

Master program in Computational Science at the University of Oslo

Learning outcomes

Planned start: Fall 2018

Computing competence

Computing means solving scientific problems using computers. It covers numerical as well as symbolic computing. Computing is also about developing an understanding of the scientific process by enhancing algorithmic thinking when solving problems. Computing competence has always been a central part of the science and engineering education.

Modern computing competence is about

- derivation, verification, and implementation of algorithms
- understanding what can go wrong with algorithms
- overview of important, known algorithms
- understanding how algorithms are used to solve mathematical problems
- reproducible science and ethics
- algorithmic thinking for gaining deeper insights about scientific problems

Key elements in computing competence

The power of the scientific method lies in identifying a given problem as a special case of an abstract class of problems, identifying general solution methods for this class of problems, and applying a general method to the specific problem (applying means, in the case of computing, calculations by pen and paper, symbolic computing, or numerical computing by ready-made and/or self-written software). This generic view on problems and methods is particularly important for understanding how to apply available, generic software to solve a particular problem.

Computing competence represents a central element in scientific problem solving, from basic education and research to essentially almost all advanced problems in modern societies. Computing competence is simply central to further progress. It enlarges the body of tools available to students and scientists beyond classical tools and allows for a more generic handling of problems. Focusing on algorithmic aspects results in deeper insights about scientific problems.

Today's projects in science and industry tend to involve larger teams. Tools for reliable collaboration must therefore be mastered (e.g., version control systems, automated computer experiments for reproducibility, software and method documentation).

Overarching description of the CS program

As a student of this program you learn to use the computer as a laboratory for solving problems in science and engineering. The program offers exciting thesis projects from many disciplines: biology and life science, chemistry, mathematics, biomedical computing and bioinformatics, physics, geoscience, materials science, mechanics, computational finance, computational informatics, big data analysis, digital signal processing and image analysis, where you select the research field according to your interests. A Master's degree from this program gives you a methodical training in planning, conducting, and reporting large research projects, often together with other students and university teachers. The projects emphasize finding practical solutions, developing an intuitive understanding of the science and the scientific methods needed to solve complicated problems, use of modern tools, and not least developing own creativity and independent scientific thinking. The thesis work is a scientific project where you learn to tackle a scientific problem in a professional manner. The program aims also at developing a deep understanding of the role of computing in solving modern scientific problems. With a degree from this program you will gain deep insights in the fundamental role computations play in our advancement of science and technology, as well as the role computations play in society.

Description of learning outcomes

The power of the scientific method lies in identifying a given problem as a special case of an abstract class of problems, identifying general solution methods for this class of problems, and applying a general method to the specific problem (applying means, in the case of computing, calculations by pen and paper, symbolic computing, or numerical computing by ready-made and/or self-written software). This generic view on problems and methods is particularly important for understanding how to apply available, generic software to solve a particular problem.

Computing competence represents a central element in scientific problem solving, from basic education and research to essentially almost all advanced problems in modern societies. Computing competence is simply central to further progress. It enlarges the body of tools available to students and scientists beyond

classical tools and allows for a more generic handling of problems. Focusing on algorithmic aspects results in deeper insights about scientific problems.

The learning outcomes are subdivided in three general categories, knowledge, skills and general competence.

- **Knowledge:** As a candidate from this program you will gain a
 - deep knowledge of the scientific method and computational science at an advanced level, meaning that you
 1. have theoretical and practical knowledge of a wide range of computational methods and mathematical algorithms, including principles for developing and generalizing such methods and algorithms
 2. understand how to apply computational methods to extract information from experimental data and solve scientific problems
 3. understand the limitations of numerical methods, including approximation errors, round-off errors and the constraints on the applicability of specific algorithms
 - and you understand the possibilities and limits of computational modeling meaning that you
 1. can transform scientific problems into generic computational models and understand how various error sources influence the accuracy and reliability of the models and the computed results
 2. have an overview of advanced algorithms for solving a wide range of problems and how they can be accessed in available software
- **Skills:** As a candidate from this program you
 - have developed a practical mastery of computing, including the interplay between scientific problems and data, mathematical models, generic algorithms and reusable software, meaning that you
 1. are able to analyze and visualize computed results and evaluate their relevance with respect to the underlying problems and/or hypotheses
 2. have a working understanding of high-performance computing elements including memory usage, vectorization and parallel algorithms, and related software tools like debuggers, test frameworks, scripts, and version control systems
 3. can program in high-level and compiled languages and make efficient use of a computer algebra system
 4. understand how to increase the efficiency of numerical algorithms and pertinent software
 5. and you are familiar with techniques for collaborative software development

- **General competence:** Upon completion of this program
 - you have developed a fundamental understanding and knowledge of scientific work and the scientific method, including ethical and societal limitations and possibilities. This means among other things that you
 1. can develop hypotheses and suggest ways to test these using relevant analytical, experimental and numerical tools
 2. can reflect on and develop strategies and tools to make science reproducible and have a sound ethical approach to scientific problems
 - you can communicate in a professional way scientific problems, results and uncertainties, orally and in written form
 - you have developed a sound, scientific intuition and can reflect over and develop efficient and personal learning strategies
 - you can work independently but also in close collaboration with others to complete a research project on time

By completing a Master degree in Computational Science, you will have developed a critical understanding of the scientific methods which have been studied, have a better understanding of the scientific process per se as well as having developed perspectives for future work and how to verify and validate scientific results.