CERI 7102/8102, CIVL 7002/8002: Spring 2022

Programming Tools in Python

Instructor:
Thomas Goebel, thgoebel@memphis.edu
Lectures: TR, 11:20 a.m. - 12:45 p.m., CERI
Location: Earthquake Ctr house 4 (Long Building), Computer Lab
Office Hours: T, 2-3 p.m., Earthquake Ctr house 2, room 107

Course Description: The ability to analyze data and construct numerical models using a scientific programming language (R, perl, python, matlab, fortran... something beyond MS excel) is an essential skill for any engineer and scientists. Data science has far reaching applications beyond academic research and is a highly-demanded skill.

This course provides an introduction to scientific computing in Python, starting with: What is an interpreted programming language and how do I use it? and ending with What are finite differences and how do I solve PDEs? The aim of the class is to introduce students to a wide set of tools that can be used to solve problems in pure and applied sciences. The introductory character of the class limits the amount of time spent on each topic. This should hopefully not lead to frustration but rather encourage each student to dive deeper into areas of personal interest. A large part of the final grade will be the final project which encourages students to spend several weeks to develop some elegant code to solve a numerical or data analysis problem. The class is designed to leave enough room for you to develop critical thinking and analysis skills within the framework of the Python programming language.
From Learn Python the Hard Way by Zed A. Shaw: *"Programming as a profession is only moderately interesting. It can be a good job, but you could make about the same money and be happier running a fast food joint. You're much better off using code as your secret weapon in another profession. People who can code in the world of technology companies are a dime a dozen and get no respect. People who can code in biology, medicine, government, sociology, physics, history, and mathematics are respected and can do amazing things to advance those disciplines."*

Course Objectives:
At the completion of this course, students will be able to:

1. Create scripts and functions for quantitative data analyses
2. Solve basic optimization problems
3. Create figures and animations
4. Construct numerical models using finite-difference methods
5. Invert for model parameters and fit functions to observed data
6. Use basic machine learning algorithms

Text Book and Resources

- **A Primer on Scientific Programming with Python by Hans Langtangen**
  -https://www.springer.com/gp/book/9783642549595#otherversion=9783642549588
  (Texts in Computational Science and Engineering) 5th ed. 2016 Edition

- For a more detailed text book see also: Programming for Computations - Python
  Hans Petter Langtangen and Svien Linge, 5th ed. 2016

- Google's free python class https://developers.google.com/edu/python/?csw=1

- Python style guide https://docs.python-guide.org/writing/style/

Software Requirements

Ideally, students will bring their on laptops to complete the assignments for this course but desktop computers will be available during class hours. Install the Anaconda python package on your computer (make sure to install **Anaconda3 for python 3.7**, the newer python3 version is not compatible with python2 (change in syntax etc.) and you may have trouble running some of the examples with python2.
Anaconda: https://www.anaconda.com/distribution/
PyCharm: https://www.jetbrains.com/pycharm/
Atom: https://atom.io/packages/ide-python
I highly recommend using an Integrated Development Environment (IDE). Examples are: **spyder, pycharm, eclipse, atom, IDLE and visual studio**. You can start with using a jupyter notebook but more complex projects are easier managed through an IDE. IDEs naturally help with developing good habits and portable code.

Preparation before class:

1. Install anaconda3, use python3.7.
2. Select and install your favorite development environment. Spyder comes with the Anaconda package and IDLE should be included with any standard system python installation. You will have to install other IDEs if you prefer to use them.
3. Bring your laptop to class every day and make sure it is fully charged.

Grading

Homework	70%
Final Project	30%

Homework and term paper topics may differ depending on course registration (7002 vs 8002 and 7102 vs. 8102).

Tentative Course Outline

Introduction to python and quantitative computing

- Week 1
  - Programming and Python Basics, command-line operations, IDEs, python modules and packages
  - in class activity: inC_1_1, inC_1_2
- Week 2
  - Python variables: lists, arrays and dictionaries, Python scripts and best practices in programming
  - If, while and for-loops, Indexing, vector and matrix operations, functions, *Data I/O-1*
  - Random variables and synthetic data
  - in class activity: inC_2_1

  Homework 1 *Functions and for-loops*
- Week 3
  - Best practices, Error handling, version control, benchmarking: ex. derivatives
  - Modules and packages, code repositories on github,
  - inC_5_1

Numerical analysis and differential equations

- Week 4
  - Numerical derivatives, root finding, optimization problems
  - inC_3_1

  Homework 2 *Roots*
- Week 5
  - Numerical integration and inverse functions

  Homework 3 *Numerical Integration*
- Week 6
  - ODEs, direction fields, Euler Formula and Runge-Kutta
  - inC_5_2
- Week 7
  - Higher order ODEs and systems of ODEs

  Homework 4 *ODEs*
- Week 8
  - Spring Break - March 7th - 11 th
- Week 9
  - Partial differential equations 1 - Diffusion equation
  - inC_6_1

  Homework 5 - *1D Heat Equation*
- Week 10
  - Partial differential equations 2 - Wave equation

Data analysis and machine learning

- Week 11
  - Working with data files, plotting, animations and geo-referenced maps, *Data I/O-2*
  - inC_7_1
  Homework 6 - *Earthquake data*
- Week 12
  - Basic model fitting, least-squares and power-laws
  - inC_8_1
  Homework 7 - *Aftershock fitting*
- Week 13
  - Machine learning 1: Types of ML algorithms, confusion matrix
  -inC_9_1 - Decision Trees
- Week 14
  - Machine learning 2: Multi-Layer Perceptron
  Homework 8 - *Image Recognition* (due 4/27)

Final Projects

- Week 15 - work on final projects

Course Policies

- **General**
  - **All submitted code must contain detailed comments or you will lose points.**
  - You are allowed to work in groups (3 students max. per group) for in-class and homework problems. Make sure that every group member contributes. The final project has to be solved by each student individually.
- **Attendance**
  - Attendance is expected for each lecture.
- **Homework**
  - Homework assignments will be posted on Canvas. Homework submissions are due every Monday before midnight.
  - We will discuss the homework and one group will present their solutions.
  - Points will be reduced by 30% on late assignments.
- **Final Project**
  The final project is a significant part of your final grade. You will have the opportunity to apply much of what you have learned in class. The final project has two components:
  1. Solve a specific problem (TBD) and develop well-documented code in Python.
  2. Write a short paper on your project.

  The final project will focus on (i) data analysis and statistics applied to scientific data (ii) Machine Learning applied to earth science or engineering data sets (iii) Solve a specific ODE or PDE. Students are encouraged to work on topics directly related to their research/dissertation. More details about the final project will be discussed toward the end of the class.

Academic Honesty in Programming

Make sure you accurately document and reference every piece of your code that is taken from other sources. If you copy and paste code, you will have to include detailed citations e.g.:

```
# This line is taken from http://www.url.com/neat/programming/idea
```

More information about academic integrity in computer programming classes can be found here: http://www.cs.cornell.edu/courses/cs1133/2020sp/about/integrity.php.