Course syllabus

Avancerad kurs i numeriska algoritmer med Python/SciPy
Advanced Course in Numerical Algorithms with Python/SciPy

FMNN25, 7,5 credits, A (Second Cycle)

Valid for: 2016/17  
Decided by: Education Board B  
Date of Decision: 2016-03-29

General Information

Elective for: D4, E4-pv, F4, F4-bs, Pi4-bs  
Language of instruction: The course will be given in English on demand

Aim

The course is intended as an algorithm oriented complement to most of the basic and specialized courses in numerical analysis, which are focused on analysis of methods. The course emphasizes the coupling between complex numerical algorithms and modern programming languages.

Learning outcomes

Knowledge and understanding

For a passing grade the student must

• be able to account for the basic principles of computational algorithms.  
• be able to describe a number of important computational problems, and ways to attack them, in detail.

Competences and skills

For a passing grade the student must

• be able to write computational programmes at a high level.  
• be able to code, test and evaluate the results of complex numerical algorithms, using established programme libraries.  
• be able to carry out a group programming project, including identifying subproblems,
distribution of tasks within the group and responsibility for the completion of his/her task.
• be able to account for a computational project, both in a written report and orally.

Contents

Introduction to Python for students already familiar with another programming language. The use of object oriented programming in scientific computing. Scipy/Numpy datastructures.

Examples of complex numerical algorithms from varying subjects in numerical analysis, Coupling to advanced numerical libraries in C and Fortran (Netlib).

Automatic tests in scientific computing. Graphical representation of mathematical results (animation). The use of Python to control system processes.

The course may be complemented with special contributions of invited guest teachers.

Examination details

Grading scale: UG
Assessment: Weekly programming assignments. A larger programming project to be carried out in group, with a written report to be presented at a seminar. Opposition on the report of another group. Compulsory attendance at all presentations.

Admission

Required prior knowledge: Basic course in numerical analysis. Programming experience in some of the languages Java, C, C++, Fortran, Python and Matlab.
The number of participants is limited to: No

Reading list


Contact and other information

Course coordinator: Claus Führer, claus.fuhrer@na.lu.se
Course administrator: Patricia Felix Poma de Kos, patricia.felix_poma_de_kos@math.lth.se
Course homepage: http://www.maths.lth.se/na/courses/FMNN25/