Subject card

<table>
<tr><td>Subject name and code</td><td colspan="6">Computational Methods of Physics, PG_00182298</td></tr>
<tr><td>Field of study</td><td colspan="6">Physics</td></tr>
<tr><td>Date of commencement of studies</td><td colspan="2">October 2026</td><td colspan="2">Academic year of realisation of subject</td><td colspan="2">2027/2028</td></tr>
<tr><td>Education level</td><td colspan="2">Bachelor's studies</td><td colspan="2">Subject group</td><td colspan="2">Obligatory subject group in the field of study
Subject group related to scientific research in the field of study</td></tr>
<tr><td>Mode of study</td><td colspan="2">full-time studies</td><td colspan="2">Mode of delivery</td><td colspan="2">at the university</td></tr>
<tr><td>Year of study</td><td colspan="2">2</td><td colspan="2">Language of instruction</td><td colspan="2">Polish</td></tr>
<tr><td>Semester of study</td><td colspan="2">3</td><td colspan="2">ECTS credits</td><td colspan="2">3.0</td></tr>
<tr><td>Learning profile</td><td colspan="2">academic</td><td colspan="2">Assessment form</td><td colspan="2">credit</td></tr>
<tr><td>Conducting unit</td><td colspan="6">Laboratory for Physics Teaching -> Institute of Experimental Physics -> Faculty of Mathematics, Physics and Informatics -> Rector</td></tr>
<tr><td rowspan="2">Name and surname of lecturer (lecturers)</td><td colspan="2">Subject supervisor</td><td colspan="4">dr Adrian Kołodziejski</td></tr>
<tr><td colspan="2">Teachers</td><td colspan="4"></td></tr>
</table>

<table>
<tr><td rowspan="3">Lesson types</td><td>Lesson type</td><td>Lecture</td><td>Tutorial</td><td>Laboratory</td><td>Project</td><td>Seminar</td><td>SUM</td></tr>
<tr><td>Number of study hours</td><td>0.0</td><td>0.0</td><td>45.0</td><td>0.0</td><td>0.0</td><td>45</td></tr>
<tr><td colspan="7">E-learning hours included: 0.0</td></tr>
<tr><td rowspan="2">Learning activity and number of study hours</td><td>Learning activity</td><td colspan="2">Participation in didactic classes included in study plan</td><td>Participation in consultation hours</td><td colspan="2">Self-study</td><td>SUM</td></tr>
<tr><td>Number of study hours</td><td colspan="2">45</td><td>0.0</td><td colspan="2">30.0</td><td>75</td></tr>
<tr><td>Subject objectives</td><td colspan="7">• Presentation and teaching of numerical methods for solving problems encountered by physicists during experimental measurements or theoretical calculations.
• Further development of programming skills and familiarization of students with the numerical, graphical, and symbolic capabilities of Python and Mathematica.
• Consolidation of skills in creating research-oriented scientific documents.</td></tr>
</table>

Learning outcomes	Course outcome	Subject outcome	Method of verification
	[FIZL3_W12] knows the methods of numerical analysis, knows at least one package for symbolic calculations at an advanced level, knows application software packages for presentation of results and data analysis; knows one programming language fluently	Student is familiar with: – the capabilities of the Mathematica software package.	[SW2] presentation/project/paper/report
	[FIZL3_U11] can use various application software packages to present results and analyze data	Student is familiar with: – the use of numerical (e.g. numpy, scipy) and graphical (matplotlib.pyplot) libraries in Python,	[SU2] presentation/project/paper/report
	[FIZL3_K02] can precisely formulate problems aimed at deepening the understanding of a given topic	The student is able to: - transform the description of a physical phenomenon into a computational problem with clearly defined objectives; - select and justify a numerical method appropriate for solving the given problem; - determine the initial and boundary conditions that allow analyzing the influence of parameters on the solution.	[SK2] presentation/project/paper/report
	[FIZL3_U08] can use numerical methods or software for symbolic calculations or simulations to describe and model physical phenomena and processes	The student is able to: – numerically determine the zeros of nonlinear functions, – perform interpolation or approximation of a function given at a limited number of nodes, – compute the numerical value of a definite integral over a finite interval, – obtain the solution of a system of linear equations, – solve initial value problems for ordinary differential equations, – generate computer code to solve selected numerical problems, – create a research-style report describing the obtained results.	[SU2] presentation/project/paper/report
	[FIZL3_U12] can write, compile, run, test and document a computer program written by himself	The student is able to: - write a computer program implementing a specified physical algorithm; - compile and run the program in a chosen programming environment; - test the program, verifying the correctness of results and stability of operation; - prepare documentation describing the assumptions, implementation, and results of the program.	[SU2] presentation/project/paper/report
	[FIZL3_W11] knows computational methods used in classical mechanics, electrodynamics, quantum mechanics and statistical physics	The student is familiar with: – computer arithmetic and methods for estimating errors arising in numerical computations, – methods for solving nonlinear equations or systems of equations, including bisection, Newton-Raphson and secant methods, – interpolation methods (e.g. Lagrange, piecewise polynomials) and function approximation techniques (e.g. least squares), – numerical integration methods, including Newton-Cotes formulas, Gaussian quadrature, and Monte Carlo techniques, – direct (e.g. Gaussian elimination) and iterative (e.g. Jacobi, Gauss-Seidel) methods for solving systems of linear equations, – explicit and implicit methods (e.g. Euler, Runge-Kutta) for solving initial value problems for ordinary differential equations,	[SW2] presentation/project/paper/report

Subject contents	1. Computer arithmetic and performing numerical computations. 2. Errors in numerical calculations. Ill-conditioned problems and algorithms. 3. Python libraries for numerical and graphical procedures: numpy, scipy, matplotlib.pyplot. 4. Solving nonlinear equations: fixed-point iteration, Newton-Raphson and secant methods. 5. Interpolation: Lagrange and piecewise polynomial methods. 6. Approximation: least squares method. 7. Direct and iterative methods for solving systems of linear equations Gaussian elimination, Jacobi method, Gauss-Seidel method. 8. Numerical integration: Newton-Cotes quadrature (simple and composite), Gaussian quadrature, Monte Carlo method. 9. Numerical differentiation. 10. Ordinary differential equations initial value problems: explicit and implicit Euler methods, Runge-Kutta methods. 11. (Optionally) Basic programming in Mathematica.
Prerequisites and co-requisites	Formal requirements: Completion of Foundations of Mathematics and Programming courses, and a passed exam in Logic and Algebra. Prerequisites: Familiarity with programming principles in Python, and knowledge of linear algebra and mathematical analysis, including definitions and analytical techniques for computing integrals, solving nonlinear equations, systems of linear equations, and differential equations.
Assessment methods and criteria	see table below

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
project	51.0%	100.0%

Recommended reading

Basic literature	• Jaan Kiusalaas, *Numerical Methods in Engineering with Python 3* , Cambridge University Press 2013 • P. L. de Vries, A first course in computational physics, John Wiley & Sons, Inc. New York 1994 • Å. Björck, G. Dahlquist, Metody numeryczne PWN 1987 • J. M. Jankowscy, Przegląd algorytmów numerycznych, Wyd. Naukowo-Techniczne 1988 • J. Stoer, R. Burlisch, Wstęp do analizy numerycznej, PWN 1987
Supplementary literature	not applicable
eResources addresses	

Example issues/ example questions/ tasks being completed	not applicable
Work placement	Not applicable

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