Course Requisites (if applicable)

Pre-requisites	For graduate student: No pre-requisites For undergraduates: MH1803 Calculus for Physics, MH2802 Linear Algebra for Scientists, PS0001 Introduction to Computational Thinking
Co-requisites	
Pre-requisite to	
Mutually exclusive to	
Replacement course to	
Remarks (if any)	

Course Aims

Both theoretical and experimental physics require the solution of numerical problems, which can be as simple as finding the best fit to experimental data satisfying some constraints, to solution of complex partial differential equations. However, being able to instruct a computer to perform the solution of certain classes of problems leads to much broader advantages: a very large part of human activities can be automatized by delegating them to computers. For instance, we might want to, not only, simulate a device, but we might want to find the optimal balance between performance and cost. We might be tempted to let a human do the second part, yet that can be seen as an optimization problem and completely delegated to a machine freeing up a lot of human time.

The course aims at providing the practical skills needed to address problems numerically. Special focus will be given to problems in physics, but several examples will show the applicability of the techniques to a broader range of problems, like industrial planning.

The course will focus on a review of numerical methods for several prototypical problems and on learning how to effectively implement them for practical applications. However, given the wide variety of possible problems that can be faced in research and industry, the course will also focus on providing the student with the skills necessary to independently learn numerical techniques not covered by the course, or interface and match different techniques to address the more complex problems typical of a professional environment.

Course's Intended Learning Outcomes (ILOs)

Upon the successful completion of this course, you (student) would be able to:

ILO 1	Effectively implement numerical techniques conforming to modern programming techniques to achieve abstraction and maximal code reusability.
ILO 2	Use a wide array of numerical techniques to solve prototypical problems
ILO 3	Analyse a numerical technique to look for the most efficient solution for the problem at hand
ILO 4	Interface numerical technique to address more complex problems that cannot be reduced to a single prototypical problem

Course Content

Intro to Python and vectorisation Root finding and function optimisation Approximation, interpolation and fitting Numerical integration (trapezoid, Gauss, Monte Carlo) Ordinary differential equations (finite differences, Runge-Kutta methods) Partial differential equations (finite differences, hybrid methods) Tutorials (Schroedinger equation, Ising model, Smart traffic light) Advanced topics (preferably topics requested by students, otherwise at the discretion of the instructor)

Reading and References (if applicable)

Hairer, Norsett, Wanner, Solving Ordinary Differential Equations I, Springer Hairer, Wanner, Solving Ordinary Differential Equations II, Springer Press, Teukolsky, Vetterling, Flannery, Numerical Recipes, Cambridge University Press (http://numerical.recipes) Thijssen Computational Physics, 2nd ed, Cambridge University Press DeVries, Hasbun A First Course in Computational Physics, 2nd ed, Jones and Bartlett Publishers Landau, Páez, Bordeianu, Computational Physics: Problem Solving with Computers, 2nd ed, Wiley-VCH

Planned Schedule

Week or Session	Topics or Themes	ILO	Readings	Delivery Mode	Activities
1	Intro to Python and vectorisation	1		In-person	Jupyter notebooks
2	Root finding, 1D function optimisation	1,2		In-person	Jupyter notebooks
3	Tutorial: Shroedinger equation	1-4		In-person	Jupyter notebooks
4	Approximation, interpolation, fitting	1,2		In-person	Jupyter notebooks
5	Integration (trapezoid, Gauss)	1-3		In-person	Jupyter notebooks
6	Ordinary Differential Equation (Runge Kutta)	1-3		In-person	Jupyter notebooks
7	Integration (Monte Carlo)	1,2		In-person	Jupyter notebooks
8	Tutorial: Ising model	1-4		In-person	Jupyter notebooks
9	Ordinary Differential Equation (Finite differences)	1,2		In-person	Jupyter notebooks
10	Partial Differential Equations (Finite Differences)	1,2		In-person	Jupyter notebooks

Week or Session	Topics or Themes	ILO	Readings	Delivery Mode	Activities
11	Partial Differential Equations (hybrid)	1,2		In-person	Jupyter notebooks
12	Multidimensional Functional Optimisation	1,2		In-person	Jupyter notebooks
13	Tutorial: Smart traffic lights	1-4		In-person	Jupyter notebooks
14	Final Project Presentations	1-4	`	In-person	Jupyter notebooks

Learning and Teaching Approach

Approach	How does this approach support you in achieving the learning outcomes?
Interactive Jupyter notebooks	The course will be based on Jupyter notebooks, which blend the study of numerical algorithms with programming. Numerical techniques will be introduced by proposing ideas, testing them and then improving on them, until a more professional approach is developed.
Integrated programming	The Jupyter notebooks integrate more standard explanations with runnable code snippets. The latter will be used to build step by step the proposed numerical algorithms and are fully runnable within the Jupyter textbook. Furthermore, the Jupyter textbook comes with many exercises where the student will be asked to attempt to implement steps of the algorithm under study.

Assessment Structure

Assessment Components (includes both continuous and summative assessment)

No.	Component	ILO	Related PLO or Accreditation	Weightage	Team/Individual	Rubrics	Level of Understanding
1	Continuous Assessment (CA): Others([group or individual projects/evaluations] CA - Individual project)	1, 2, 3, 4		60	Individual	Holistic	Multistructural
2	Continuous Assessment (CA): Others([assignments (e.g. term paper, essay)] CA - Homework)	1, 3, 4		40	Individual	Analytic	Multistructural

Description of Assessment Components (if applicable)

Description of Assessment Components:

Assessment component 1: Individual Project – 60%

You will apply the knowledge and skills acquired during the course to produce the numerical solver for a complex problem. The topic and the tasks of the projects will be negotiated on an individual basis with the instructor. You are welcome to propose a project on a topic of your interest. The final project will often require the application of more than one of the studied numerical algorithms and/or adapting them. You will have to accompany the submitted code with a summary of the task, the chosen numerical strategy and why, the structure of the code, and a typical run of the code.

Assessment component 2: Homework - 40%

You will be assigned four homework throughout the course. The homework will be computational assignments and can be solved in teams. The assignments must be periodically submitted and will provide feedback on the progress of your study.

Formative Feedback

The classes will be highly interactive. The instructor will solicit you to provide creative ideas to the problems that need to be addressed before teaching the topic of the class. You will receive feedback as grades and corrections to the continuous assessment homework. You will also be highly encouraged to interact with the instructor.

NTU Graduate Attributes/Competency Mapping

This course intends to develop the following graduate attributes and competencies (maximum 5 most relevant)

Attributes/Competency	Level

Course Policy

Policy (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. On the use of technological tools (such as Generative AI tools), different courses / assignments have different intended learning outcomes. Students should refer to the specific assignment instructions on their use and requirements and/or consult your instructors on how you can use these tools to help your learning. Consult your instructor(s) if you need any clarification about the requirements of academic integrity in the course.

Policy (General)

You are expected to complete all assigned pre-class readings and activities, attend all seminar classes punctually and take all scheduled assignments and tests by due dates. You are expected to take responsibility to follow up with course notes, assignments and course related announcements for seminar sessions you have missed. You are expected to participate in all seminar discussions and activities.

Policy (Absenteeism)

Absence from class without a valid reason will affect your overall course grade. Valid reasons include falling sick supported by a medical certificate and participation in NTU's approved activities supported by an excuse letter from the relevant bodies. If you miss a lecture, you must inform the course instructor via email prior to the start of the class.

Policy (Others, if applicable)

(3) Homework and Final Project

You will be given plenty of time for both the homework and the final project. A MC for the day or days around the submission deadline will not be considered valid excuses for delayed submission. More serious health problems will be considered on a case-by-case basis.

Last Updated Date: 28-11-2024 02:32:36