UNIVERSIDAD NACIONAL DE COLOMBIA

Introduction to Computational Electromagnetics (2025782)

Python Installation and Package Management

Benjamin Joseph Mailhé
bmailhe@unal.edu.co

Python Install

Depending on the OS installed on your PC, please select the correct installer from the official Python page:

https://www.python.org/downloads/

It is recommended to use the last version of Python (3.8.5) or at least Python 3.7, since drastic changes were realized between this version and previous ones.

Indeed , using an older version such as Python 2.7 might result in errors when trying to reproduce the code we will be developing together, or incompatibilities with libraries.

If you need some help to install Python, you can refer to the following documentation:

https://docs.python.org/3/using/index.html

Remember that if you are using a UNIX-based system, such as Linux, you can easily install Python through the command line or application store. Please refer to your distribution documentation for more information.

Also, keep in mind that it might be useless since Linux generally integrates Python in its core package. You can verify the existence of Python and its version using the following command:

```
python --version
```

You might find that various versions are installed. If it is the case, please follow carefully the instructions given below to create a virtual environment which will use the last version of Python.

Virtual Environments

To avoid conflicts between packages and Python versions, and also avoid “breaking” your system-wide installation, it is *highly* recommended to create virtual environments. Those are used to isolate a Python configuration relative to a project you are working on.

To install the *virtualenv* tool, open a terminal and type:

```
python -m pip install --user virtualenv
```

If you are not sure about which “user” to put as input, you can omit this flag:

```
python -m pip install virtualenv
```

If you have different versions of Python present on your machine, you can specify the version of Python to install *virtualenv* for:

```
python3.8 -m pip install virtualenv
```

For more information about *virtualenv*, please refer to:

https://virtualenv.pypa.io/en/stable/installation.html

Create a virtual environment

Create a folder for your project. Open a terminal and go to the created folder. To exemplify the process, here follows the command-line used to go in a folder on the Desktop on Linux (just change the *path* on Windows or MacOS):

```
cd ~/Desktop/my_project
```

Once in your folder, you can create your virtual environment using the following command:

```
virtualenv myenv
```

“myenv” can be changed to any environment name of your choice. Once this process is realized, a new folder should appear in your project folder. This folder as the name “myenv”. You can now activate your virtual environment:

```
source myenv/bin/activate
```

You are now in an isolated Python environment! To go back system-wide, you can `deactivate` your environment.

Install needed packages

Once you are in your virtual environment “myenv” (the name should appear on the left side of your terminal line-input), you can install packages in it. To do so, we are gonna use the *pip* installation process.

As for now, we need the following packages:

- jupyter
- pandas
- xlrd
- matplotlib
- numpy

This way, you can install all 5 package using the following command:

```
pip install jupyter pandas xlrd matplotlib numpy
```

You can now test the installation:

```
jupyter notebook
```

This should open a Jupyter Notebook within your web browser. You can then create an IPython page and try to import the different packages:

```
import matplotlib.pyplot as plt
import pandas
import numpy
```

If no errors shoes up, you’re done! Ready to code!

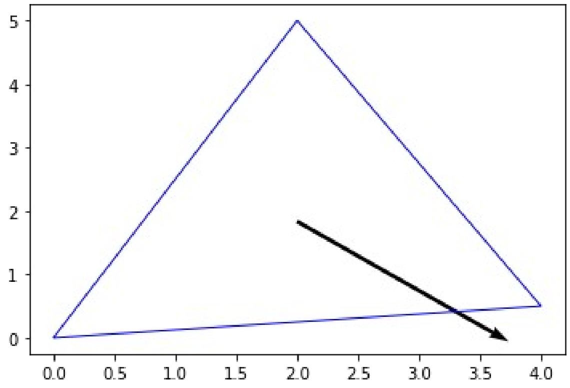
UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE BOGOTÁ

EMC-UN
COLOMBIA

Introduction to Computational Electromagnetics

PART I: The Finite Element Method

Benjamin J. Mailhé, Ph.D


0
1
2
3
4
5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0