Master program in Computational Science at the University of Oslo

Program Structure

Planned start: Fall 2018

Thesis directions

The program has the following study directions/options

- Computational Science: Applied Mathematics and Risk Analysis
- Computational Science: Astrophysics
- Computational Science: Bioinformatics
- Computational Science: Bioscience
- Computational Science: Chemistry
- Computational Science: Geoscience
- Computational Science: Imaging and Biomedical Computing
- Computational Science: Materials science
- Computational Science: Mechanics
- Computational Science: Physics

Structure and courses

The table here is an example of a suggested path for a Master of Science project, with course work the first year and thesis work the last year.

	10 ECTS	10 ECTS	10 ECTS
4th semester	Master thesis	Master Thesis	Master Thesis
3rd semester	Master thesis	Master Thesis	Master Thesis
2nd semester	Master courses	Master courses	Master courses
1st semester	Master courses	Master courses	Master courses

The program is very flexible in its structure and students may opt for starting with their thesis work from the first semester and scatter the respective course load across all four semesters. Depending on interests and specializations, there are many courses on computational science which can make up the required curriculum of course work.

Presently available courses at UiO

The University of Oslo offers the following courses in Computational Science, split here according to main disciplines/fields. All Master of Science courses will require a revision of their descriptions. This revision will include the addition of professional competences. In this process, one could think of reshaping some of the existing courses below and rename them in order to align them closely with the present program.

Mathematics and Computer Science, including Mechanics and Statistics
MAT-INF3360 Introduction to Partial Differential Equations
MAT-INF4110 Mathematical Optimization
MAT-INF4130 Numerical Linear Algebra
MAT-INF4140 Numerical Analysis
MAT-INF4160 Topics in Geometric Modelling
MAT-INF4300 Partial differential equations and Sobolev spaces I
MAT-INF4310 Partial differential equations and Sobolev spaces II
MEK4250 Finite Element Methods in Computational Mechanics
MEK4470 Computational Fluid Mechanics
INF4300 Digital image analysis
INF4331 Problem solving with high level languages
INF4820 Algorithms for artificial intelligence and natural language processing
INF5620 Numerical Methods for Partial Differential Equations
INF5631 Project on Numerical Methods for Partial Differential Equations
INF5670 Numerical methods for Navier-Stokes equations
INF5840 Computability theory
INF5850 Machine Learning for Image Analysis
STK4021 Applied Bayesian Analysis and Numerical Methods
STK4520 Laboratory for Finance and Insurance Mathematics

Physical Sciences: Physics, Astrophysics, Geosciences and Chemistry
FYS4150 Computational Physics I
FYS4411 Computational Physics II
FYS4460 Computational Physics III
GEO4310 Stochastic methods in hydrology
GEO4510 Atmosphere and Oceans on Computers: Fundamentals
GEO4450 Geophysical Fluid Dynamics
AST5210 Stellar Atmospheres I
AST9110 Numerical Modeling

Bioscience including Bioinformatics
INF4490 Biologically inspired computing
INF4350 Introductory Course in Bioinformatics
INF-BIO5121 High Throughput Sequencing technologies and bioinformatics analysis
INF5380 High-performance computing in bioinformatics
INF5560 Computational Physiology
MBV-INF4410 Bioinformatics for Molecular Biology
MBV3070 Bioinformatics

New Courses

In order to build a common study program and identity as a Computational Science student, the students will have to choose two out of three compulsory courses that aim at providing topics of common and broad interest. Both courses have a workload of 10 ECTS each. These three **new courses** (final label to be determined, only tentative name here) are

- **CS-MATH1-?**: *Data analysis and machine learning*, (10 ECTS)
    1. Monte Carlo methods and statistical data analysis
    2. Optimization of data and handling of large data sets
    3. Machine learning, including genetic algorithms and neural networks

Summary: This will be a new 10-credit course to be offered at MAT-NAT. The course is first and foremost designed as one basic ingredient of the curriculum for the "Master program in Computational Science". However, this course should also be useful for other students, as well as scientific staff at UiO, who want to learn about statistical data analysis, Monte Carlo methods and machine learning.

The content of this course focuses on three main topics

1. Monte Carlo methods, statistical data analysis, information theory and Bayesian statistics. Methods like bootstrapping and blocking to analyse data will be discussed

2. Optimization of data and algorithms for handling large data sets.

3. Machine learning with a focus on topics such as decision tree learning, maximum entropy model, support vector machine and kernel methods, graphic models, Bayesian learning methods, dimension reduction, neural networks and genetic algorithms.

4. **CS-MATH2**: *Mathematical modeling in Science*, (10 ECTS)
    (a) Central algorithms in mathematics applied to various fields in the Sciences
    (b) Applications to physical systems

5. **CS-INF1**: *High-Performance Computing and Numerical projects*, (10 ECTS
    (a) This course teaches you to develop and structure large numerical projects, from code writing to finalizing a report
    (b) Topics which are included are parallelization and vectorization
    (c) Machine architecture and GPU-CPU programming
    (d) Optimization of code and benchmarking
    (e) Numerical methods from linear algebra will be discussed as well as examples from life science.

Summary: This will be a new 10-credit course to be offered at MAT-NAT. The course is first and foremost designed as one basic ingredient of the curriculum for the "Master program in Computational Science". However, this course should also be useful for other students, as well as scientific staff at UiO, who want to learn the fundamental knowledge about efficiently applying modern computer architectures to solving computation-heavy scientific problems.

Overlap: 5 credit points of overlap with INF3380.

Replacing INF3380 altogether? Probably yes.

Pre-requirement of students: basic knowledge about programming and numerical methods.

Learning outcome:

1. capable of effectively using laptop/desktop, as well as with thread-based parallelism and OpenMP programming, for solving simple numerical problems;

2. capable of extending numerical calculations (that fit for a distributed memory model) to a computer cluster with the help of basic MPI programming;

3. be informed about cutting-edge hardware (such as GPUs) and programming tools (such as OpenMP v4, OpenACC, CUDA, MPI v3);

4. ready to learn more about advanced parallel computing.

Content: The course's content can be roughly divided into two big parts:

- How to efficiently use a laptop or desktop computer
    1. a quick introduction of modern computer architecture (processor, memory, cache, registers, instruction-level parallelism, non-uniform access shared memory, compiler)
    2. the realistically achievable performance of a computational problem (due to hardware limitations, independent of actual implementation)
    3. good programming practices ("clean code")
    4. implementation of typical numerical calculations with efficient data structures and clean coding (through examples and hands-on exercises)

5. hands-on encounters with compiling and debugging/profiling
6. vectorization of scientific code (this is one of the persistent topics for extracting the capability of modern processors)
7. other typical single-core performance optimizations (such as loop fusion, unrolling)
8. thread-based programming (OpenMP) that makes use of multiple cores while assuming a shared memory model

- High-performance scientific computing beyond a laptop/desktop
  1. typical cluster architecture (interconnect as the new performance bottleneck, distributed memory as the new programming challenge)
  2. explicit work and data partitioning that matches with the distributed memory scenario
  3. collaboration in form of communication and synchronization, also the resulting scalability "curse"
  4. basic issues of parallel program design: load balancing, reducing overhead of communication, speedup (and its limitation)
  5. MPI programming (despite its many details and a steep learning curve, this standard will prevail in the foreseeable future, possibly also in the exascale era)
  6. GPU as a computing accelerator and the streaming programming style (CUDA or OpenCL), because it is highly probable that many future supercomputers will use GPUs
  7. Hybrid parallel computing (shared memory + distributed memory)
  8. A peek into adanved programming tools (such as OpenMP v4, OpenACC, CUDA, MPI v3)

Mandatory assignment/project:

1. one mandatory assignment (a prescribed numerical problem, intended for using a shared-memory computer system)

2. one project (with the possibility of configuring an individual project for each student, in the numerical problem, target parallel hardware, programming language)

Exam form: This needs a thorough discussion. However, there are readily two possibilities: (1) a standard 4-hour written or digital exam (2) An oral presentation by each student of her/his project and follow-up questions. The second possibility will have a challenge for the teacher(s) when the number of students is large.

Grade: A-F or pass/no pass Needs a discussion together with the exam form.

Suggested textbook (the so-far best textbook): Introduction to High Performance Computing for Scientists and Engineers (Chapman & Hall/CRC Computational Science) Paperback – 7 Jul 2010 by Georg Hager (Author), Gerhard Wellein (Author)

Revising existing courses

Some of these courses could incorporate (or base themselves upon) existing ones. The courses here are organized according to their corresponding disciplines. NOTE: These are suggestions for the discussions within each working group.

- Mathematics

  1. **CS-MATH1**: Data analysis and machine learning (new)
  2. **CS-MATH2**: Mathematical modeling in Science (new)
  3. **CS-MATH3**: Mathematical Foundations of data science (MAT-INF4110 and STK4021?)
  4. **CS-MATH4**: Computational Linear Algebra (MAT-INF4130)
  5. **CS-MATH5**: Computational differential equations (INF5620+INF5631)
  6. **CS-MATH6**: Computational finance (based on STK4520)
  7. **CS-MATH7**: Computational Fluid Mechanics and Navier Stokes equations (INF5670)

- Physical sciences (Astrophysics, geoscience, physics, chemistry and materials science)

  1. **CS-PHYS1**: Computational Physics (FYS3150/4150)
  2. **CS-PHYS2**: Computational Molecular dynamics in life science and materials science (new)
  3. **CS-PHYS3**: Computational Astrophysics (AST9110 and perhaps AST5210?))
  4. **CS-PHYS4**: Computational quantum mechanics (FYS4411)
  5. **CS-PHYS5**: Computational statistical mechanics (FYS4460)
  6. **CS-PHYS6**: Computational Materials Science (new, perhaps use elements from FYS-MENA4111)
  7. **CS-PHYS7**: Computational GeoScience (GEO4510 and GEO4450?)
  8. **CS-PHYS8**: Advanced Computational GeoScience (new or mix of GEO4510 and GEO4450?)

- Bioscience

  1. **CS-BIO1-INF**: Computing for biosciences (INF4490?)
  2. **CS-BIO2-INF**: Introductory bioinformatics (INF4350+MBV3070?)
  3. **CS-BIO3-INF**: Computational Bioinformatics (INF5380?)
  4. **CS-BIO4-INF**: Computational Physiology (INF5560?)
  5. **CS-PHYS2**: Computational Molecular dynamics in life science and materials science (new)

- Computer science
  1. **CS-INF1**: High-Performance Computing and Numerical projects (new)
  2. **CS-INF2**: Problem solving with high level languages (INF4331)
  3. **CS-INF3**: Machine Learning for Image Analysis (NF5850?)
  4. **CS-INF4**: Digital Image Analysis (INF4300?)
  5. **CS-INF5**: Artifical intelligence (INF4820)
- Mechanics
  1. **CS-MECH1**: Computational Mechanics (MEK4250?)
  2. **CS-MECH2**: Advanced Computational Mechanics (MEK4470)

Graduate Certificates

The program plans to offer graduate certificates in

- Three of the courses with label CS-MATH gives a certificate in Computational Mathematics
- Three of the courses with label CS-PHYS gives a certificate in Computational Physics, Astrophysics, Chemistry, Materials Science and Geoscience
- Three of the courses with label CS-BIO gives a certificate in Computational life science.
- Three of the courses with label CS-INF gives a certificate in High-performance computing.

Dual Degrees

The program plans to offer dual degrees (more text to come)