

Summary of course

Morten Hjorth-Jensen Email morten.hjorth-jensen@fys.uio.no^{1,2}

¹Department of Physics and Center of Mathematics for Applications, University of Oslo

²National Superconducting Cyclotron Laboratory, Michigan State University

Nov 28, 2019

What? Me worry? No final exam in this course!



What did I learn in school this year

Our ideal about knowledge on computational science

Does that match the experiences you have made this semester?



Topics we have covered this year

- Linear algebra and eigenvalue problems. (Lecture notes chapters 6.1-6.5 and 7.1-7.5. Projects 1 and 2).
- Numerical integration, Gaussian quadrature and Monte Carlo integration (Lecture notes chapters 5 and 11 and project 3)
- Ordinary differential equations (Lecture notes chapter 8 and project 5)
- Partial differential equations (Lecture notes chapter 10 and project 5)
- Monte Carlo methods (Lecture notes chapters 11, 12, 13 and 14, projects 3, 4 and 5)

Linear algebra and eigenvalue problems, chapters 6.1-6.5 and 7.1-7.5

- Know Gaussian elimination and LU decomposition (project 1)
- How to solve linear equations (project 1)
- How to obtain the inverse and the determinant of a real symmetric matrix
- Iterative solvers (project 5)
- Cubic spline
- Tridiagonal matrix decomposition (project 1)
- Householder's tridiagonalization technique and finding eigenvalues based on this (project 2)
- Jacobi's method for finding eigenvalues (project 2)

Monte Carlo methods (Chapters 11, 12 and 13)

- Numerical integration of multidimensional integrals (project 3)
- Random walks and Markov chains
- Metropolis algorithm (project 4)
- Applications to statistical physics systems (project 4)
- Monte Carlo sampling (project 4 and 5)
- Applications to quantum mechanical systems (project 5)

Ordinary differential equations (Chapter 8)

- Euler's method and improved Euler's method, truncation errors (project 5)
- Runge Kutta methods, 2nd and 4th order, truncation errors
- Verlet algorithm (project 5)
- How to implement and solve a second-order differential equation, both linear and non-linear (project 5).
- How to make your equations dimensionless (projects 2 and 5).

Partial differential equations, chapter 10

- Set up diffusion, Poisson and wave equations up to 2 spatial dimensions and time
- Set up the mathematical model and algorithms for these equations, with boundary and initial conditions. The stability conditions for the diffusion equation.
- Explicit, implicit and Crank-Nicolson schemes, and how to solve them. Remember that they result in triangular matrices (project 5).
- Diffusion equation in two dimensions (project 5).
- Poisson equation in two dimensions (project 5)
- How to make equations dimensionless or scale them properly (project 5)

Learning outcomes and overarching aims of this course

- Develop a critical approach to all steps in a project, which methods are most relevant, which natural laws and physical processes are important. Sort out initial conditions and boundary conditions etc.
- This means to teach you structured scientific computing, learn to structure a project.
- A critical understanding of central mathematical algorithms and methods from numerical analysis. In particular their limits and stability criteria.
- Always try to find good checks of your codes (like solutions on closed form)
- To enable you to develop a critical view on the mathematical models and the specific scientific case under study.

Additional learning outcomes

- you have a thorough understanding of how computing is used to solve scientific problems
- you know some central algorithms used in science
- you have knowledge of high-performance computing elements: memory usage, vectorization and parallel algorithms
- you understand approximation errors and what can go wrong with algorithms
- you have experience with programming in a compiled language (Fortran, C, C++)
- you have experience with debugging software
- you have experience with test frameworks and procedures
- you can critically evaluate results and errors
- you understand how to increase the efficiency of numerical algorithms and pertinent software
- you understand tools to make science reproducible and has a sound ethical approach to scientific problems
- and finally you know how to write a scientific report

Disasters attributable to poor code

Hopefully, after this course, you may be aware of some of the problems below

Have you been paying attention in your numerical analysis or scientific computation courses? If not, it could be a costly mistake. Here are some real life examples of what can happen when numerical algorithms are not correctly applied.

1. [The Patriot Missile failure](#), in Dhahran, Saudi Arabia, on February 25, 1991 which resulted in 28 deaths, is ultimately attributable to poor handling of rounding errors.
2. [The explosion of the Ariane 5 rocket](#) just after lift-off on its maiden voyage off French Guiana, on June 4, 1996, was ultimately the consequence of a simple overflow.
3. [The sinking of the Sleipner A](#) offshore platform in Gandsfjorden near Stavanger, Norway, on August 23, 1991, resulted in a loss of nearly one billion dollars. It was found to be the result of inaccurate finite element analysis.
4. [Read more here](#)

Other courses in Computational Science at UiO

Bachelor/Master/PhD Courses.

- [MAT4110](#) Introduction to Numerical Analysis"
- [MAT3360](#). Introduction to Partial Differential Equations
- [IN5270](#) Numerical methods for PDEs, finite element method
- [FYS4411](#) Computational physics II (Parallelization (MPI), object orientation, quantum mechanical systems with many interacting particles), spring semester
- [FYS4460](#) Computational physics III (Parallelization (MPI), object orientation, classical statistical physics, simulation of phase transitions, spring semester
- [IN3110/IN4110](#) Problem solving with high-level languages (Python), fall semester
- [IN3200/4200](#), High-Performance Computing and Numerical Projects. Spring semester

New Courses in Computational Science and Data Science

1. [STK2100 Machine learning and statistical methods for prediction and classification.](#)
2. [IN3050/IN4050 Introduction to Artificial Intelligence and Machine Learning.](#) Introductory course in machine learning and AI with an algorithmic approach.
3. [STK-INF3000/4000 Selected Topics in Data Science.](#) The course provides insight into selected contemporary relevant topics within Data Science.
4. [IN4080 Natural Language Processing.](#) Probabilistic and machine learning techniques applied to natural language processing.
5. [STK-IN4300 – Statistical learning methods in Data Science.](#) An advanced introduction to statistical and machine learning. For students with a good mathematics and statistics background.
6. [IN-STK5000 Adaptive Methods for Data-Based Decision Making.](#) Methods for adaptive collection and processing of data based on machine learning techniques.
7. [IN5400 Machine Learning for Image Analysis.](#) An introduction to deep learning with particular emphasis on applications within Image analysis, but useful for other application areas too.
8. [TEK5040 – Dyp læring for autonome systemer.](#) The course addresses advanced algorithms and architectures for deep learning with neural networks. The course provides an introduction to how deep-learning techniques can be used in the construction of key parts of advanced autonomous systems that exist in physical environments and cyber environments.

Additional courses of interest

1. [STK4051 Computational Statistics](#)
2. [STK4021 Applied Bayesian Analysis and Numerical Methods](#)

New Master of Science program at UiO in Computational Science from Fall 2018

The program aims at offering thesis projects in a variety of fields. The scientists involved in this program can offer thesis topics that cover several disciplines.

- 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

The thesis projects will be tailored to the your interests, wishes and scientific background. The projects can easily incorporate topics from more than one discipline.

The program opens up for flexible backgrounds

While discipline-based master's programs tend to introduce very strict requirements to courses, we believe in adapting a computational thesis topic to the student's background, thereby opening up for students with a wide range of bachelor's degrees. A very heterogeneous student community is thought to be a strength and unique feature of this program. Most study directions have a minimum course requirement of 120 ECTS (European Credit Transfer System) at the undergraduate level (bachelor degree or equivalent) in Astrophysics, bioscience, chemistry, computer science and informatics, geoscience, mathematics, materials science, mechanics and physics. Of these 120 ECTS, 40 ECTS have to include basic mathematics and programming courses, equivalent to the University of Oslo mathematics courses MAT1100, MAT1110, MAT1120 and at least one of the corresponding computing and programming courses INF1000/INF1110 or MAT-INF1100/MAT-INF1100L/BIOS1100/KJM-INF1xxx. The remaining 80 ECTS have to be within at most two of the fields of astrophysics, bioscience, chemistry, computer science and informatics, geoscience, mathematics, materials science, mechanics and physics. 40 of these 80 ECTS have to be advanced undergraduate courses at the 2000 and 3000 level and a minimum of 20 ECTS must be at the 3000 level within physics/material science/mechanics/astrophysics/informatics/mathematics/bioscience/chemistry/geoscience.

New Master of Science program at UiO in Data Science from Fall 2018

Three specializations

- Statistics and Machine Learning

- Database Integration and Semantic Web
- Data Science and Life Science

Best wishes to you all and thanks so much for your heroic efforts this semester

