# Physics 210: Introduction to Computational Physics (Fall 2014)

**COURSE HOME PAGE (this page)**: http://laplace.physics.ubc.ca/210/

| | |
|---|---|
| **Instructor**: Matthew (Matt) W. Choptuik | **Office Hours**: Mon & Wed: 2:00-3:00 PM & Drop-in (e-mail appt. preferred) |
| **Office**: Hennings 403 | **Web page:** http://laplace.physics.ubc.ca/People/matt/ |
| **Office Phone**: 604-822-2412 | |
| **E-mail**: choptuik@physics.ubc.ca | **TAs:** Arman Akbarian and Amanda Parker |

**SCHEDULE:**

- **LECTURES: TUESDAY & THURSDAY 12:30-13:30 -- HENNINGS 201**
- **LABS:**
  - **L1A: TUESDAY & THURSDAY 13:30-15:30 -- HENNINGS 205**
  - **L1B: TUESDAY & THURSDAY 15:30-17:30 -- HENNINGS 205**
  - **NOTE:** The computer lab is open 24/7, but the doors to Hennings are locked on the weekends, and from 6:30 PM to 7:00 AM on weekdays**.**

- **Course Announcements will be made through Connect/Blackboard**

**COURSE LINKS**

- **COURSE NOTES**
- **SYLLABUS / SCHEDULE** (Contains links to lab activities)
- **HOMEWORK**
- Online Course Resources
- Course Software Availability for Personal Machines
- Learning Goals & Course Topics
- Suggested Hard Copy References
- Term Projects
  - Term Project Ideas
  - See Course Notes page for sample animations from representative projects
  - Instructor's sample term project proposal [PDF]
  - Previous student project proposals from
    - 2009 [Group 1 PDF | Group 2 PDF]
    - 2012 [L1A PDF | L1B PDF]
    - 2013 [L1A PDF | L1B PDF]
- Student Pages
- PHAS IT Catalogue

### Course Summary

This course will provide an *introduction* to techniques and applications in computational physics. Topics to be covered include: Unix / Linux fundamentals, an / introduction to symbolic & numeric computation and programming with Maple; MATLAB (octave) and MATLAB programming, and specific topics and applications in physics and numerical analysis.

*There will be a significant programming component in most stages of the course.*

See the Syllabus below for a provisional lecture/lab schedule, as well as the Learning Goals & Course Topics page for a more detailed overview.

### Text, Reference Material and Notes

Due in large part to the diversity of topics to be covered, *there is no required text for the course*. However, because much of the course will be MATLAB based, I have adopted the following as an *optional* text:

- MATLAB: An Introduction With Applications, 5th edition, Amos Gilat, John Wiley & Sons (2014)

This book is written at a suitable level for an introductory course, has generally been well-received by students in reviews that I have seen, and should be especially useful if you have little or no experience in MATLAB, and/or little or no experience in computer programming. The UBC bookstore currently has about 15 copies in stock for $112/62 for purchase/rental or $40 for the eBook version. However, earlier versions of the text, including the 2nd, 3rd and 4th editions, will suffice for the course, and you may be able to get those from Amazon etc. at a good discount.

Note that although we will be using MATLAB in this course, there is an open-source (freely available) language called octave, which is close to a clone of MATLAB. Our coverage of MATLAB will be such that what you learn about it (including programming) will also apply to octave.

You should also observe that there is a wealth of online material available about MATLAB (I've accumulated a few links to some key sites in the Online Course Resources page, including a link to a site that provides (for individual use only), a complete text by the author of the first version of MATLAB.

The Course Resources page also contains links to sites relevant to other topics that we will cover in the course. Some of these topics, such as Unix/Linux and basic MATLAB programming, will be directly discussed in lectures or covered in labs. Others, such as the use of a text editor of your choosing, will be self-study topics, since a key goal of this course is to enhance your ability to use help facilities, online resources and the like to master new algorithms and software applications.

Finally, at times I will distribute notes to the class (or at least make them available on-line via the Course Notes page). However, at other times, I will lecture using the blackboard, and then you will be responsible for taking your own notes.

### Computer Access

To participate in this course, you must have a Physics and Astronomy (PHAS) computer account, which will provide you with access to the computers in the PHAS computer lab, Hennings 205, and and use of the machines in that lab should suffice for completion of your homework and projects. If you do not already have an account, you can self-register for one during the first lab (or otherwise as early as possible) in Hennings 205 using the workstation with the "Register Here" sign on it. There is also a workstation in Hennings 203 that can be used for registration.

For information concerning the services provided by the IT section of the dept, please refer to the IT catalogue.

You may also be able to use your laptop/home machine to do some of the class work, especially if you are able and willing to install a Linux distribution (Ubuntu recommended) on it/them. (Note, however, that you will be doing the installation at your own risk; we can not be responsible for the loss of the original operating systems, or of any data on your machine.) However, you should not expect to be able to do all of the coursework remotely, so be prepared to spend some time in the computer lab outside of our regularly scheduled meetings times.

**NOTE:** The computer lab is open 24/7, but the doors to Hennings are locked on the weekends, and from 6:30 PM to 7:00 AM on weekdays.

### Grades: Tests, Homework & Labs, Term Projects and Late Work Policy

**EXTREMELY IMPORTANT!! Please refer to the Homework Page for the course policy on Homework / Term Projects and Academic Dishonesty**

Your final grade in this course will be determined on the basis of your performance on three homework assignments and a term project, with the following weighting:

- Homework Assignments: 60%
- Term Projects (including writeup): 35% (due Wednesday December 5, 9:00 AM)
  - Topic selection and in-class presentation of proposal: 5% (no evaluation; full marks for completion)
  - Project per se: 30%

Final marks *may* be subject to small adjustments based on overall class performance.

#### Tests

There will be **NO** tests or exams in this course.

### Homework and Labs

**Homework**

See the syllabus below for (provisional) scheduled homework due dates. *Homework will be assigned about 2 weeks before it is due; late homework* ***may*** *be accepted at the my discretion, and as per the* ***Late Homework Policy*** *described below.* As the course progresses, the Homework Schedule web page will be updated with information concerning the assignments including the homework handouts themselves.

*Each homework will contribute equal weight to your final mark, but again; the homework component of your mark may be subject to adjustments based on overall class performance. Be warned that many of you will find that the homeworks become significantly more challenging as the course progresses.*

**Labs**

A chief purpose of the labs is to provide you with time to acquire the extremely important "hands on" skills needed to master the course material, and which by nature, is difficult to teach/learn in a traditional lecture setting. Some of the lab sessions will be concerned with specific topics, in which case I will generally provide a set of online notes that we will work through together. For others, you will be have free time to work on your assignments and term projects, assisted as necessary by the TAs, myself, and your classmates. In the early stages of the course, you should also take advantage of the lab time to discuss possible term project ideas with me. Finally, at any time, you should feel free to use available lab time to ask any of us about aspects of the computer work that are giving you trouble.

*Lab work will not be graded.*

### Late Work Policy (Strictly Enforced)

**You are strongly urged to submit your homework by the due date. However, from time to time, and provided that the circumstances are sufficiently extenuating, work may be submitted late, subject to the following conditions:**

1. If an extension is required, the extendee must submit a request for an extension, via e-mail, to the (the instructor), *before the assignment is due.*
2. Submitted homework, which *absolutely must be submitted before the homework key is distributed*, must similarly be accompanied by an e-mail indicating completion of the work.

Note that all messages are to be sent to me, *not the TA*, and that if you finish the homework on time, *no additional action on your part is required*.

Finally note that if you are unable to complete an assignment or term project on time due to illness or an equivalent circumstance (e.g. severe illness and/or death of a family member), please inform me as soon as possible and I will ensure that you are given sufficient time to complete your work once your situation has been resolved.

### Term Projects

Completion of your term project is without a doubt the most important component of your work in PHYS 210. For most of you, it will also be the most challenging aspect of the course.

Please read the following carefully:

Either individually or in consultation with me, each student must choose a topic for a term project in some area of computational physics or closely related field, prepare and present a proposal to the class, carry out the project and produce a write-up of it in the basic style of a scientific/technical paper.

You are encouraged to develop your own project ideas, but *all project topics must be approved by me.* Some possibilities for term projects are posted on the Term Project Ideas page, which may be updated as the course progresses. I expect that many of you will complete a project from one of the suggestions, and there are no restrictions on the number of students who can select any given topic.

Topics for term projects must be chosen no later than October 16, and by that date *each of you must have sent me an e-mail stating what topic has been selected.* During the classes and lab periods on October 21 and 23, each student will give a brief presentation on their proposed project; speaking order will be alphabetical by last name. The amount of time available for each presentation will be approximately 7 minutes, so talks will need to be carefully prepared and efficiently executed. Some form of presentation software, including Powerpoint, must be used to prepare your talk and you must generate a PDF version that you will need to e-mail to one of the TAs in a timely manner so that all of the talks can be assembled into a single set of slides. Details concerning this will be provided later.

**There will no *evaluation* of this aspect of your term projects. However, on-time e-mail submission of your approved topic and your in-class presentation are worth 5% of your grade: i.e provided that you *do* submit your topic and make a presentation you will receive the 5%.**

Note that the main purpose of this exercise is to ensure that you *have* chosen an appropriate topic, and that you have a good (though perhaps not complete) understanding of what will be required to complete it.

In keeping with the spirit of the course, all term projects must involve programming to a significant extent, and you are expected to use MATLAB to implement your project. Implementation of your project should be, as much as possible, "from scratch", i.e. you are expected to do more than use some built in MATLAB facility to perform the bulk of your computations.

Provided that there is sufficient rationale for doing so, you may also use another programming language for your project: if you wish to do this, I ask that you check with me before you start work on your proposal so that I can ensure that use of the language that you propose is well motivated, and that the overall project appears appropriate.

All term projects must be written up in the style of a scientific/technical paper; a typical structure will be:

- Title and Abstract
- Introduction, including basic description of problem to be solved, simulated, analyzed etc.
- Mathematical formulation of the problem as relevant
- Description of techniques, algorithms, analysis tools etc. used to solve the problem, including discussion of overall flow of the program
- Discussion of computations (numerical experiments) that were performed
- Analysis of results
- Conclusions (may include suggestions for future work)
- References / Bibliography
- Appendix including program listing (program code), if you wish

Note that for some projects not all of the above sections will be relevant; as always, feel free to check with me should you have any questions about your writeup. I will also ask you to make any programs that you write for your term project available to me through your homework directories on your PHAS accounts, and in all cases I must be able to run your programs on my own PHAS account. If there is any concern on your part about this last point, please talk to me about it *before* you start your your project.

The suggested paper length is *about* 15-20 pages, double spaced (please!), including title page, figures and graphs and references. If you include program listings (code), they should be single spaced. You are encouraged to use the LaTeX typesetting system to write your paper, but this is not mandatory.

As noted above, the term project itself, including the write-up, is worth 35% of your final mark, 30% if you exclude the essentially automatic 5% that you will receive for e-mailing me your approved topic choice and orally presenting your propose. Factors that will be taken into account in my grading will include (but are not necessarily limited to): scope and difficulty of the problem, degree to which the project was completed successfully, effort devoted to the project, originality, and completeness and quality of the written report.

**Your written report and the source code for your project are due by Wednesday December 3, 9:00 AM, except under very extenuating circumstances. You must deliver a hardcopy of your report to my office by that time: i.e. electronic submissions will not be accepted.**

**IMPORTANT!!** You should note that completing a good term project is *much* different than finishing a homework, or even a few homeworks: in particular, it is virtually impossible to do a decent job on a term project in the space of a few days. It is the nature of computational physics (as in experimental physics and in many other pursuits) that things *will* go wrong unexpectedly, and it can often take much more time than anticipated to get programs to work. Moreover, coding a functional program is typically just the first stage in completion of the project; you also will need time to generate and analyze results, as well as to write things up.

**IMPORTANT!!** Note that projects will be graded **rigorously**, and that doing well in the homeworks will not automatically guarantee that you do similarly well with your project.

In summary then, please take your term projects very seriously, and do your best to begin work on them as soon as is feasible.

Finally, be sure that you understand and abide by the University and course policies concerning Academic Honesty as they pertain to your term projects, and as are laid out in the Homework page.

### Other Help

You should also feel free to contact me via e-mail (preferred) or phone if you have quick questions, or if you are having difficulty getting something to work.

Perhaps most importantly, you should strive to develop the ability to make effective use of the available documentation for the software you are using (on-line help, man pages, Web resources, etc.). As you are all aware, the amount of information online, combined with the power of search engines such as Google, provides a powerful resource for self-education on a broad range of topics. This is especially true for computer-related subjects.

## SYLLABUS / SCHEDULE

| Tuesday | Thursday |
|---|---|
| | *September 4*<br>Course Overview & Unix/Linux<br>*Introduction to Computer Lab, account configuration* |
| *September 9*<br>Unix [SOTD Web \| SOTD 1 \| SOTD 2]<br>*Unix Lab 2* | *September 11*<br>Unix<br>*Unix Lab 3* |
| *September 16*<br>Unix<br>*Unix Lab 4, Free time* | *September 18*<br>Overview of Programming / Maple<br>*Maple Lab 1* |
| *September 23*<br>Maple<br>*Maple Lab 2* | *September 25*<br>Maple Programming [**HW1 due**]<br>*Maple Programming Lab 1* |
| *September 30*<br>Term Projects / Maple Programming<br>*Maple Programming Lab 2* | *October 2*<br>Maple Programming<br>*Maple Programming Lab 3* |
| *October 7*<br>Finite Difference Approximation<br>*MATLAB Lab 1* | *October 9*<br>Finite Difference Approximation<br>*MATLAB Lab 2* |
| *October 14*<br>Finite Difference Approximation<br>*Overview of MATLAB Programming*<br>*MATLAB Lab 3 (Programming)* | *October 16*<br>Finite difference Approximation [**HW2 due**]<br>*MATLAB Lab 4 (Programming)*<br>[**Term project topics must be chosen!!**] |
| *October 21*<br>12:30-13:30: Project Proposal Presentations 1, L1A<br>13:30-15:30: Project Proposal Presentations 2, L1A<br>15:30-17:30: Project Proposal Presentations 1, L1B<br>Note: All presentations in Computer Lab | *October 23*<br>12:30-13:30: Project Proposal Presentations 2, L1B<br>13:30-15:30: Project Proposal Presentations 3, L1A<br>15:30-17:30: Project Proposal Presentations 3, L1B<br>Note: All presentations in Computer Lab |
| *October 28*<br>Finite Difference Approximation (Pendulum)<br>*MATLAB Lab 5 (Programming)*<br>*Free time* | *October 30*<br>Finite Difference Approximation (Pendulum)<br>*Nonlinear pendulum 1*<br>*Free time to work on homework / projects* |
| *November 4*<br>Finite Difference Approximation (N-body)<br>*Nonlinear pendulum 2*<br>*Free time to work on homework / projects* | *November 6*<br>Finite Difference Approximation (N-body)<br>*Cellular automata and visualization utilities*<br>*Free time to work on homework / projects* |
| *November 11* [**HW3 due**]<br>Free time to work on projects (L1A)<br>*Free time to work on homework / projects* | *November 13*<br>Free time to work on projects (L1B)<br>*Free time to work on projects* |
| *November 18*<br>Free time to work on projects (L1A)<br>*Free time to work on projects* | *November 20*<br>Free time to work on projects (L1B)<br>*Free time to work on projects* |
| *November 25*<br>Free time to work on projects (L1A)<br>*Free time to work on projects* | *November 27*<br>Free time to work on projects (L1B)<br>*Free time to work on projects*<br>**[Projects due WEDNESDAY DECEMBER 3, 9:00 AM]** |

### Syllabus Notes

- Lecture topics are listed in regular font; *Lab activities, other than working on the current homework and/or term projects, and which will be updated throughout the course, are listed in italics, and will link to notes for the lab activities when appropriate*. [SOTD] entries are links associated with "Simulation of the day" animations shown in class.
- *Homework assignments are denoted* **H1** *through* ***H3 and have due dates as indicated above.***
- *See Learning Goals & Course Topics page for a more detailed outline of course material.*
- *Term projects are due* **WEDNESDAY DECEMBER 3, 9:00 AM** *(note that this is the Wednesday following the last day of classes).*

### Other Important Dates

- *Tuesday, September 16:* Last day for withdrawal from this course without withdrawal standing of "W" recorded on your academic record.
- *Friday, October 11:* Last date for withdrawal from this course with withdrawal standing of "W" recorded on your academic record.
- *Monday, October 13:* Thanksgiving Day, University closed.
- *Monday, November 10:* Remembrance Day. University closed.
- *Friday, November 28:* Last day of classes.
- *Tuesday, December 2:* Examinations begin.
- *Wednesday, December 17:* Examinations end.

*See the UBC 2014/2015 Calendar and Academic Year [all year] pages for more information*

*Maintained by choptuik@physics.ubc.ca.*

## Physics 210: Intro Computational Physics
## Learning Goals & Course Topics / Outline

### LEARNING GOALS

1. **THEMATIC GOALS**

    1. To become acquainted with the use of modern computer technology to formulate and solve problems from physics (and related fields) computationally. This will generally involve:
        - Identifying or isolating a specific problem that requires solution.
        - Formulating the problem in mathematical terms, as precisely as possible.
        - Identifying appropriate approximations, algorithms, existing software etc. that will allow you to solve the problem.
        - Implementing the solution process on the computer, using programming (scripting etc.) in one or more computer languages as necessary.
        - Performing the calculations on the computer using your implementation.
        - *Analyzing and interpreting the results of the calculations.*
        - Possible iteration of one or more of the above steps in view of the results and analysis.

    2. To become familiar with basic techniques in computer programming that will be of use in solving problems from physics and related fields.

    3. To be exposed to selected topics in physics and mathematics that are representative of some typical application areas in "real world" computational physics: some of this material may already be familiar to you.

    4. To gain experience in searching for, and finding, information on specific topics/areas; in understanding that information, and then applying it (i.e. research and self-instruction!)

    5. To gain experience in writing up the results of scientific work in the form of a scientific paper.

2. **SPECIFIC GOALS**

    Successful completion of this course---which includes understanding the lecture material, completing the homeworks with a reasonable degree of proficiency, and finishing and submitting a good term project---should provide you with the ability to do the following *at a minimum*:

    1. Work comfortably within a Unix / Linux environment with an emphasis on the use of the command-line.
    2. Use Maple to interactively perform basic symbolic manipulation and numerical computations.
    3. Write simple Maple procedures (programming) as part of an introduction to the use of Maple as a powerful computing environment.
    4. Perform basic to intermediate level numerical computations using MATLAB interactively.
    5. Write basic to intermediate level MATLAB scripts and functions (programming).
    6. Use your MATLAB programming skills to address specific applications from physics and mathematics including:
        1. The use of finite difference techniques to approximately solve simple ordinary differential equations (equations of motion), of the type encountered in particle dynamics.
        2. Dynamics of one or more particles in interaction with one another or with an external potential using finite difference techniques.
        3. A moderately challenging problem of your own choosing---i.e. your term project!

**COURSE TOPICS & OUTLINE (subject to adjustment)**

### Unix: 3 lectures, 4 labs

- Unix / Linux fundamentals with a focus on use of the command line

### Maple: 5 lectures, 5 labs

- Use of a modern "symbolic manipulation" language for routine computations
- Basic Maple programming

### MATLAB: 2 lecture, 5 labs

- Introduction to MATLAB as an interactive tool for numerical calculations
- Introduction to MATLAB plotting facilities
- MATLAB programming: writing scripts and functions
- Specific MATLAB scripts/programs mostly motivated by topics covered in lectures

### Project Proposal Presentations: 2 lectures and labs

### Finite Difference Approximations With Applications to Dynamics: 6 lectures

- Definition of finite difference approximation (FDAs)
- Use of FDAs to approximate simple ordinary differential equations, such as are encountered in particle dynamics

### Final Project Free Time: 6 lectures and 6 labs

*Maintained by choptuik@physics.ubc.ca.*

# Physics 210: Intro Computational Physics: Homework Assignments

*This document will be updated throughout the course.*

***Note: Note, the due dates listed below are approximate, and subject to change!***

*To ensure that you download the most recent version of homework assignments, it is safest to first clear the disk and memory caches of your browser, or ensure that the **Preferences/Advanced/Cache** (or equivalent) setting of your browser is set so that cached documents are compared to on-line versions **every time**.*

| Homework | Due Date | Topic | Problem Set |
|---|---|---|---|
| H1 | September 25 | *Unix / Linux, Web page authoring (HTML)* | Handout [PDF]<br>Topics for Prob. 2 |
| H2 | October 16 | *Maple: Worksheets, programming* | Handout [PDF] |
| H3 | November 11 | *MATLAB programming* | Handout [PDF] |

**IMPORTANT!! HOMEWORK & TERM PAPER POLICY / ACADEMIC MISCONDUCT**

First, please refer to the section of the UBC Calendar on Policies and Regulations, especially the sections:

1. Student Declaration & Responsibility
2. Academic Honesty & Standards
3. Academic Misconduct
4. Disciplinary Measures

and ensure that you fully understand them.

*In addition, in the context of this specific course, all students must understand and abide by the following policies:*

Consultation and discussion with classmates is permitted, and in fact encouraged.

**HOWEVER, ALL HOMEWORK & TERM PROJECTS SUBMITTED MUST BE YOUR OWN WORK.**

To be more specific, the following occurrences (not an exhaustive list) *WILL* be treated as possible cases of academic misconduct. (I assume in the following that cheating is fundamentally a two-person interaction; let X and Y be two students)

1. Work where student X's work is byte-wise identical to Y's work for no good reason, and there seldom is a good reason.
2. Work where X's source code is the same or very nearly the same as Y's, with primarily comments and/or names of variables changed.

**ADDITIONAL REMARKS CONCERNING TERM PROJECTS**

Again, although you are free to consult and discuss with your classmates (and others) concerning your term projects, the work that you do for your project, as well as your writeup, must be your own work. Additionally, you must NOT use materials, particularly source code, that you locate on the Web or elsewhere in your term project: all programming and analysis that you do for your project must be original to you, although the ideas and/or algorithms underlying your programming need not be, as long as they are properly cited. Bear in mind that if you copy something from the Web, it is now quite easy for an instructor to find the same location that you did!

**The University takes all forms of academic misconduct very seriously, and so do I.**

***All strong evidence of cheating will therefore be reported to, and dealt with through, the Head of the Department of Physics & Astronomy.***

*Maintained by choptuik@physics.ubc.ca.*