Nanyang Technological University
Division of Physics and Applied Physics

Academic Year	AY1819	**Semester**	2
Course Coordinator	Assoc Prof Pinaki Sengupta		
Course Code	PH4419		
Course Title	Computational Physics		
Pre-requisites	PH3101 Quantum Mechanics II		
Mutually Exclusive	PH4505, PAP455		
No of AUs	4		
Contact Hours	52 hours (Lecture hours: 26 hours; Tutorial hours: 26 hours)		
Proposal Date	4 April 2019		

Course Aims

This course aims to provide you with the quantitative skills needed to study complex physical situations, such as multi-dimensional systems, nonlinear phenomena, and stochastic phenomena. Emphasis is placed on practical analysis, problem-solving, and debugging skills. These skills are developed through programming assignments, in which you will learn how to tackle a variety of physics problems in electromagnetism, quantum mechanics, and statistical mechanics.

Intended Learning Outcomes (ILO)

By the end of this course, you (as a student) would be able to:

1. Describe and explain the numerical algorithms commonly used in physics calculations.
2. Analyze the efficiency of various numerical algorithms (e.g. Big-O runtime scaling analysis and code profiling).
3. Write and debug programs for solving differential equations in physics problems.
4. Write and debug programs for performing Monte Carlo simulations of physical systems.
5. Use computational methods for comprehensive investigation of an individual research project.

Course Content

Numerical solutions of differential equations in classical mechanics, quantum mechanics and electromagnetism. Monte Carlo method for statistical mechanics simulation. Optimization and data analysis.
Various advanced topics including Quantum Monte Carlo and Density Functional Theory.

Course Outline

S/No.	Topic	Lecture Hours	Tutorial Hours
1	Introduction to Computation. Data structures.	4 hrs	10 hrs
2	Numerical Linear Algebra.	2 hrs	
3	The Finite Difference Method. Sparse matrices.	4 hrs	
4	Numerical integration. Initial Value Problems.	4 hrs	16 hrs
5	Discrete Fourier transforms	4 hrs	
6	Markov chains and Monte Carlo methods.	6 hrs	
7	Advanced topics	2 hrs	

Assessment (includes both continuous and summative assessment)

Component	Course LO Tested	Related Programme LO or Graduate Attributes	Weighting	Team/Individual	Assessment Rubrics
1. Individual project	5	Competency 1, 2, 4, 5, 7 Creativity 1, 2 Communication 1, 2 Character 1, 2	60%	Individual	Appendix 1
2. Continuous Assessment 1 (CA1): Assignment	1,2, 3,4	Competency 1, 2, 4, 5, 7 Communication 1, 2	40%	Individual	Appendix 2
Total			100%		

Graduate Attributes

Upon the successful completion of the PHY, APHY and PHMA programs, graduates should be able to:

Competency	1	demonstrate a rigorous understanding of the core theories and principles of physics involving (but not limited to) areas such as classical mechanics, electromagnetism, thermal physics and quantum mechanics;
		[PHMA only] demonstrate a rigorous understanding of the core theories and principles of mathematical sciences involving (but not limited to) areas such as analysis, algebra and statistical analysis;
	2	read and understand undergraduate level physics content independently;
	3	make educated guesses / estimations of physical quantities in general;
	4	apply fundamental physics knowledge, logical reasoning, mathematical and computational skills to analyse, model and solve problems;
	5	develop theoretical descriptions of physical phenomena with an understanding of the underlying assumptions and limitations;
	6	critically evaluate and distinguish sources of scientific/non-scientific information and to recommend appropriate decisions and choices when needed;

	7	demonstrate the ability to design and conduct experiments in a Physics laboratory, to make measurements, analyse and interpret data to draw valid conclusions.
Creativity	1	propose valid approaches to tackle open-ended problems in unexplored domains;
	2	offer valid alternative perspectives/approaches to a given situation or problem.
Communication	1	describe physical phenomena with scientifically sound principles;
	2	communicate (in writing and speaking) scientific and non-scientific ideas effectively to professional scientists and to the general public;
	3	communicate effectively with team members when working in a group.
Character	1	uphold absolute integrity when conducting scientific experiments, reporting and using the scientific results;
	2	readily pick up new skills, particularly technology related ones, to tackle new problems;
	3	contribute as a valued team member when working in a group.
Civic Mindedness	1	put together the skills and knowledge into their work in an effective, responsible and ethical manner for the benefits of society.

Formative feedback

During the computer lab sessions, the instructor is present to discuss programming/debugging problems with the students as they arise.

When each programming assignment is graded, the students are provided with individualized feedback explaining the mistakes made and suggesting improvements.

Learning and Teaching approach

Approach	**How does this approach support students in achieving the learning outcomes?**
Programming assignments	The students learn about programs by actually programming.

Reading and References

a. Computational Physics, 2nd ed, Jos Thijssen, Cambridge University Press, 978-0521833462, 2007
b. A First Course in Computational Physics, 2nd ed, Paul L. DeVries and Javier E. Hasbun, Jones and Bartlett Publishers, 978-0763773144, 2010
c. Computational Physics: Problem Solving with Computers, 2nd ed, Rubin H. Landau, Manuel J. Páez and Cristian C. Bordeianu, Wiley-VCH, 978-3527406265, 2007

Course Policies and Student Responsibilities

(1) General

Each student is expected to be present at the weekly computer lab session, where the programming assignments are worked on in the presence of the course instructor(s).

If a programming assignment cannot be completed by the end of the lab sessions, the student is expected to work on it at home.

(2) Compulsory Programming Assignments

Each student is required to complete five compulsory programming assignments over the course of the semester. These contribute equally to the CA (50% total). These assignments will be graded not only according to the correctness of the program outputs, but also on programming style and adherence to good programming practices.

Academic Integrity

Good academic work depends on honesty and ethical behaviour. The quality of your work as a student relies on adhering to the principles of academic integrity and to the NTU Honour Code, a set of values shared by the whole university community. Truth, Trust and Justice are at the core of NTU's shared values.

As a student, it is important that you recognize your responsibilities in understanding and applying the principles of academic integrity in all the work you do at NTU. Not knowing what is involved in maintaining academic integrity does not excuse academic dishonesty. You need to actively equip yourself with strategies to avoid all forms of academic dishonesty, including plagiarism, academic fraud, collusion and cheating. If you are uncertain of the definitions of any of these terms, you should go to the academic integrity website for more information. Consult your instructor(s) if you need any clarification about the requirements of academic integrity in the course.

Course Instructors

Instructor	Office Location	Phone	Email
Pinaki Sengupta	SPMS-PAP-05-03	65921801	PSENGUPTA@ntu.edu.sg

Planned Weekly Schedule

Week	Topic	Course LO	Readings/ Activities
1	Data structures	1	Read lecture notes, online resources on Python

2	Numerical linear algebra	1,2	Read lecture notes and textbooks; execute the sample codes provided in the lecture notes
3	Eigenvalue problems	1,2	Read lecture notes and textbooks; execute the sample codes provided in the lecture notes; solve similar eigenvalue problems by changing the sample codes provided or writing your own code.
4	Finite-difference methods	1,2	Read lecture notes and textbooks; execute the sample codes provided in the lecture notes; solve similar eigenvalue problems by changing the sample codes provided or writing your own code.
5	Sparse matrices	1,2	Read lecture notes and textbooks;
6	Numerical integration	1,2	Read lecture notes and textbooks; execute the sample codes provided in the lecture notes
7	Initial-value problems	1,2	Read lecture notes and textbooks; execute the sample codes provided in the lecture notes; set up and solve similar problems by writing your own code
8	Discrete Fourier transform	1,2	Read lecture notes and textbooks;
9	Discrete Fourier transform	1,2	Read lecture notes and textbooks; execute the sample codes provided in the lecture notes
10	Markov chains	1,3,4	Read lecture notes and textbooks; execute the sample

			codes provided in the lecture notes
11	Monte Carlo methods	1,3,4	Read lecture notes and textbooks; execute the sample codes provided in the lecture notes
12	Monte Carlo methods	1,3,4	Read lecture notes and textbooks; execute the sample codes provided in the lecture notes
13	Advanced topics	1,5	Research on an individual topic; identify a suitable numerical method to study the problem; develop the numerical algorithm; analyse results.

Appendix 1: Individual Project Rubrics

Criterion A: Complexity of the Physics problem (10%)
Criterion B: Choice of the numerical method (10%)
Criterion C: Execution of the numerical method (40%)
Criterion D: Application of the numerical results to understand the Physics problem (40%)

Standards	Criterion A	Criterion B	Criterion C	Criterion D
A+ to A-	Novel, challenging, open problem	Most efficient and elegant numerical approach	Excellent, clean, readable, and fully commented code with correct outputs, including original contribution to algorithm	Excellent analysis of the problem using numerical data
B+ to B-	Standard, challenging problem	Efficient numerical method	Code quality is acceptable but can be improved. Program outputs are correct or have only minor problems.	Adequate analysis of the problem using numerical data
C+ to C	Standard, simple problem	Inefficient, brute force method	Code quality is poor, and/or program outputs are incorrect	Inadequate analysis of the problem
D+ to D	Standard, worked out problem	Inapplicable method	Incomplete programs, substantially incorrect code, and/or output errors.	No understanding of the relevance of numerical results to the problem
F	No project problem	No method chosen	Completely missing, incorrect, and/or unreadable code.	No numerical data generated and no analysis presented

Appendix 2: Assessment Criteria for CA

Standards	**Criteria**
A+ to A-	Excellent, clean, readable, and fully commented code with correct outputs.
B+ to B-	Code quality is acceptable but can be improved. Program outputs are correct or have only minor problems.
C+ to C	Code quality is poor, and/or program outputs are incorrect.
D+ to D	Incomplete programs, substantially incorrect code, and/or output errors.
F	Completely missing, incorrect, and/or unreadable code.