Physics 315
Computational Physics
Spring 2024

Brad Hinaus
Office: B107 SCI
bhinaus@uwsp.edu

Th B112 12:30-1:45 B104 Science
Lab: W 3-5:50 B104 Science

Final Exam Time:

PHYS 315-01 (40453)	Computational Physics (Lecture)	5/14/2024, Tuesday	8:00AM - 10:00AM	Science Building (SCI) B104	7

Office: B107 Science
Phone: 715-254-5141 (cell)
Zoom Address:
Email: bhinaus@uwsp.edu

Student Hours:
MWF 10-11
Tu 10-11
Th 10-11 -

Office Hours – You may also make an appointment (via text) to come to office hours or show up at the listed times above. If I am not in my office, feel free to text me at 715-254-5141 to let me know you are waiting or if you need to make an appointment.

Textbook from Text Rental: *Computational Physics* by Nicholas Giordano
Handouts during class

What is Computational Physics?

Computational Physics is a branch of physics, which uses computers to solve physics problems. Physicists use the computer to *simulate* complex physics situations. These simulations can be used to *model* experimental data to determine the physics that applies to the data to determine the appropriate physics in an experiment, or it can be used in reverse, to predict outcomes if certain physical concepts are included. The computer can also be used in a pure *numerical* sense to compute an integral or a derivative, for example. In any case, the focus of computational physics is to gain an insight into the physics involved in a scenario using a computer, not necessarily on writing elegant compact code.

Programming Software

In this course, we are going to learn how to write computer simulations using Python. 1.) The syntax (how commands are typed) is relatively straightforward. 2.) Once you know a little Python, you can quickly pick up another language. 3.) Python is Open Source and Free. We will download it via Anaconda Navigator. Python has other extensions, such as PyGame, which allows one to make a video game, VPython which allows for simple 3D simulations.

Learning Outcomes

Learning Outcomes for Computational Physics Course:

When you finish this course, you should be able to do the following. Program using a high-level programming language.

- Solve physics problems using computational methods that solve differential equations, perform integration, or use stochastic methods to study random systems.
- Model the physics in various systems starting with the basic physics and solve the model with the appropriate computational techniques.
- Analyze output data for correctness, by making a plausibility argument, an analytic calculation for a limiting case, or an order of magnitude calculation based upon a simplifying assumption.

Learning Outcomes for Physics Majors:

- Integrate conceptual reasoning, critical thinking skills, mathematical skills, and principles from both theoretical and applied physics courses to explain and solve problems related to the physical processes in nature, applied mechanics, applied electronics, and those appropriate for the education setting. .

- Investigate a problem experimentally by identifying the problem, developing an appropriate experiment, collecting reliable data, quantitatively analyzing results, determining uncertainties and probable errors, and drawing justifiable conclusions.

- Communicate effectively within the profession by writing clearly and concisely and by articulating clearly.

Learning Outcomes for the General Education Program for Communication in the Major:

- Create an oral presentation that is well organized, informative, and smoothly delivered and analyze other's presentations to provide effective feedback.
- Write various sections of journal manuscripts in the style of physics community based on the computational research performed in class. Analyze professionally written papers in terms or organization, style, and content.

Grading

Letter Range	Percentage
A	93-100
A-	90-92.9
B+	87-89.9
B	83-86.9
B-	80-82.9
C+	77-79.9
C	73-76.9
C-	70-72.9
D	60-69.9
F	0-59.9

You will be graded on the following: homework, papers (lab report and project reports) group projects, presentations, exams and an individual final project.
The final course grades will be weighted as follows:

Exams	40%
Papers and Presentations	30%
Homework/In Class Work	30%

All graded items will receive numerical scores. The adjacent table shows the ranges of percentage points for the final grades in the class.

Late Work Policy: The best policy is to communicate when a student knows of a late assignment, especially presentations and exams. Any assignment (homework) that is more than 7 days beyond the due date (as indicated in Canvas) will receive a reduction of up to 20% of the score at the discretion of the instructor. If students are having difficulties completing the assignments due to factors outside of class (health, family, work, other classes, etc.), they should communicate this to the instructor before the 7-day grace period to avoid the full reduction. If difficulties are coming because of class material, it is suggested that the student **advocate** for themselves and make an appointment with the instructor for further assistance.

Presentations and Exams are to be completed on the due date as outlined in Canvas/Class. The instructor has full discretion if a student is allowed to make up a presentation or exam and the amount of the reduction in the grade.

Accommodations: UWSP is committed to providing reasonable and appropriate accommodations to students with disabilities and temporary impairments. If you have a disability or acquire a condition during the semester where you need assistance, please contact the Disability and Assistive Technology Center on the 6th floor of Albertson Hall (library) as soon as possible. DATC can be reached at 715-346-3365 or DATC@uwsp.edu.

Contents of the Course

Unit 1:

- Programming: Basics of Python, While Loops, For Loops, If statements, Modules, and User Defined Functions
- Numerical: Root Findings, Summations, Max/Min, Numerical Integration
- Math: Analytic Solutions of Differential Equations
- Physics: Equations of Motion for a Pendulum
- Communication: Giving a Presentation

Unit 2:

- Programming: Implementing Python to solve physics problems and program Structure, Initial Value Problems, Boundary Value Problems
- Numerical.: Solving Differential Equations by using the Euler Method or the Euler-Cromer Method
- Physics: One Dimensional Motion and Newton's Laws. Writing Newton's Second Law as a Differential Equation, Multi-Dimensional Motion. Analysis of Numerical Results for Correctness, Heat Equation
- Communication: Writing Analysis and Discussion, Making a Graph and Writing a Caption.

Unit 3:

- Physics: Random Systems: Nuclear Decay and Poison Statistics,1-D Icing Models, Random Walks in 1D
- Programming: Arrays/List, User Defined Functions Again, and Nested Loops.
- Numerical: Random Numbers Metropolis Method,
- Communication: Writing Papers, Writing Introduction and Method and Measurements

Section 4

- Physics: Individual Capstone Projects
- Numerical: User Choice (Something interesting, something new, both in physics or a computation technique)

My Teaching Philosophy

I think the college classroom should reflect basketball practice. Mentally picture what basketball practice looks like. What do you see? Its active, people are moving around and doing things. Players don't spend 100% of their time watching their coach draw diagrams on the chalkboard then go on the floor and walk through the plays. The ball players spend a good portion of their time working on the skills themselves. That is what I want us to do, work on our skills during class *with each other*. Will we eliminate the lecture? No, but I hope to reduce the amount of time in that mode, so we can practice and ask questions. (If basketball doesn't work for you, substitute learning a musical instrument)

<u>Inclusivity Statement</u>

It is my intention that students from all backgrounds are well served in this course. Backgrounds can include gender, race, orientation, age, disability, religion, culture, and other ways a person identifies. Other backgrounds that give students various perspectives of this course are their current mathematical abilities, their developing problem-solving abilities, past courses, life experiences growing up, classroom environments they have experienced, preferred learning style and more. In this course, it is expected that each other's thoughts and comments be respectfully listened to and/or responded to during class, lab, and discussion. There are numerous ways to have a computer solve a problem. It is also expected during the course times that students work to assist each other in the learning process.

To help you with the difficulties of this class, I am available in scheduled office hours and meeting by appointment. In past semesters I have had standing 1-hour meetings with individual students each week. I am welcome to those. If you would like the hear a different perspective, the STEM Drop-in tutoring is available in CBB 190 (see below) If you have suggestions for me on how to make this class more inclusive between instructor/student and student/student interactions or activities, please let me know.

Additional References:

- *Computational Physics: Problem Solving with Computers* by Rubin H. Landua and Manuel J. Paez. Internet Site: http://www.physics.orst.edu/~rubin/CPbook/
- *An Introduction to Computer Simulation Methods: Applications to Physical Systems* by Harvey Gould and Jan Tobochnik
- *Python:* www.python.org

Online Python Tutorial:

- http://www.learnpython.org/ An online tutorial sorted by topic. It teaches a topic, gives an assignment, and within the webpage, it allows a user to type in the code, run it, and see the output in a second window on the web page.
- https://www.codecademy.com/learn/python allows user to type in code and run it within a web page.

Online Vpython Video Tutorials

- http://vpython.org/contents/doc.html see list of video topics and written documentation