Slides PHY 480 and PHY 905 Lectures, introduction to the course

Phil Duxbury[1]

Morten Hjorth-Jensen[2,3]

[1]Department of Physics and Astronomy, Michigan State University
[2]Department of Physics, University of Oslo
[3]Department of Physics and Astronomy and National Superconducting Cyclotron Laboratory, Michigan State University

Spring 2016

Overview of first week

- Thursday: First lecture: Presentation of the course, aims and content
- Thursday: Second Lecture: Introduction to C++ programming and numerical precision. Exercises for first week.
- Friday: Numerical precision and C++ programming, continued and exercises for first week (chapter 2 of lecture notes)
- Numerical differentiation and loss of numerical precision (chapter 3 lecture notes)
- Computer lab: Thursday and Friday. First time: Thursday and Friday this week, Presentation of hardware and software at room FV329 first hour of every labgroup and solution of first simple exercises. The first two weeks we focus on simple programming exercises and to set up github and QTcreator.

Lectures and ComputerLab

- Lectures:
- Weekly reading assignments needed to solve projects.

- First hour of each lab session may be used to discuss technicalities, address questions etc linked with projects.
- Detailed lecture notes, exercises, all programs presented, projects etc can be found at the homepage of the course.
- Computerlab:
- Weekly plans and all other information are on the official webpage.
- Final oral exam:

Course Format

- Several computer exercises, 4 compulsory projects. Electronic reports only using Git for repository and all your material.
- Evaluation and grading:
- The computer lab consists of xxx Linux PCs, but many prefer own laptops. C/C++ is the default programming language, but Fortran2008 and Python are also used. All source codes discussed during the lectures can be found at the webpage and github address of the course. We recommend either C/C++, Fortran2008 or Python as languages.

ComputerLab

day	teacher
Group 1: Thursday 10am-2pm	MHJ
Group 2: Thursday 2pm-6pm	MHJ
Group 3: Friday 10am-2pm	MHJ
Group 4: Friday 2pm-6pm	MHJ

Topics covered in this course

- Numerical precision and intro to C++ programming
- Numerical derivation and integration
- Random numbers and Monte Carlo integration
- Monte Carlo methods in statistical physics

- Quantum Monte Carlo methods
- Linear algebra and eigenvalue problems
- Non-linear equations and roots of polynomials
- Ordinary differential equations
- Partial differential equations
- Parallelization of codes
- High-performance computing aspects

Syllabus

Linear algebra and eigenvalue problems, chapters 6 and 7.

- Know Gaussian elimination and LU decomposition
- How to solve linear equations
- How to obtain the inverse and the determinant of a real symmetric matrix
- Cholesky and tridiagonal matrix decomposition

Syllabus

Linear algebra and eigenvalue problems, chapters 6 and 7.

- Householder's tridiagonalization technique and finding eigenvalues based on this
- Jacobi's method for finding eigenvalues
- Singular value decomposition
- Qubic Spline interpolation

Syllabus

Numerical integration, standard methods and Monte Carlo methods (chapters 4 and 11).

- Trapezoidal, rectangle and Simpson's rules
- Gaussian quadrature, emphasis on Legendre polynomials, but you need to know about other polynomials as well.
- Brute force Monte Carlo integration
- Random numbers (simplest algo, ran0) and probability distribution functions, expectation values
- Improved Monte Carlo integration and importance sampling.

Syllabus

Monte Carlo methods in physics (chapters 12, 13, and 14).

- Random walks and Markov chains and relation with diffusion equation
- Metropolis algorithm, detailed balance and ergodicity
- Simple spin systems and phase transitions
- Variational Monte Carlo
- How to construct trial wave functions for quantum systems

Syllabus

Ordinary differential equations (chapters 8 and 9).

- Euler's method and improved Euler's method, truncation errors
- Runge Kutta methods, 2nd and 4th order, truncation errors
- How to implement a second-order differential equation, both linear and non-linear. How to make your equations dimensionless.
- Boundary value problems, shooting and matching method (chap 9).

Syllabus

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. Their stability conditions.
- Explicit, implicit and Crank-Nicolson schemes, and how to solve them. Remember that they result in triangular matrices.
- How to compute the Laplacian in Poisson's equation.
- How to solve the wave equation in one and two dimensions.

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/analytical form)
- To enable you to develop a critical view on the mathematical model and the physics.

And, there is nothing like a code which gives correct results!!

- J. J. Barton and L. R. Nackman,*Scientific and Engineering C++*, Addison Wesley, 3rd edition 2000.
- B. Stoustrup, *The C++ programming language*, Pearson, 1997.
- H. P. Langtangen INF-VERK3830 http://heim.ifi.uio.no/~hpl/INF-VERK4830/
- D. Yang, *C++ and Object-oriented Numeric Computing for Scientists and Engineers*, Springer 2000.

Extremely useful tools, strongly recommended

and discussed at the lab sessions.

- GIT for version control and hand in of projects, short demo at computer lab plus video

- ipython notebook
- QTcreator for editing and mastering computational projects (for C++ codes, see webpage of course), demonstration at lab
- Armadillo as a useful numerical library for C++, highly recommended, demonstration at lab plus video
- Unit testing libraries, to be discussed later