Physics 315

Computational Physics

Spring 2017

Brad Hinaus Lecture: M 10-1 pm A104 Science

B207 Science Lab: W 10-1 pmA104 Science

[bhinaus@uwsp.edu](mailto:bhinaus@uwsp.edu) Office Hours M,W: 1 am

346-4872 R: 10 am

F: 9-11am

Walk UP if Door Open

**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 or to determine what happens under other experimental conditions. 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 through the use of 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 and 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 skip a lot of the other capabilities that it has.

Learning Outcomes

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.
* Solve physics problems using computational methods that solve differential equations, perform integration, or use stochastic methods (random).
* Model the physics in various complex systems starting with the basic physics and solve the model using 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.
* Create an oral presentation that is well organized, informative, and smoothly delivered.

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

Final Individual Project 20%

Exams 40%

Papers and Presentations 20%

Homework/In Class Work 20%

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

Contents of the Course

Unit 1:

* Programming: Basics of Python, While Loops, For Loops, If statements, Modules, and Methods
* Numerical: Root Findings, Summations, Max/Min, Integration
* Physics: One Dimensional Motion and Newton’s Laws. Writing Newton’s Second Law as a Differential Equation.
* Math: Analytic Solutions of Differential Equations
* Communication: Giving a Presentation

Unit 2:

* Programming: Implementing Python to solve physics problems and program Structure
* Numerical.: Solving Differential Equations by using the Euler Method or the Euler-Cromer Method
* Physics: Multi-Dimensional Motion. Analysis of Numerical Results for Correctness
* Communication: Writing Papers, Writing Introduction and Method and Measurements

Unit 3:

* Physics: Random Systems: Nuclear Decay, Monte Carlo Simulations, 1-D Icing Models. Boundary Value Problems: Heat Equation, Schrodinger Equation – Infinite Square Well Potential
* Programming: Application of the material from the semester.
* Numerical: Metropolis Method, Random Numbers, Shooting Method
* Writing: Analysis and Discussion

Section 4

* Physics: Individual Capstone Projects
* Numerical: User Choice (Something interesting, something new, both in physics and 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](http://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

Appendix A

Learning Outcomes

1. Numerically Solve Different Types of Physics Problems Using Multiple Techniques
   1. Ordinary Differential Equations
      1. Types of Differential Equations
         1. 1st Order Differential Equation
         2. 2nd Order Differential Equations
         3. Differential Equations which need to be written in Component Form
         4. Coupled Differential Equations
      2. Techniques for Solutions of Ordinary Differential Equations
         1. Euler Methods
         2. Euler Cromer Method
         3. Verlet Method
      3. Types of Problems using Differential Equation
         1. Initial Value Problem
         2. Boundary Value Problems
      4. Partial Differential Equations – Heat Equation (1-D)
      5. Final Project
   2. Integrals
      1. Techniques for Solving
         1. Endpoint Rule (Rectangular)
         2. Trapezoid Rule
         3. Simpson’s Rule
         4. Monte Carlo Method
      2. Dealing with Singularities (Divergences)
   3. Stochastic (Random) Processes
      1. Uniform Random Number Generators
      2. Small Sample vs. Large Sample
      3. Monte Carlo Simulations (Ising Model)
2. Understand the Fundamentals of a Programming Languages
   1. Useful Features on Microsoft Excel
      1. Absolute Reference vs. Relative Reference
      2. Variable Naming
      3. Graphing
   2. VPython
      1. Understand the Structure of a Program by
         1. Developing Flow Charts or Algorithms
      2. Correctly use the Syntax of the Programming Language (VPython)
         1. Comment Lines
         2. Arithmetic
         3. Integer Division
         4. Real Division
         5. While Loops
         6. For Loops
         7. If Statements
         8. User Defined Functions/Methods
         9. Type change (i.e. changing an integer number to a real value)
         10. Arrays (1D)
      3. Synthesize the Algorithm and Syntax to Write a Program
      4. Analyze output to ensure validity of output
         1. working from known answers
         2. working in limits
         3. conceptual justifications (not always fool proof)
         4. use of different methods/routines
      5. Debug any incorrectly running program
3. Effective Communication
   1. Write Clearly and Concisely to Communicate the Physics, Methods and Results of a Project
      1. An effective paper will
         1. Be written to the appropriate audience
         2. Briefly Summarize the entire paper (Abstract)
         3. Describe the basics physics of the problem
         4. Explain the appropriate numerical technique being used
         5. Interpret and analyze the data
         6. Present the results in an understandable format i.e. descriptions, tables or graphs
         7. Justify the results are correct by using a conceptual explanation, limiting cases analysis, analytic solution, solution by multiple numerical techniques, etc.
         8. Estimate limits of the solution
         9. Convey an understanding of the broader application of the solution
      2. Papers will include the following Sections
         1. Title
         2. Abstract
         3. Methods and Measurement Section
         4. Results and Discussion
         5. Conclusion
   2. Oral Presentations
      1. An effective talk will have the same elements as an effective paper
      2. Presenters will be
         1. Comfortable
         2. Confident
         3. Professional

Appendix B

Net Force No Name

Comp

In front of the University Center there was a crane that lifted material into the air when it was being remodeled. Let’s say the crane is lifting a box and the box is not on the ground.

a.) Draw the two main vertical forces acting on the box. Label each force with what exerts the force and what feels the force.

Box

b.) The box is moving upward but begins to slow down as it approaches one of the floors where it is needed. As this happens, which of the two forces on the box is bigger, or are they the same. Explain your reasoning.

Thinking In Extremes or in Baby Steps

Jars

Monte Hall Problem