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

Students of this program 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, informatics, physics, geophysics, mechanics, geology, computational finance, computational informatics, b ig data analysis, digital signal processing and image analysis – the candidates select research field according to their interests.

A Master's degree from this program gives the candidate 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 many tools, and not least developing own creativity and independent thinking. The thesis work is a scientific project where the candidates 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. A candidate from this program gains deep insights in the fundamnetal 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**: A candidate from this program
  - has deep knowledge of the scientific method and computational science at an advanced level, meaning that the candidate
    1. has the ability to understand advanced scientific results in new fields
    2. has fundamental understanding of methods and tools
    3. can develop and apply advanced computational methods to scientific problems
    4. is capable of judging and analyzing all parts of the obtained scientific results
    5. can present results orally and in written form as scientific reports/articles
    6. can propose new hypotheses and suggest solution paths
    7. can generalize mathematical algorithms and apply them to new situations
    8. can link computational models to specific applications and/or experimental data
    9. can develop models and algorithms to describe experimental data
    10. masters methods for reproducibility and how to link this to a sound ethical scienfitic conduct
    11. has a thorough understanding of how computing is used to solve scientific problems
    12. knows fundamental algorithms in computational science
  - has a fundamental understanding and knowledge of scientific work, meaning that
    1. the candidate can develop hypotheses and suggest ways to test these
    2. can use relevant analytical, experimental and numerical tools and results to test the scientific hypotheses
    3. can generalize from numerical and experimental data to mathematical models and underlying principles
    4. can analyze the results and evaluate their relevance with respect to the actual problems and/or hypotheses
    5. can present the results according to good scientific practices
- **Skills**: A candidate from this program

- has a deep understanding of what computing means, entailing several or all of the topics listed below
  1. knows the most fundamental algorithms involved, how to optimize these and perform statistical uncertainty quantification
  2. has overview of advanced algorithms and how they can be accessed in available software and how they are used to solve scientific problems
  3. has knowledge and understands high-performance computing elements: memory usage, vectorization and parallel algorithms
  4. can use effeciently high-performance computing resources, from compilers to hardware architectures
  5. understands approximation errors and what can go wrong with algorithms
  6. has knowledge of at least one computer algebra system and how it is applied to perform classical mathematics
  7. has extensive experience with programming in a high-level language (MATLAB, Python, R)
  8. has experience with programming in a compiled language (Fortran, C, C++)
  9. has experience with implementing and applying numerical algorithms in reusable software that acknowledges the generic nature of the mathematical algorithms
  10. has experience with debugging software
  11. has experience with test frameworks and procedures
  12. has experience with different visualization techniques for different types of data
  13. can critically evaluate results and errors
  14. can develop algorithms and software for complicated scientific problems independently and in collaboration with other students
  15. masters software carpentry: can design a maintainable program in a systematic way, use version control systems, and write scripts to automate manual work
  16. understands how to increase the efficiency of numerical algorithms and pertinent software
  17. has knowledge of stringent requirements to efficiency and precision of software
  18. understands tools to make science reproducible and has a sound ethical approach to scientific problems

- **General competence**: A candidate from this program
  - is able to develop professional competence through the thesis work, entailing:

1. mature professionally and be able to work independently
2. can communicate in a professional way scientific results, orally and in written form
3. can plan and complete a research project
4. can develop a scientific intuition and understanding that makes it possible to present and discuss scientific problems, results and uncertainties

- is able to develop virtues, values and attitudes that lead to a better understanding of ethical aspects of the scientific method, as well as promoting central aspects of the scientific method to society. This means for example that the candidate

  1. can reflect on and develop strategies for making science reproducible and to promote the need for a proper ethical conduct
  2. has a deep understanding of the role basic and applied research and computing play for progress in society
  3. is able to promote, use and develop version control tools in order to make science reproducible
  4. is able to critically evaluate the consequences of own research and how this impacts society
  5. matures an understanding of the links between basic and applied research and how these shape, in a fundamental way, progress in science and technology
  6. can develop an understading of the role research and science can play together with industry and society in general
  7. can reflect over and develop learning strategies for life-long learning.

By completing a Master of Science thesis, the candidate will have developed a critical understanding of the scientific methods which have been studied, has 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.