# Summary of course

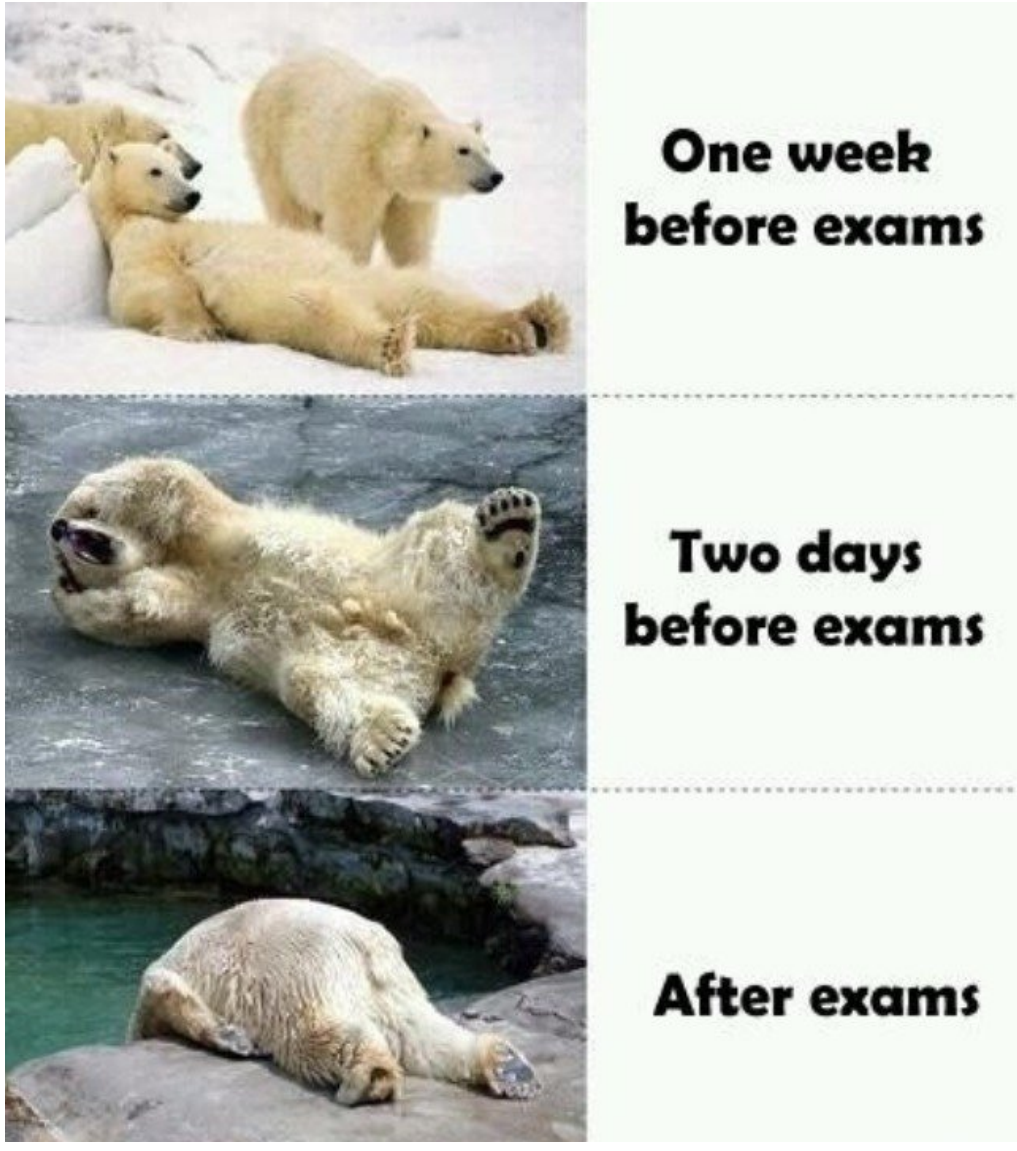
Morten Hjorth-Jensen Email morten.hjorth-jensen@fys.uio.no[1,2]

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

National Superconducting Cyclotron Laboratory, Michigan State University[2]

Aug 23, 2017

# 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?


©Zits Partnership

# Topics we have covered this year

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

- Random walks and Markov chains
- Metropolis algorithm (projects 4 and 5)
- Applications to statistical physics systems (project 4)
- Applications to quantum mechanical systems (project 5)

# Ordinary differential equations (Chapter 8)

- Euler's method and improved Euler's method, truncation errors (project 3)
- Runge Kutta methods, 2nd and 4th order, truncation errors
- Leap-frog and Verlet algoritm (projects 3 and 5)
- How to implement and solve a second-order differential equation, both linear and non-linear (projects 3 and 5).
- How to make your equations dimensionless (projects 2, 3, 4 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).

# 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

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

# Other courses in Computational Science at UiO

## Bachelor/Master/PhD Courses

- INF-MAT4130 Numerical linear algebra
- MAT-INF4300, PDEs and Sobolev spaces I
- MAT-INF4310, PDEs and Sobolev spaces II
- MAT-INF3360 - Introduction to Partial Differential Equations
- INF5620 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
- INF3331/4331 Problem solving with high-level languages (Python), fall semester
- INF3380 Parallel computing for problems in the Natural Sciences (mostly PDEs), spring semester

# 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. The program has 60 study places and the study directions are

- Computational Science: Applied Mathematics and Risk Analysis (14)
- Computational Science: Astrophysics (2)
- Computational Science: Bioinformatics (5)
- Computational Science: Bioscience (4)
- Computational Science: Chemistry (4)
- Computational Science: Geoscience (5)
- Computational Science: Imaging and Biomedical computing (8)
- Computational Science: Material Science (2)
- Computational Science: Mechanics (8)
- Computational Science: Physics (8)
- Computational Science: Economy and Finance (not clear yet)

# 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

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