Vědecké programování v Pythonu (Scientific Programming in Python)

Anotace:

Cílem tohoto kurzu je osvojení základů moderního programovacího jazyka Python se zaměřením na vědecké výpočty. Důraz je kladen na efektivní řešení reálných problémů. Výuka probíhá interaktivně a formou praktických cvičení, jejichž obsah může být přizpůsoben obsahu dalších předmětů nebo tématům studentských prací. Studenti jsou rovněž zapojováni do probíhajícího výzkumu. V úvodní části kurzu se studenti seznámí se základními vlastnostmi jazyka Python – od základních typů až po objektově orientované nebo funkcionální programování. Větší část kurzu je věnována specifickým vlastnostem Pythonu pro vědecké programování. Prezentovány jsou hlavní numerické knihovny NumPy, SciPy a grafická knihovna Matplotlib. Ukážeme, jak tvořit efektivní kód, jak lze Python kombinovat s jinými jazyky, jaké nástroje využívat.

Požadavky:

Povinné: Není požadováno předchozí absolvování žádného konkrétního předmětu
Doporučené: Základní znalost nějakého programovacího jazyka (C/C++, Fortran, Matlab, Java, Pascal apod.), základy numerických metod

Osnova cvičení:

Samostatné, případně skupinové programování konkrétních úloh s použitím získaných znalostí. Součástí cvičení bude i použití větších simulačních kódů a knihoven, případně práce na aktivně řešeném výzkumném projektu. Studentům bude rovněž dán prostor pro řešení úloh spojených s další výukou nebo jejich bakalářskou či diplomovou prací.

Osnova (a sylabus):

1. Úvod do Pythonu – základní vlastnosti a nástroje, konvence, datové typy, podmínky, funkce
2. Kontejnery a (im)mutable typy, iterátory, generátory
3. Funkcionální a objektově orientované programování, moduly
4. Výjimky, unit testy, Python debugger, základní moduly
5. Kompletní projekt v Pythonu – konvence, dobré praktiky, dostupné nástroje, dokumentace (Sphinx), distribuce balíků
6. Úvod do NumPy – třídy ndarray, základní operace, polynomy
7. Grafický výstup – Matplotlib, čtení a zápis z/do souborů

8. Pokročilá práce s NumPy – specifika ndarray, další třídy (matrix, masked array), lineární algebra
9. Úvod do SciPy a SymPy
10. Optimalizace numerických výpočtů – vektorizace, profilování, Cython, f2py
11. Paralelní výpočty – vlákna, procesy, message passing

Studijní materiály:

Povinná literatura:
1. V. Haenel, E. Gouillart, G. Varoquaux: Python Scientific Lecture Notes, http://scipy-lectures.github.com
2. H.P. Langtangen: A Primer on Scientific Programming with Python

Doporučená literatura:
3. H.P. Langtangen: Python Scripting for Computational Science
4. M. Pilgrim: Dive Into Python 3, http://getpython3.com/diveintopython3
5. Z.A. Shaw: Learn Python The Hard Way, http://learnpythonthehardway.org

Web:

server.ipp.cas.cz/~urban/PythonLectures

Klíčová slova:

Programování, Python, numerické výpočty, počítačová algebra.

Rozsah a obsahové zaměření individuálních stud. prací a způsob kontroly:

Zápočet je podmíněn zpracováním kompletního projektu v Pythonu, který je třeba obhájit před vyučujícím a ostatními studenty. Projekt je zadán v průběhu semestru, cíle mohou být přizpůsobeny studijnímu zaměření studenta.

Cíle studia:

Znalosti: Základy jazyka Python, vlastnosti Pythonu pro řešení vědeckých úloh, přehled o dostupných nástrojích.
Schopnosti: Efektivní návrh a implementace vědecké úlohy v jazyce Python, schopnost vyhledat a použít dostupné nástroje.

Synopsis:

The aim of this course is to learn the fundamentals of the modern Python programming language with a focus on scientific computing. Emphasis is placed on effective solutions to real problems. The course is performed in an interactive form of practical exercises, whose topics can be tailored to the content of other subjects or student theses. Students are also involved in ongoing research. In the introductory part of the course, students learn the basic features of Python—from basic types to object oriented or functional programming. The greater part of the course focuses on specific features of Python for scientific programming. Presented are the main numerical libraries NumPy, SciPy and the Matplotlib graphics library. We show how to generate efficient code, how to combine Python with other languages, what tools are available.

Prerequisites:

Mandatory: No particular subject needed for qualification
Recommended: Practical knowledge of at least one suitable programming language (C/C++, Fortran, Matlab, Java, Pascal, etc.), knowledge of basics of linear algebra and numerical methods (1st term level)

Outline and Sylabus of Excercises:

Individual or group specific programming tasks using the acquired knowledge. Exercises will use larger simulation codes and libraries, or work on ongoing research projects. Students will also be given space to deal with problems associated with their further education or bachelor's or master's theses.

Outline and Sylabus:

1. Introduction to Python - basic features and tools, conventions, data types, conditions, functions
2. Containers and (i)mutable types, iterators, generators
3. Functional and object-oriented programming, modules
4. Exceptions, unit tests, Python debugger, core modules
5. Complete project in Python - conventions, good practices, documentation, available tools, documentation (Sphinx), package distribution
6. Introduction to NumPy - class ndarray, basic operations, polynomials
7. Graphical output - Matplotlib, reading and writing from / to files
8. Advanced work with NumPy – specifics of ndarray and other classes (matrix, masked array), linear algebra

9. Introduction to SciPy and SymPy
10. Optimization of numerical calculations - vectorization, profiling, Cython, f2py
11. Parallel Computing - threads, processes, message passing

Source Materials:

Key references:

1. V. Haenel, E. Gouillart, G. Varoquaux: Python Scientific Lecture Notes, http://scipy-lectures.github.com
2. H.P. Langtangen: A Primer on Scientific Programming with Python

Recommended references:

3. H.P. Langtangen: Python Scripting for Computational Science
4. M. Pilgrim: Dive Into Python 3, http://getpython3.com/diveintopython3
5. Z.A. Shaw: Learn Python The Hard Way, http://learnpythonthehardway.org

Equipment:
Computer laboratory with UNIX/Linux OS and Python installed

Keywords:

Programming, Python, numerical computation, computer algebra.

Goals of Study:

Knowledge: Basics of Python, Python properties for solving scientific problems, an overview of available tools.
Skills: Effective design and implementation of scientific tasks in Python, the ability to find and use available tools.