Faculty of Science

FYSN33, Physics: Applied Computational Physics and Machine Learning, 7.5 credits

Fysik: Tillämpad beräkningsfysik och maskininlärning, 7,5 högskolepoäng

Second Cycle / Avancerad nivå

Details of approval

The syllabus was approved by The Education Board of Faculty of Science on 2024-12-12 and was last revised on 2026-04-30. The revised syllabus comes into effect 2026-04-30 and is valid from the spring semester 2027.

General information

The course is an elective course for second-cycle studies for a Degree of Master of Science (120 credits) with a specialisation in physics.

Language of instruction: Swedish and English
If needed, the course is given in English in its entirety.

Main field of study	*Specialisation*
Physics	A1N, Second cycle, has only first-cycle course/s as entry requirements

Learning outcomes

The overall purpose of the course is to give knowledge and skills to approach a physics problem and to select, motivate and implement a suitable numerical solution technique.

Knowledge and understanding

On completion of the course, the students shall be able to:

- explain the difference between central numerical methods and their applicability to physics problems,
- assess the strengths and weaknesses of a selection of machine learning models in physics applications,

- explain the importance of performance considerations and how the choice of implementation affects computational efficiency.

Competence and skills

On completion of the course, the students shall be able to:

- develop numerical solvers for ordinary and partial differential equations using Python and C++, and effectively visualize their results,
- implement Monte Carlo simulations and machine learning classifiers to solve probabilistic and classification problems in physics,
- use modern version control systems for managing code, collaborating, and documenting their project progress.

Judgement and approach

On completion of the course, the students shall be able to:

- critically evaluate different numerical methods and machine learning algorithms, choosing the most appropriate approach for a given problem in physics,
- analyze the trade-offs in computational performance and resource use (e.g., CPU vs. GPU, runtime environments) for different types of physics simulations,
- discuss the impact of code design and algorithm efficiency in scientific computing, including sustainability aspects such as energy consumption in large-scale simulations,
- demonstrate a systematic approach to physics problems by selecting, motivating, and evaluate their choice of numerical methods, computational techniques, and tools,
- demonstrate adaptability to new computational techniques and libraries by leveraging AI assistance where applicable, while being mindful of its limitations.

Course content

The course covers a variety of numerical and analysis techniques, technical skills as well as aspects of algorithm design and soft skills necessary for completing applied computational physics projects. The course covers:

- Numerical techniques for solving ordinary and partial differential equations, e.g.: Runge-Kutta, matrix method, forward and backward Euler algorithms, as well as the Finite Element Method.
- Monte Carlo techniques for solving probabilistic problems, e.g.: Pseudo-random number generation, distribution sampling, Metropolis algorithm and genetic algorithms.
- Machine learning techniques for classification tasks and generative models, e.g.: Boosted Decision Trees (BDTs), k-means/clustering, Generative Adversarial Networks (GANs) /normalizing flows.
- Technical programming skills, including C++ and Python programming, the latter in various environments, packaging of libraries, basics of GPU programming.

- Aspects of algorithm design and soft skills, including application of algorithm paradigms (such as e.g. divide-and-conquer or greedy algorithms) on concrete problems, sustainability aspects on implementation choices, practical experience with AI assistance for implementation, and its limitations, use of version control in concrete projects, and using it for publishing code.

Course design

The course is mainly oriented towards problem-oriented project work, with focus on hands-on experience in solving physics problems using the knowledge, competence, and skills listed above. The course is divided into modules containing major and minor projects. All projects are mandatory and must be documented by the student in an online portfolio, based on a version control system. The portfolio should document both the theoretical background of algorithms used, theoretical or practical considerations behind choices made, and the actual code written.

Each module will be accompanied by lectures covering the theory, as well as workshop sessions with project supervision.

Assessment

The examination consists of a an oral presention of one of the major projects at a semianr, as well as an oral examination at the end of the course based on the project portfolio. All course projects must be approved before the oral examination can be taken.

Students who do not pass a regular exam are offered a re-exam shortly after the regular exam.

The examiner, in consultation with Disability Support Services, may deviate from the regular form of examination in order to provide a permanently disabled student with a form of examination equivalent to that of a student without a disability.

Grades

Grading scale includes the grades: Fail, Pass
To pass the course, all projects must be completed and the project reports approved, an approved presentation at a student seminar, and the oral exam passed.

Entry requirements

The prerequisites required for admission to the course are: 75 credits in Physics and 45 credits in Mathematics or a Bachelor of Science in Physics, in both cases including knowledge corresponding to an introductory course in numerical method, and basic knowledge of Python such as NUMA01, Numerical Analysis: Computational Programming with Python, 7.5 credits. English 6/B and basic eligibility.

Further information

The course replaces FYTN03, Theoretical Physics: Computational Physics, 7.5 credits and may not be credited towards a degree together with this one. The same applies to the course FYSM01 Physics 4, Introduction to advanced studies in physics, if this includes FYTN03, Theoretical Physics: Computational Physics, 7.5 credits as a module.

The course cannot count towards a degree in Physics together with BERN04, Computational Science: Introduction to Artificial Neural Networks and Deep Learning, 7.5 credits.

Knowledge corresponding to MNXB11: Introduction to Programming and Computing for Scientists, 7.5 credits is recommended but not required.

The course is given by the Physics department, Lund Univeristy