

Computational Physics Spring 2010

Compulsory course for the Master Program in Computational physics

<http://www.physto.se/english/student/coursefacts.php?kurs=FK8002>

Course web page

http://www.atom.physto.se/~lindroth/comp08/comp_phys_10.html

Teacher: Eva Lindroth, lindroth@physto.se, 08-5537 8616, room C4:1001

Aim

This course focuses on the solution of physical problems through application of numerical methods and computing. You will train your ability to:

- transform a physical problem to a suitable form for numerical treatment,
- find and apply a suitable numerical approach and
- analyze the calculated result.

The core of the course is the assignments and projects. During the first part of the course you will get a new assignment every week. Towards the end of the course you will have to do a slightly bigger project. The lectures will prepare for the coming assignment and allow time to discuss problems with the ongoing assignment. Erik Waltersson will give you hands-on help and advice.

Prerequisites

Prior knowledge required for this course includes the material covered by the compulsory courses for the Master program in Computational physics:

[*Programming and computing for physicists*](#)

[*Numerical Methods for Physicists II*](#)

[*Mathematical Methods in Physics*](#)

as well as Quantum Mechanics, e.g. as given by the undergraduate courses *Quantum Physics* FK5011 and FK5012:

Examination

Examination is by assignments and projects. All assignments have to be handed in at a given **deadline†**. In the end of the course you will have to do a slightly bigger project. You can choose to work together with one other student or by yourself†. You should be prepared to present and discuss your solutions in class. **This is part of the examination.**

How to report on the Assignments

Your solutions should be handed in on paper. They should be clearly written, easy to follow and accompanied by the program code. Also the code should be well written. This means that it should be commented and easy to follow. In most cases you will work with physics problem. Then it is important that your

solutions, illustrating graphs etc. are displayed in appropriate units. Although you might choose to let the program work with some reduced units you should present your result in physically relevant units. All information you might need on constants and units can be found at: <http://physics.nist.gov/PhysRefData/contents.html>. For the longer project you should hand in a written report and give a short oral presentation.

Programming language

You can use the language of your choice. However, you cannot get help from us (your teachers) with all languages. Between the two of us we can help you with Mathematica, MathLab, or various versions of Fortran. Some of the assignments can be carried through with MathLab, and perhaps even with Mathematica, but there will also be assignments where program languages such as Fortran or C/C++ will be more practical.

Where to find ideas and information

- Several numerical tools which you now will put to work have been discussed in the previous course Numerical Methods II: <http://www.csc.kth.se/utbildning/kth/kurser/DN2251/>. Remember to consult your material from this course: L. Edsberg "Introduction to computation and modelling for Differential equations" and A. Ruhe "Topics in Linear Algebra"
- Remember to use: Arfken & Weber: Mathematical Methods for Physicists (Elsevier Academic Press) as a *handbook* in mathematical methods.
- A good handbook for scientific programming is; *Numerical Recipes: The Art of Scientific Computing, Third Edition* (2007) [William H. Press](#), [Saul A. Teukolsky](#), [William T. Vetterling](#), [Brian P. Flannery](#) Cambridge University Press (ISBN-10: 0521880688, or ISBN-13: 978-0521880688). <http://www.nr.com/> This Third Edition has software in C++ (which you get on file if you buy the CD). **Free on-line versions** (of the text - not the software) of older editions of Numerical Recipes are available: <http://www.nrbook.com/a/>
- A program library that is generally available is LAPACK - Linear Algebra PACKage (in many cases other versions of algorithms discussed in Numerical Recipes are available here). For example you have access to Lapack at our computers. <http://www.netlib.org/lapack/index.html> The original version is written in Fortran77 and provides routines for solving systems of simultaneous linear equations, least-squares solutions of linear systems of equations, eigenvalue problems, and singular value problems. There are interfaces for other languages: Fortran95: <http://www.netlib.org/lapack95/> C/C++ <http://www.netlib.org/lapack++/> ,

<http://math.nist.gov/tnt/>

- There are many on-line resources for programming languages. At the home page of the course “Introduction to High-Performance computing” (Recommended optional courses for the master program in computational physics), you find both Fortran90/95 and C++ “by examples”, see: http://www.pdc.kth.se/systems_support/training/2008/summerschool/coursework/handouts.html#proglang. A general source for information about syntax etc. for Fortran is for example: <http://www.lahey.com/other.htm>.

Assessment Criteria

The grading is from A-F.

The ability to choose and apply numerical methods to physical problems is graded based on the reports and programs written for the assignments and projects. Several aspects are then taken into accounts; reliability, efficiency, presentation (written and oral) and documentation.

A - Excellent

B - Very Good

C - Good

D - Satisfactory

E - Passed

Fx - Insufficient

F - Failed

†

In order to pass the course (grades A-E) you should hand in all your assignments before or at their deadlines. Since unexpected things might happen, such as illnesses etc, we are aware that there are occasions when a student cannot meet the deadline. If this happens just at one occasion you might still pass the course. However, only students that hand in *all* their assignments before the deadlines have the possibility to achieve grades A-C. During the class following the deadline we will discuss different solution possibilities and you will be asked to present your solution. Students who do not meet the deadlines cannot participate in such discussions in a meaningful way. This is the reason for these rather strict rules. The same rules apply to projects in the end of the course.