Scientific Computing for Physical Systems

Physics 3310, Weber State University, fall semester 2026

Instructor: Dr. Daniel Schroeder
Office: TY 322
Email: dschroeder@weber.edu (best way to reach me)
Phone: 801-626-6048 (voicemail is rarely checked; please use email)
Course web sites: scores in the WSU Online (Canvas) system; documents and links at `https://physics.weber.edu/schroeder/scicomp/`
Office hours: MWF 10:30–11:30 (in my office) and M 1:30–2:30 (in our classroom). I will often be available at other times on MWF as well, though rarely on TTh. Feel free to request an appointment if you would like to nail down a time to meet.

What is this course about?

Introductory physics courses are full of simplifications: projectiles fly without air resistance, pendulums swing only at small angles, orbits are always circular, and no more than two particles move at any time. These kinds of simplifications are necessary and appropriate when you're first trying to understand the basic laws of nature. But the real world is far more complex, and far more interesting. Because the ultimate goal of physics is to understand the real world, students deserve a course that applies the laws of physics to more complex situations.

Fortunately, modern electronic computers make it possible to perform extremely lengthy calculations in a negligible amount of time. These days, therefore, computers offer the best avenue toward applying the basic laws of nature to complex and realistic physical systems. A computer program that models the behavior of a physical system is called a *computer simulation*. Creating and using computer simulations is an integral part of modern science and engineering.

This course is a hands-on introduction to computer simulations of physical systems, using the Python programming language. The goals of the course are as follows:

- Learn enough of the Python language and the VPython graphics package to write programs that do numerical calculations with graphical output;
- Learn some step-by-step procedures for doing mathematical calculations (such as solving differential equations) on a computer;
- Gain a better understanding of Newton's laws and other physical principles;
- Study a variety of physical systems that are too complex for simple pencil-and-paper calculations, and see what sorts of behavior emerge in such systems.

Prerequisites

Before taking this course you should have completed a semester of introductory physics, covering Newton's laws of motion, conservation principles, and a bit of thermodynamics. You should also have taken at least one semester of calculus. Prior expertise with the

Python language is *not* required, but you should be fairly comfortable using a web browser, word processor, and spreadsheet, and you should have some experience at being careful with computer syntax (in any programming language).

Required materials

Naturally, you'll need a computer. We will use a cloud-based version of Python called Web VPython, which requires only an internet connection and a modern web browser. Occasionally you'll need to use a spreadsheet program. But you won't need to install any special-purpose software. If you don't have a laptop computer of your own for use in this course, you are welcome to use one of the laptops that lives in our classroom. (A tablet device without a physical keyboard is not adequate for this course.)

From time to time you will need a scientific calculator. A calculator phone app would work most of the time, but you won't be allowed to use your phone during tests, so I suggest using a stand-alone calculator for everything. The best calculator is the one you're most comfortable using. If that's a graphing calculator, fine—but I personally prefer a simpler scientific calculator like the TI-30 series.

Finally, you'll need a few low-tech materials such as scratch paper, pencils, and a ruler. A three-ring binder will be useful for holding your project instructions and completed reports.

Meeting time and location

MWF 12:30–1:20 pm, in TY 127. We also have a scheduled "lab" session on Friday from 1:30–2:20 pm. Because this is a face-to-face class, I'll expect you to attend at all of these scheduled times. If you cannot attend due to illness or other emergency, please contact me and your lab partner immediately to make appropriate arrangements. (Read on for more about what to do in case of illness.)

Although it is not required, I also encourage you to stay and keep working during the 1:30 hour on Mondays. I will stay in the room during that hour as long as at least one of you is also present. I would adopt the same policy for Wednesdays, but on Wednesdays we will usually have Physics Department seminars at 1:30, and I don't want to discourage you from attending those.

We will not meet on official school holidays: Monday, September 7 (Labor Day); Friday, October 9 (fall break); and Friday, November 27 (Thanksgiving break).

Room policies

You may use your own laptop computer or help yourself to one of the laptops stored in the cart. These laptops may not be taken from the room. When you're finished with one of these laptops, please return it to the cart and plug in its power cord. You may also use the computers that live under each table in the room (connected to the big screens). Don't rely on saving documents on any WSU-owned computers; use cloud storage or a USB stick for backup. Don't use any of these computers for non-academic purposes.

The printer in TY 127 is for printing graphics to include in your lab reports for this class. Don't use it for any other purpose without permission. Please avoid printing images with dark backgrounds that would waste toner.

To keep the electronic equipment clean, please don't eat any messy snacks or drink anything other than water while in the room. You're welcome to step out of the room briefly for a snack or for other reasons.

Lab partners

I will assign you to work in groups of two or three, with different groupings for each project. In some cases you will still write your own individual code, comparing notes with your lab partner(s) from time to time, with each helping the other as needed (but not copying actual code or exercise answers). In other cases I will ask you to work together on the same computer and turn in a single program (or a single set of programs) that represents your joint effort. Even then, though, you will each turn in separate lab reports. You will share some responsibility for your lab partner's work. If absence or tardiness interferes with your ability to collaborate effectively with your lab partner, your project score will suffer. Please don't work ahead in a way that would make effective collaboration with your lab partner difficult. If you nevertheless finish a project early, I'll still expect you to attend class and work with your lab partner until they are also finished.

If you have any concerns about working with a particular individual as a lab partner, please contact me privately.

Tentative course outline

The course will be organized into projects. Tentatively, these will be:

1. Making Shapes (introduction to VPython)
2. Projectile Motion (including air resistance)
3. Pendulum (including large angles and chaotic driven motion)
4. Orbits (including Kepler's laws and interactions between planets)
5. Molecular Dynamics (hundreds of mutually interacting particles; phase behavior)
6. Random Processes (modeling simple systems that behave randomly)
7. The Ising Model (random behavior of a model magnetic system)

We'll spend about one week on the first project and about two weeks on each of the others. Exact due dates will be announced well in advance. For most of the projects you will also submit some preliminary work when you are about half-finished. Late submissions will be subject to a penalty of 10% per day, not counting weekends or school holidays.

We will have **three midterm tests**, one after each even-numbered project. The tests will be given in class, using pencil and paper, with no use of a computer, notes, or other course materials. You may use a calculator to do arithmetic, but not to store information. Test dates will be announced well in advance.

We will also have a comprehensive final exam, similar in format to the midterms but longer, given in our classroom during our scheduled final exam period on Wednesday, December 9, from 12:30 until 2:20 pm.

The tests will cover a mix of coding syntax, numerical algorithms, and physics concepts and results. I hope they will be fairly easy as long as you have engaged fully with every issue that arises in the projects.

How to do the projects

For each project in this class you will write a computer program or (more often) a small number of closely related computer programs. Rather than giving you complete programs to run, the project instructions will provide only code fragments and general guidelines on how to write your programs. This way, once you have completed each program, it will be yours.

As you create your computer programs, you will inevitably have questions and encounter difficulties. While you don't want to spend too much time being stuck on something, it is also essential that you spend time wrestling with tasks that are difficult. The trick is finding the right balance.

When you can't figure something out in a reasonable amount of time, the next thing to do is to *talk to someone* about it: first your lab partner, then (if necessary) another classmate or your instructor. We can all learn from each other, and discussing our difficulties is an essential part of the learning process!

What you should *not* do is seek help via the internet: no web searches, no browsing message boards, no watching instructional videos, and absolutely no queries to AI chatbots or AI coding tools. All such activity is absolutely prohibited in this course and will be treated as academic dishonesty, with all the associated sanctions and consequences for your academic career.

Similarly, you may not seek help from human experts (other than your instructor) who have already mastered large parts of the material being taught in this course.

So what's wrong with using AI coding tools (and other outside sources of help)? Don't professional programmers use these tools?

Here's an analogy. A forklift is a powerful and useful tool, used by professionals who work in warehouses and similar settings. But you wouldn't bring a forklift to the gym to help you lift the weights. Why not? Because that's not why people go to the gym. They go to the gym to build their physical strength; lifting the weights, without a forklift, is an effective way to accomplish that. Using a forklift would defeat the purpose.

Similarly, I assume you're in this course because you want to build certain aspects of your mental strength—to make yourself a stronger and better person. Enlisting an AI tool (or an outside human expert) to do the heavy lifting for you would defeat the purpose of the course.

I hasten to add that there may be other courses on this campus where the use of AI coding tools is complete appropriate. There may even be courses in how to use AI coding tools. There may also be a course that teaches you how to operate a forklift. But this isn't that course.

Exercises and questions will be sprinkled throughout your project instructions, with space for you to write your answers. Please make every effort to work each exercise and answer each question immediately, before you read on.

As you'll see, the philosophy of the project instructions is often that you'll learn more by *trying* something than by reading a comprehensive *explanation* of it. Computer languages are like ordinary languages in this respect: We normally learn new words by hearing, reading, and using them in context, not by studying a dictionary. But when something seems unclear even after you've seen it in context, be sure to ask your instructor for a fuller explanation.

Computer programming is fun because it's so open-ended. You'll constantly think of things to try that go beyond the explicit instructions. By all means, try anything you want! If you're not sure how to add a certain feature to one of your simulations, or if you're not sure whether it's practical to do so within a limited amount of time, be sure to ask your instructor.

When you finish a project, gather the instruction pages and staple them together with any printed output from your programs. This stapled packet, together with the source code of your computer programs, will be your "lab report."

What this course is not

This course is not a comprehensive introduction to the Python programming language. Many features of the language are not needed for the types of simulations we'll be doing, so we'll ignore them. Several other features will be used once or twice but never fully explained.

I've tried to design the examples in this course to illustrate good programming practices that are appropriate to the relatively small scale of the projects. This is not a course on professional software design.

This course is not a survey of numerical analysis or numerical algorithms. You'll get a good taste of the subject, but for a broader view you may want to take the numerical analysis courses offered in the Math Department.

The projects in this course will touch on some fascinating fields of physics, including nonlinear dynamics, celestial mechanics, and phase transformations. But we won't dwell on any of these physics subjects, so again you might want to follow up with further coursework.

Project grading standards

In grading your projects I will consider the following:

- Completeness and correctness of your programs (approximately 40%)
- Clarity and organization of your code (approximately 10%)
- Written exercise responses (approximately 40%)
- Effectively working with your lab partner (approximately 10%)

Note that the percentages are approximate, and will vary somewhat from one project to another due to differences in the projects. I reserve the right to modify these percentages significantly to handle unusual situations. I may also grant some extra credit, if you go beyond the required work for a project.

Final grades

Grades will be computed according to the following weights:

Projects	55%
Midterm tests, 3 @10%	30%
Final exam	15%

The projects will be weighted equally, except for the first one ("Making Shapes"), which will have half the weight of the others.

Keeping in touch

Please check your WSU email account regularly for announcements related to this course. In case of a snow day or other campus emergency or closure, check your email as soon as you can for possible changes to due dates or test dates.

In case of illness

Please take reasonable precautions to avoid spreading Covid-19 or any other contagious disease to the rest of us and our families. Should you become sick or need to isolate due to the likelihood of being contagious, contact me and your lab partner(s) immediately so we can make a plan for you to continue your work remotely.

Give some thought in advance to how you could work from home in case of illness. If you do not have a home computer and internet connection that would be adequate for this use, please let me know during the first week of the semester.

Miscellaneous policies

The penalty for turning in work that is not your own, or for other violations of the WSU Student Code, will range from failure on the assignment to failure in the course. Severe cases may also be referred to a higher-level hearing committee or the WSU Dean of Students for further sanctions.

The WSU Student Code contains several provisions that essentially require you to treat your fellow students with respect, both in and out of the classroom. Inappropriate behavior toward other students will not be tolerated and will be reported to appropriate authorities for possible sanctions.

Any student requiring accommodations or services due to a disability must contact the Disability Services office in room 181 of the Student Service Center. Disability Services can also arrange to provide course materials (including this document) in alternative formats if necessary.