EML4930/EML6934 — Python Programming — Fall 2017
1 Credit — Thursday 10:40 - 11:30

Limited seats available — enroll now!

Instructor: Charles Jekel — cjekel@ufl.edu — `http://jekel.me`

July 3, 2017

- **free** and **open source**
- viable replacement of MATLAB for everyday use
- for numerical and scientific research
- exceptional with data

Course Description Python is a general purpose programming language. Course covers the basics, linear algebra, plotting, and more to prepare students for solving numerical problems with Python. Python is a viable **free** and **open** alternative to MATLAB. Prerequisite: COP 2271 MATLAB or equivalent.

About Python Python is a powerful language for scientist and engineers that is free and open. Python has become particularly popular for data analysis, and is used everyday by top scientist and engineers. Everything done in MATLAB, can also be done in Python. In 2016 IEEE SPECTRUM rated Python as the third most popular programming language. How long can you stay competitive without knowing Python?

Python is used for a variety of purposes. Often Python is used as a simple language to manipulate and pass data to other more powerful and faster codes. This can be done either as a wrapper to fast Fortran or C++ code, or in popular software packages such as Abaqus or LAMMPS. Python becomes a powerful numerical analysis tool applicable to many researchers, with the inclusion of libraries like NumPy[1], SciPy[2], and Matplotlib[3]. My absolute favorite thing about working in Python is just the shear number of libraries available. Libraries can range from various APIs to the state-of-the-art machine learning libraries (such as TensorFlow[4] and Theano[5]).

Python syntax is beautiful and elegant. Just take a look at the following code which prints lines from a text file. The Python language emphasizes readability, making Python a great language for collaboration.

```
with open('myFilename.txt') as f:
    lines = f.readlines()
    for line in lines:
        print line
```

[1] NumPy is the fundamental package for scientific computing with Python. `http://www.numpy.org/`

[2] SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering. `https://www.scipy.org/`

[3] Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. `https://matplotlib.org/`

[4] An open-source software library for Machine Intelligence. `https://www.tensorflow.org/`

[5] Theano is a Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. `http://deeplearning.net/software/theano/`

My Experience with Python I use Python extensively in my research. In fact, Python has become a part of my everyday life. First Python is a great alternative to MATLAB. Since Python is free, students won't run into licensing issues which may happen with heavy use of MATLAB towards the end of semesters. I do all of my numerical analysis in Python, and I find Python particularly useful for numerical optimization with external programs. My Python use includes running a C based optimizer while editing and executing non-linear finite element routines on the HPC. Additionally I've used Python on numerous projects ranging from a bot that tweets when the power goes off, to developing my own virtual strain gauge library.

Python allows me to implement new ideas faster than with other programming languages. I've done extensive work in Python related to numerical optimization, surrogate modeling, and machine learning. I want to help MAE graduate students by providing them some guided exposure to Python.

The Proposed Course I am teaching a one credit special graduate course on Python which will prepare students for using Python for their everyday numerical work. As it stands, there are few alternatives to learning Python with access to an instructor, and learning is always easier with guidance. I can provide students with libraries and tricks relevant to their research in the MAE department.

Tentative lecture outline There will be 14 fifty-minute lectures. Here is the tentative schedule for lecture topics.

0. About Python (2 vs 3), IDEs, IPython[6], notebooks, and installation
1. Basics: data types, math, loops
2. Python classes, objects, namespace
3. Python libraries and pip
4. Numpy[1] and Matrix operations
5. More Numpy[1] and Matplotlib[3] for 2D plots
6. Contour plots, 3D plot, Histograms
7. Statistical distributions and functions
8. Optimization in Scipy[2]
9. Python read and write: opening and modifying text/csv files
10. Symbolic math with SymPy[7], DOE with pyDOE
11. Scikit-learn[8] surrogate modeling
12. Scikit-learn[8]: surrogate modeling and machine learning
13. Pandas[9] and DataFrames

[6]IPython provides a rich architecture for interactive computing with Python. `https://ipython.org/`

[7]SymPy is a Python library for symbolic mathematics. `http://www.sympy.org`

[8]Machine Learning in Python. `http://scikit-learn.org`

[9]*pandas* is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. `http://pandas.pydata.org/`