Introduction to scientific computing with Python (3 credits)

Lecturer: Dr. Scott K. Hansen
1-2-5066

This is a hands-on introduction to scientific computing for earth scientists, consisting of a series of two-week modules, each covering a different topic. Each module incorporates weekly lectures and computer lab sessions during which students work on a relevant programming assignment (upon which their grades will be based). The course aims to familiarize students with the premiere scientific programming language ecosystem, that of Python, and via hands-on application to currently relevant topics in simulation, inference, numerical analysis and data analysis.

Course outline

Module 1: Introduction to scientific computing with Python and Numpy [2 weeks]
A gentle introduction to the usage of Python for scientific computing. Basic programming and data exploration are covered, as well as ways to improve performance with vectorization and parallelization.

Module 2: Further exploring the Python library ecosystem [2 weeks]
Several key scientific libraries that interoperate with Numpy are introduced: Scipy, Matplotlib, and Pandas. Their usage is discussed and Jupyter notebooks are introduced.

Module 3: Classical data analysis [2 weeks]
The use of the Python ecosystem for classic statistical analysis tasks is covered. Topics include single and multi-variate regression, principal component analysis, and advanced visualization.

Module 4: Numerical solution of equations [2 weeks]
The focus of this module is on numerical solution of algebraic and differential equations. Students learn numerical root finding, how to manually solve a simple differential equation with finite differences, and may be introduced to the FEniCS system for automated finite element solution, time permitting.

Module 5: Introduction to data science and machine learning [2 weeks]
The Scikit-Learn Python machine learning framework is introduced, and its usage for data clustering, classification, and dimensionality reduction discussed.

Module 6: Bayesian analysis and Monte Carlo simulation [2 weeks]
Monte Carlo simulation techniques, including numerical particle tracking, are covered. Bayesian model calibration is introduced, and students learn how to infer model parameters using MCMC.

Epilogue: Introduction to GPU computing [1 week]
Problems in which GPU computation radically improves performance relative to traditional CPU computation are discussed, and examples using the Numba compiler for Python.

Structure of final grade

Evaluation will be based on periodic programming assignments, from which students select a subset of problems most relevant to their work and interests.