# **GE CURRENT COURSE CONTENT**

Academic Year	2023/2024	Semester	1
School/Programme	SPMS		
Course Coordinator	Asst Prof Marco Battiato		
Course Code	PH7013		
Course Title	Advanced Numerical Met	hods for Phys	sicists
Pre-requisites	For graduate student: No pre-requisites For undergraduates: <u>MH1803 Calculus for Physics, MH2802</u> <u>Linear Algebra for Scientists</u> , PS0001 Introduction to Computational Thinking		
No of AUs	4		
Contact Hours	52 hours (Lecture hours: 26 hours; Tutorial hours: 26 hours)		
Proposal Date i.e. date proposal was drafted	July 2023		
Expected Implementation date of new/revised course	January 2024		
Suggested Class Size	30		
Any cross-listing? Is course opened to all Postgraduate students (including IGP) or specific program (please indicate)?	Yes, SPMS Master by Research		

#### **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.

Intend	ed Learning Outcomes (ILO)
By the	end of this course, you should be able to:
1.	Effectively implement numerical techniques conforming to modern programming techniques to achieve abstraction and maximal code reusability.
2.	Use a wide array of numerical techniques to solve prototypical problems
3.	Analyse a numerical technique to look for the most efficient solution for the problem at hand
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)

Assessment (includes both continuous and summative assessment)

Note: It is advised that Group component and class participation should not be more than 40% and 20% respectively, unless with good justification.

Component	ILO	Weighting	Team/Individual	Assessment
	Tested			Rubrics
1. Individual project	1, 2, 3, 4	60%	Individual	Appendix 1
2. Homework	1, 3, 4	40%	Individual	Appendix 1
Total	·	100%		

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.

## Learning and Teaching Approach

*Note: Please include and indicate TEL component.* 

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.

#### **Reading and References**

- 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 (<u>http://numerical.recipes</u>)
- 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

(1) 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.
(2) 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.
(2) Homework and Final Droject
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.
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 <u>academic integrity website</u> for more information. Consult your instructor(s) if you

## **Course Instructors**

**Course Policies and Student Responsibilities** 

Instructor	Office Location	Phone	Email
Marco Battiato	SPMS-PAP-05-07	65138039	marco.battiato@ntu.edu.sg

need any clarification about the requirements of academic integrity in the course.

# Industry Participation

Company Name	Description of involvement (e.g., co-curation of course, speaker or instructor), include no. of course hours if known.	Contact Person	Email

# Planned Weekly Schedule

Week	Торіс	ILO	Readings/ Activities
Week 1	Intro to Python and vectorisation	1	Jupyter notebooks
Week 2	Root finding, 1D function optimisation	1,2	Jupyter notebooks
Week 3	Tutorial: Shroedinger equation	1,2,3,4	Jupyter notebooks
Week 4	Approximation, interpolation, fitting	1,2	Jupyter notebooks
Week 5	Integration (trapezoid, Gauss)	1,2,3	Jupyter notebooks
Week 6	Ordinary Differential Equation (Runge Kutta)	1,2,3	Jupyter notebooks
Week 7	Integration (Monte Carlo)	1,2	Jupyter notebooks
Week 8	Tutorial: Ising model	1,2,3,4	Jupyter notebooks
Week 9	Ordinary Differential Equation (Finite	1,2	Jupyter notebooks
Week 10	Partial Differential Equations (Finite Differences)	1,2	Jupyter notebooks
Week 11	Partial Differential Equations (hybrid)	1,2,3	Jupyter notebooks
Week 12	Multidimensional Functional Optimisation	1,2	Jupyter notebooks
Week 13	Tutorial: Smart traffic lights	1,2,3,4	Jupyter notebooks
Week 14	Final Project Presentations	1,2,3,4	Jupyter notebooks
Other inform	nation(s)		
Jil			

## Appendix 1: Assessment Criteria for \_\_\_\_\_ Final Project & Homework \_\_\_\_\_

Criteria in no specific order

- 1) Choice of appropriate numerical methods based on performance and applicability to the task
- 2) Correct application of the numerical methods
- 3) Correct handling of edge cases
- 4) Completeness in addressing the task
- 5) Quality and efficiency of the developed numerical approach
- 6) Code clarity
- 7) Code efficiency
- 8) Code modularity and reusability
- 9) Ease of input and completeness of output