Physics 315
Computational Physics
Spring 2021

Brad Hinaus
B 107 SCI
bhinaus@uwsp.edu
Cell Phone 715-254-5141

Lecture: Asynchrones Online via Canvas (2hours 50 minutes)
Lab: Asynchrones Online via Canvas (2hours 50 minutes)

Office Hours

Office Hours will be via Zoom. Use the following link for all office hours or anytime we meet via Zoom.

https://uwsp.zoom.us/j/99652668535?pwd=eVNaNjhxZUtjRUFmTm5PeWFkSEVFZz09
Meeting ID: 996 5266 8535
Passcode: 546109

Monday 1-2:45
Tuesday 9:30 -11:00
Wednesday 1-2:45
Thursday 11:00 -12:45
Friday 10-12
Always by appointment made via email or text. If you hang out in the waiting room too long send me an email or text me.

Text Book 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 VPython. It is based on the Python programming language with a visual module to make 2D graphs or 3D visual simulations. There are numerous reasons why we use VPython. 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. 4.) VPython allows 2D plots from within the program and has simple abilities for 3D simulations. Python has other extensions, such as PyGame, which allows one to make a video game. As a note, we will use more of the numerical aspects of VPython and the 2D graphing capabilities, but skip many the 3D capabilities (although it is not too difficult to learn). The VPython is ready to use at https://www.glowscript.org/ and requires no installation of software. If you would like VPython installed on your computer, you can use the Anaconda package at https://www.anaconda.com/ . Contact me if you are interested in installing it on your computer.

Learning Outcomes from Various Sources

Learning Outcomes for Computational Physics Course:

When you finish this course, you should be able to do the following. See Appendix for more detail

- Program using a high-level programming language using assignment statements, while loops, for loops, if statements, modules, built in functions, and user derfined functions.
- Solve physics problems using computational methods that solve differential equations, perform integration, or use stochastic methods (random numbers)
- Model the physics in various systems starting with 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.
- Communicate effectively in oral presentations, and begin to write in a professional manner.

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

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

Final Individual Project	20%
Exams	40%
Papers and Presentations	20%
Homework/In Class Work	20%

Note: All students who are juniors or seniors and declared as Physics Majors at the beginning of this course will be required to take Major Field Test (MFT) in Physics to assist in the assessment of the department. Details will come later. Failure to take the MFT will result in a failing grade.

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

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

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, Techniques of Numerical Integration (Riemann Sum, Trapezoid, Simpson, and Monte Carlo)
- 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 problem that are Initial Value Problems or 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

Unit 3:

- Physics: Random Systems: Nuclear Decay and Poison Statistics, Monte Carlo Simulations, 1-D Icing Models,
- Physics: Nuclear Decay, Poisson Statistics, Magnetism /Ising Model
- Programming: Arrays and Embedded 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 computation)

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)

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