1st Semester, AY 2022-2023

May T. Lim - may@nip.upd.edu.ph

National Institute of Physics
University of the Philippines Diliman

Applied Physics 155

Course Guide

00 Preface

Instructor's Notes

1. This App Physics 155 Course Guide includes a syllabus and a weekly study guide that includes the list of readings, activities, and learning tasks for the next 16 weeks.

2. Programming is an activity where "learning by doing" is effective. Even if you can copy/paste from starter codes — retyping (and even rewriting it to reflect your own style) helps.

3. Engage in group discussions. Join us on Discord.

4. In trying out new tools, some may work wonderfully well and some may be turn out to be downright awful. If you encounter issues, give us a heads up before you start debugging.

Part I

Course Syllabus

02 About the Course

Course code	Applied Physics 155
Course title	Computer Methods in Physics I
Credit	4 units
Prereq	Math 122
Description	Number systems and number representations; overview of computer hardware and software; computer programming methods; numerical analysis; research data processing
Lecturer	May T. Lim - may@nip.upd.edu.ph
Lab instructors	Adrian Chester Balingit - abalingit@nip.upd.edu.ph Sean Fortuna - sjfortuna@nip.upd.edu.ph Louie John Rubio - lrubio@nip.upd.edu.ph
Course goal	For students to develop knowledge and skills to solve problems in Physics using computer algorithms and implement code using Python to get reliable and accurate answers and avoid common pitfalls in computation.

03 Course Topics and References

Course Topics

0. Python programming for physicists
1. Integrals and derivatives
2. Solutions of linear and nonlinear equations
3. Fourier transforms
4. Ordinary differential equations
5. Partial differential equations
6. Random processes and Monte Carlo methods

References

1. [CP] M. Newman. Computational Physics. CreateSpace Independent Publishing Platform. 2012. http://www-personal.umich.edu/~mejn/cp/

2. [PfCP] S. Linge and H.P. Langtangen. Programming for Computations - Python. Springer. 2020. https://link.springer.com/content/pdf/10.1007%2F978-3-030-16877-3.pdf. License: CC-BY-4.0

3. [NumFys] NumFys: A resource for use of computational physics with Python, covering many topics in physics (Dept. of Physics, Norwegian University of Science and Technology). https://www.numfys.net (main code reference — enter the provided keywords in the Search box). License: CC-BY-NC-4.0

04 Mode and channels of delivery

AP155 will be conducted in a blended mode through a combination of UVLe (reports & quizzes), Discord (async discussions), and Zoom (sync discussions).

Lecture check-in via Zoom (with Discord backup)

There will be an organizational meeting on September 6, 10:00AM for the THU lecture class and September 7, 10:00AM for the WFU lecture class via Zoom to go over the plans for the semester.

Recurring Meeting ID: 930 6182 7654 Passcode: ap155@nip

Lab classes

Lab-specific posts are at #lab-notes-tuesday, #lab-notes-wednesday, and #lab-notes-friday.

Polls / Consultation @ Discord

Periodically pop into the #polls channel to see what's up and to let us know how thing are working out at your end.

05 Grading system

90-100	1.0
85-89	1.25
80-84	1.5
75-79	1.75
70-74	2
65-69	2.25
60-64	2.5
55-59	2.75
50-54	3
45-49	4
Below 45	5

Course requirements

80% Reports (6 corresponding to course topics 1-6)
20% Quizzes (opens right after the previous one closes at lab start time)

Reports are one-page PDF (minimum font size = 12) write-ups. List down:
a) top three (3) takeaways (i.e. key concepts) from that topic;
b) top three (3) code snippets that you think are essential to modeling/analyzing the topic; and
c) top three pitfalls (3) that have either tripped you or almost did.

There is no mandatory format (i.e. essay, presentation slide, ...) except the 1-page limit and the minimum font size.

We will be checking for concept accuracy, insights, and succinctness. Some points will be deducted for poor grammar (so spell check at the minimum). Maximum score per report is 100 points. **All submissions must be your own work.**

Practice good design -- https://developer.apple.com/design/tips/

06 Coding exercises and Coding habits

Coding exercises

As a start, imagine replicating every figure you encounter in your readings, then try replicating a few of them. Gain an understanding of the core concepts by asking and answering follow-up questions that starts with ‘What would happen if ____?’. See how modifying some parameters, variables, or even functions would answer those questions.

Coding habits that should ideally be formed by the end of the semester:

1. using version control;
2. using variables instead of hardcoded numerical parameters;
3. labeling figures and placing appropriate scales;
4. instinctively non-dimensionalizing; and
5. writing clean code*.

*Reference: (**https://drive.google.com/file/d/1TraVwRkbkCbHq-s_-NS69ZEbRNwH8XNh/view**)

Part II

Weekly Guide

08 Weeks 1-2: Python programming for physicists

Learning tasks

By the end of this period, you must be able to:

1. Run code in either Google Colaboratory or DeepNote and establish your remote computing workflow. Optional: Set up your local computing platform (Visual Studio Code, JupyterLab, …)
2. Set up *git* to perform version control for your codes and sync to GitHub.
3. Understand the need for testing code and write basic code to perform testing.
4. Write Python code to define variables, functions, loops, conditional statements, lists and arrays.
5. Visualize data in Python, including scatter, line, and density plots.
6. Demonstrate understanding of numerical errors in programming and avoid associated pitfalls.

Readings and resources

Programming and testing
Plotting and programming in Python - https://swcarpentry.github.io/python-novice-gapminder/
[CP] Chapters 2 - 4
[PfCP] Chapters 1 - 5; 6.6. Testing Code, pp. 150 - 161K. Reitz and T. Schlusser. Testing your Code. The Hitchhiker's Guide to Python. https://docs.python-guide.org/writing/tests/
Writing clean code - https://drive.google.com/file/d/1TraVwRkbkCbHq-s_-NS69ZEbRNwH8XNh/view

Tools, including version control
What is Colaboratory? https://colab.research.google.com
DeepNote - https://deepnote.com/education
GitHub Cheat Sheet. https://education.github.com/git-cheat-sheet-education.pdf

Working codes (URL are not displayed)
[NumFys] Basic plotting * Introduction to NumPy * Intermediate plotting * Intermediate NumPy

09 Weeks 3-4: Integrals and Derivatives

Learning tasks

By the end of this period, you must be able to:

1. Sketch how numerical integration and differentiation works.
2. Identify instances in Physics when you would need to integrate or differentiate.
3. Calculate integrals numerically using different methods.
4. Understand the sources of errors in numerical integration.
5. Calculate derivatives numerically.
6. Solve Physics problems using numerical integration and differentiation.

Readings and resources

[CP] Chapter 5

[NumFys] Numerical integration * Monte Carlo integration in one-dimension * Monte Carlo integration in D-dimensions

[PfCP] Chapter 6

10 Weeks 5-6: Solutions of linear and nonlinear equations

Learning tasks

By the end of this period, you must be able to:

1. Identify physics problems wherein you have to solve: (a) simultaneous linear equations; and/or (b) nonlinear equations.

2. Solve simultaneous linear equations numerically by familiarizing yourself with the needed linear algebra tasks (decomposition, inversion, matrix operations, matrix eigenvalues and eigenvectors).

3. Discuss the concepts behind solving nonlinear equations (relaxation method/fixed-point iteration, bisection method, Newton-Raphson method, …).

Readings and resources

[CP] Chapter 6

[NumFys] Linear algebra in Python * Bisection method * Newton-Raphson method * Fixed-point iteration

[PfCP] Chapter 7

11 Week 7-8: Fourier transforms

Learning tasks

By the end of this period, you must be able to:

1. Review the key concepts of the Fourier series.
2. Identify Physics problems that can be solved using Fourier transforms.
3. Explain how the Discrete Fourier transform works — and identify situations when it would “fail”.
4. Explain what happens when you change the position of the sample points while maintaining the sampling interval.
5. Work out extending the one-dimensional Fourier transform to its two-dimensional form.
6. Discuss the physical interpretation of the Fourier transform.
7. Demonstrate using the Fourier transform to solve Physics problems.

Readings and resources

[CP] Chapter 7

[NumFys] Discrete Fourier transform * Simple sound filtering using discrete Fourier transform * Image filtering using discrete Fourier transform * Doppler effect

12 Weeks 9-10: Ordinary differential equations

Learning tasks

By the end of this period, you must be able to:

1. Demonstrate knowledge of the different methods for solving first-order and second-order differential equations.
2. Demonstrate knowledge of the different methods for solving boundary value problems.
3. Implement these computational methods to solve Physics problems.

Readings and resources

[CP] Chapter 8

[NumFys] Simple pendulum * Euler's method * Verlet integration * Implicit Euler method * Double pendulum and chaos * Roller coaster * Runge-Kutta methods

[PfCP] Chapter 8

13 Weeks 11-12: Partial differential equations

Learning tasks

By the end of this week, you must be able to:

1. Identify the classic partial differential equations that you've encountered in your past Physics classes.
2. Distinguish between initial value problems and boundary value problems.
3. Describe the challenges in solving PDE's, including numerical stability.
4. Walk through the different methods for solving partial differential equations (finite difference method, Crank-Nicholson method, spectral method, finite element method, …)
5. Check out the various the PDE solvers available in the market (in the context of continuum mechanics)

Readings and resources

[CP] Chapter 9

[NumFys] Partial Differential Equations - Two Examples * Relaxation methods for solving PDE's * Iterative Gauss-Seidel method

[PfCP] Chapter 9

14 Weeks 13-14: Random processes & Monte Carlo methods

Learning tasks

By the end of this period, you must be able to:

1. Understand how random numbers are generated by computers.
2. Perform integration using Monte Carlo methods.
3. Implement Monte Carlo simulations for Physics applications.

Readings and resources

[CP] Chapter 10

[NumFys] Diffusion-limited aggregation * Self-avoiding random walk in 2D * Introduction to Brownian motion and diffusion

[YouTube, truly optional] https://www.youtube.com/watch?v=u3xIs0aajN4 * https://www.youtube.com/watch?v=gOwLEVQGbrM

[PfCP] Chapter 6 (6.7.3)

15 Weeks 15-16: Winding down

Learning tasks

By the end of this week, you should be able to:

1. Easily recall the key points of the past six topics.
2. Highlight key programming pitfalls in computational physics.

16 Beyond AP155

Readings and resources

[Reproducible science] Conda + Docker * Open Science Framework * ORCID

[Creative Commons] https://creativecommons.org/choose/

[Licensing a repository] https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/licensing-a-repository