people.reed.edu/~jfrankli/downloads-2/`

This is a course on the computation of relevant information for physics. For each topic, we will look at a number of physical (and physically inspired) problems that require numerical solution, and discuss the theory and implementation of that solution. In some ways, this class is a specialized "mathematical methods" course, in which the mathematical methods of interest are computationally intensive.

As a course taught from within the physics department, we will be careful to separate the three elements that we must cover: 1. Physics (that motivates our interest in a particular method), 2. The idea behind the numerical method, and 3. The rendering of that method into a particular computer language.

Evaluation

Each class meeting will be accompanied by a reading assignment, posted on the course website. Every week, I will put a homework assignment on the website by Friday at 5 pm, and these are due by the following Friday at noon (homework submission via gradescope).

In addition to homework, I will assign "lab problems" with numerical answers (roughly) every other Friday. The lab problems are where you will develop the codebase and skills needed for the two in-class, hour-long midterm examinations. The lab problems are not due every week, instead you will pick two of them over the course of the semester, and write them up as "presentation problems" (due by Friday at noon, two weeks after the lab problems are assigned, sent to me via e-mail). Your writeups should include:

- A clear statement of the problem in your own words, and with your own emphasis. (20%)
- The problem's connection to work we have done in class. (20%)
- A clear and correct solution to the problem. (40%)
- Possible extensions or implications stemming from your solution. (20%)

and be no more than three pages long.

The midterms will be on October 15th and December 1st. They will consist of three to five problems, with some similar to the ones assigned as "lab problems." You will use the code you have developed over the course of the semester to solve these problems locally on your computer (i.e. without accessing the internet).

There will be a final project – this could be based on your thesis, summer work, or extensions of elements from your classes. The presentation of your project will occur during finals week, when we will have a class-wide poster session. Project proposals will be due the week after fall break (on October 24th), and we'll discuss their format, and the poster presentation itself at the appropriate times. There will be short, 5 minute in-class presentations of your problem statements/solution methods (to make the poster experience more engaging) near the end of the semester.

The scoring break down is:

Homework	15%
Lab Problem (best of 2)	15%
Midterm Exam ($10/14$)	15%
Midterm Exam ($11/23$)	15%
Poster presentation (proposal + talk + poster)	$10 + 10 + 20\%$

LLM models and other forms of AI that produce effectively uncitable (and, in many cases, useless) information may not be used for any component of this class.

Course Targets

By the end of the course, students should be able to:

- Identify a numerical method that could reliably solve, exactly, or in approximation, problems from: Electricity and Magnetism, Quantum Mechanics, Classical Mechanics.
- Understand how to translate a well-posed physical problem into a form for numerical evaluation (examples of this sort of translation include the process of rendering equations dimensionless, and encoding general, vector-valued ordinary differential equations into first order form).
- Be able to correctly implement (including trouble-shooting) the numerical methods discussed in the course, and use them reliably, with an understanding of their strengths and limitations.
- Translate numerical results into physically meaningful output, and communicate those results efficiently in both written and verbal form.

Week	Date	Topic
1	9/1	Computation. Why? How?
	9/3	ODE motivators
2	9/8	
	9/10	Discretization, Stability, Accuracy
3	9/15	Verlet
	9/17	Runge-Kutta
4	9/22	Root finding motivators
	9/24	bisection, Newton
5	9/29	Integration motivators
	10/1	Simpson's method, error estimates
6	10/6	PDE I motivators
	10/8	Lax-Friedrichs, weak solutions
7	10/13	Crank-Nicholson
	10/15	Midterm I
	Fall Break	
8	10/27	PDE II motivators
	10/29	Finite difference
9	11/3	Eigenvalue problem
	11/5	Project proposal due
10	11/10	Fast Fourier Transform
	11/12	
11	11/17	Chaos
	11/19	
12	11/24	Artificial Neural Networks
	11/26	**Thanksgiving**
13	12/1	Midterm II
	12/3	
14	12/8	