

Subject card

Subject name and code	Computational Mathematics of Evolutionary Equations, PG_00171117						
Field of study	Mathematical Modeling and Data Analysis						
Date of commencement of studies	October 2023		Academic year of realisation of subject		2025/2026		
Education level	Bachelor's studies		Subject group				
Mode of study	full-time studies		Mode of delivery		at the university		
Year of study	3		Language of instruction		English		
Semester of study	6		ECTS credits		6.0		
Learning profile	academic		Assessment form		exam		
Conducting unit	Institute of Mathematics -> Faculty of Mathematics, Physics and Informatics -> Rector						
Name and surname of lecturer (lecturers)	Subject supervisor		dr hab. Karolina Kropielnicka				
	Teachers						
Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	30.0	0.0	30.0	0.0	0.0	60
	E-learning hours included: 0.0						
Learning activity and number of study hours	Learning activity	Participation in didactic classes included in study plan		Participation in consultation hours		Self-study	SUM
	Number of study hours	60		10.0		80.0	150
Subject objectives	The aim of the course is to introduce students to numerical methods for solving evolutionary equations, i.e., equations that describe phenomena evolving in time and space, commonly arising in physics, biology, chemistry, and engineering. The course covers spatial and temporal discretisation techniques (e.g., finite difference, finite element, and spectral methods) and investigates the numerical algorithms used to solve such models. Special emphasis is placed on understanding key algorithmic properties such as accuracy, stability, efficiency, and robustness.						
Learning outcomes	Course outcome		Subject outcome		Method of verification		
	[MMiADL3_K02] is ready to precisely formulate questions to deepen his/her own understanding of a given topic or to find missing elements of reasoning		The student is ready to formulate precise questions related to numerical and computational topics in order to deepen their own understanding and to identify missing steps in mathematical or algorithmic reasoning.		[SK1] oral statement/conversation/discussion		
	[MMiADL3_U10] is able to recognise problems, including practical issues, that can be solved algorithmically; can make a specification of such a problem		The student is able to identify mathematical problems, particularly evolutionary equations, that can be solved algorithmically, and can formulate precise specifications for their computational implementation.		[SU4] test/exam - oral or written		
	[MMiADL3_U11] knows how to arrange and analyse an algorithm in accordance with the specification and save it in the selected programming language		The student is able to design and analyse a numerical algorithm for solving evolutionary equations and implement it in Python according to a given specification.		[SU4] test/exam - oral or written		
	[MMiADL3_W09] knows and understands the basics of computational and programming techniques supporting mathematician's work and understands their limitations		The student understands the fundamentals of computational techniques used to solve evolutionary equations and is able to apply programming tools with awareness of their capabilities and limitations.		[SW4] test/exam - oral or written		

Subject contents

1. Introduction to Evolutionary Equations:
 - Examples from wave, diffusion, and Schrödinger-type problems.
2. Foundations of Computational Mathematics for Evolution Equations:
 - Space-time discretisation: grid concepts, time/space steps.
 - Algorithmic quality criteria: accuracy, stability, convergence, efficiency.
3. Time Discretisation:
 - Explicit and implicit methods (Euler, Crank-Nicolson, Runge-Kutta).
 - Stability and stiffness (A-stability, L-stability).
4. Space Discretisation:
 - Finite difference methods (FDM).
 - Finite element methods (FEM).
 - Introduction to spectral methods and applications.
5. Solving Evolutionary Equations:
 - Coupling space and time discretisation.
 - Schemes for parabolic, hyperbolic, and Schrödinger-type equations.
6. Numerical Analysis of Algorithms:
 - Error sources: truncation, rounding.
 - CFL condition and its implications.
 - Conservation of physical quantities (e.g. energy) in numerical schemes.
7. Implementation and Visualisation:
 - Python implementation (NumPy, Matplotlib, SciPy).
 - Visualising solutions time evolution, spatial profiles.
8. Selected Topics and Student Projects:
 - Applications to realistic problems (e.g. wave propagation, heat transfer).
 - Projects using real-world physical or biological models.

Prerequisites and co-requisites	Prerequisites:		
	• Basic knowledge of calculus (derivatives, integrals, elementary differential equations). • Familiarity with linear algebra (systems of linear equations, matrices, vector spaces). • Basic programming skills in Python (variables, loops, functions, use of numerical libraries like NumPy). • Ability to communicate in English at a level sufficient for following lectures and participating in exercises.		
	Additional recommended preparation:		
	• Completion of the <i>Foundations of Computational Mathematics</i> course (not required but helpful). • Familiarity with basic numerical methods (e.g. Eulers method, interpolation, numerical integration). • Interest in applications of mathematics to natural sciences and engineering.		
Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	egzam	51.0%	100.0%
Recommended reading	Basic literature	1. Peter Deuflhard, Folkmar Bornemann, Scientific computing with ordinary differential equations, Springer, 2002. 2. Peter Deuflhard, Martin Weiser, Adaptive numerical solution of pdes, DeGruyter, 2012. 3. Ernst Hairer, Gerhard Wanner, Solving ordinary differential equations II, 2nd edition, Springer, 1996. 4. A. Iserles, A First Course in the Numerical Analysis of Differential Equations, CUP, 2009.	
	Supplementary literature	1. Ben Leimkuhler, Sebastian Reich, Simulating Hamiltonian dynamics, CUP, 2004. 2. C. Lubich, From Quantum to Classical Molecular Dynamics: Reduced Models and Numerical Analysis, EMS, 2008.	
	eResources addresses		
Example issues/ example questions/ tasks being completed	• Differences between explicit and implicit time discretisation methods. • Numerical stability the concept of A-stability and its importance. • Space discretisation: comparison of finite difference, finite element, and spectral methods.		
Work placement	Not applicable		

Document generated electronically. Does not require a seal or signature.