INSTALLING PYTHON AND ITS BASIC TOOLS

The Open Distribution System Simulator™ (OpenDSS)

Celso Rocha and Paulo Radatz
OpenDSS Users

Table of Contents

Abstract

This document describes how to install Python and its basic tools in order to run the example *dssvplot.py* available in OpenDSS's "Examples" folder C:\Program Files\OpenDSS\Examples\Python) for those who have already installed OpenDSS.

For those who haven't done it yet, it can be found here.

The particular interest on Python is because, like OpenDSS, it is developed under an open source license, making it freely usable and distributable, even for commercial use (https://www.python.org/about/) and, as described in **OpenDSSPrimer.pdf**, it can be used through Microsoft's COM Interface to drive the OpenDSS engine and to build complex algorithms.

Installation

OpenDSS download and installation

First, you must download and install OpenDSS. You can download it here.

More information about how to download and install the software are available at:

- OpenDSS Training Workshop 2014 (From Slide 11 to Slide 25)
- OpenDSS Manual (Page 20)
- Portuguese video. Although this video is in Portuguese, it may be helpful.

Python download and installation-First Approach

The user should download Python 2 since this is the syntax that was used for developing the *dssvplot.py* example. Python 2.7.11 can be downloaded here.

If you have a Windows machine, you should download either *Windows x86-64 MSI installer* (Python 64 bits) or *Windows x86 MSI installer* (Python 32 bits) depending on your operating system.

RUNNING THE EXAMPLE *DSSVPLOT.PY*

1. Open the CMD window and type the address where the example *dssvplot.py* is located. If you didn't change the default settings during the OpenDSS installation, it should be at *C:\Program Files\OpenDSS\Examples\Python*. The Figure 1 shows the CMD opened at the correct address.
    a. To open the CMD window, type CMD at your Windows Explorer or in the Search box on the Start button.

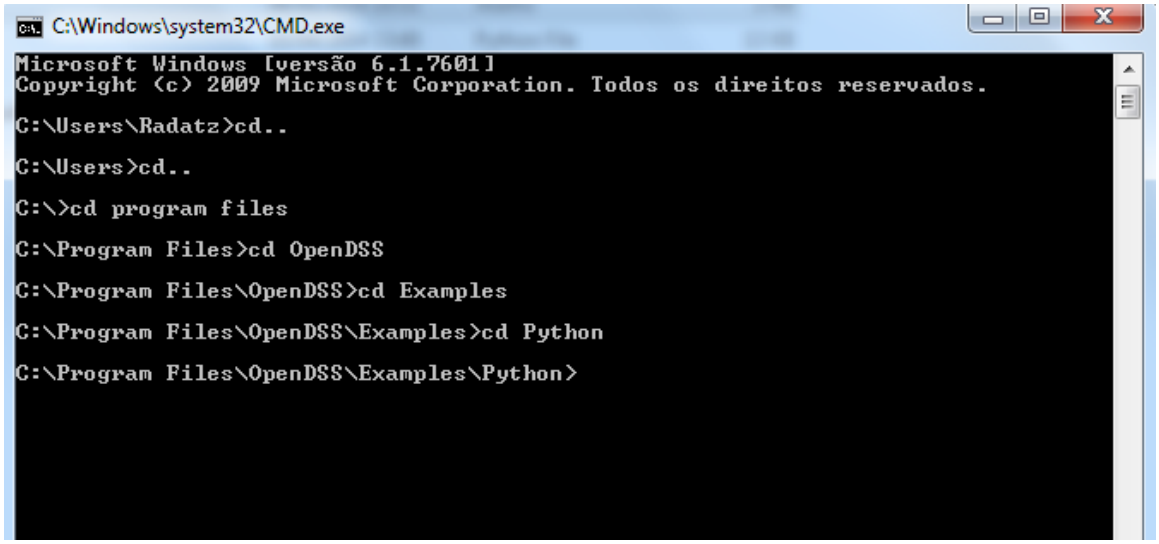

Figure 1

2. Now, you can try typing *dssvplot.py* and see what happens. It should be the same as the Figure 2. As can be seen, we do not have the module named *win32com.client*. This happens due the fact that at the beginning of the code, Figure 3, the *dssvplot.py* attempts to import some modules that will be used further in the code and the first one is the win32com module.
   The use of each module is explained in the document "**Example how to control OpenDSS using Python**".

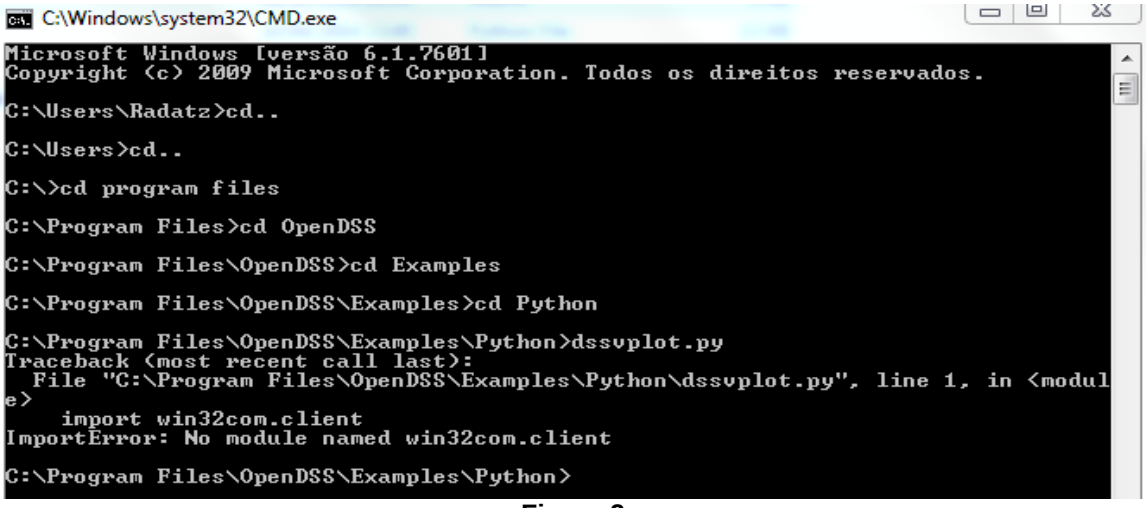

Figure 2

```
import win32com.client
from numpy import *
from pylab import *
import matplotlib.pyplot as plt
from matplotlib.collections import LineCollection
from matplotlib.colors import ColorConverter
import matplotlib.text as text
colorConverter = ColorConverter()
import re
```

Figure 3

DOWNLOAD *WIN32COM.CLIENT*

1. The *win32com.client* can be downloaded here.
   Notice that *pywin32-220.win32-py2.7.exe* refers to a 32bit version and *pywin32-220.win-amd64-py2.7.exe* to a 64bit version. You should download the one corresponding to your Python version.

2. Install it by executing the exe file.
3. Now you can try again typing at CMD *dssvplot.py*. It should be the same as the Figure 4. As you can see, we do not have the module named numpy.

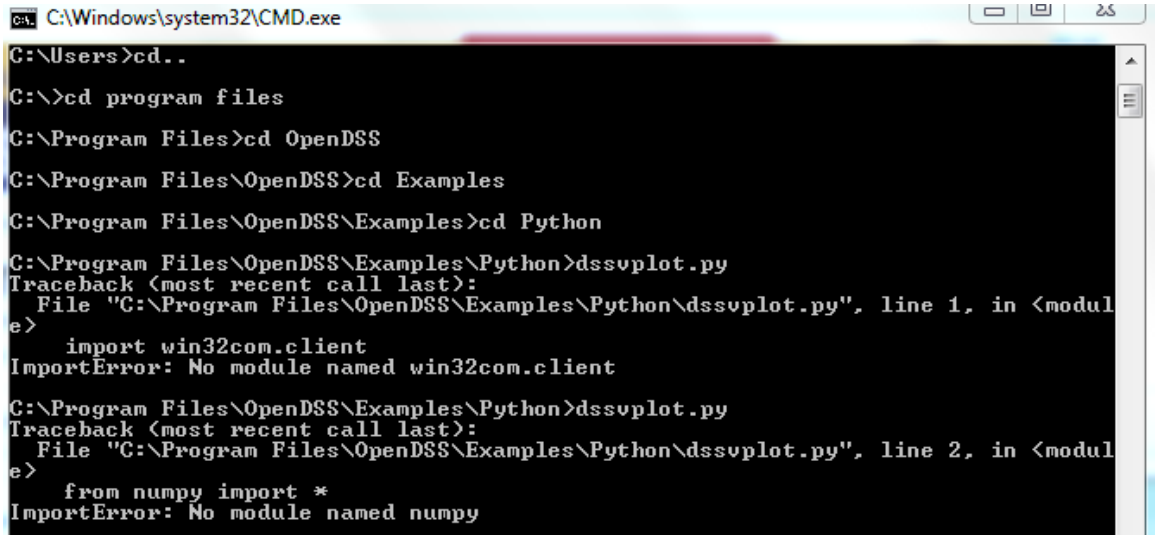

Figure 4

DOWNLOAD AND INSTALL NUMPY

1. Download numpy, which can be found here.
   Notice that *numpy-1.10.2+mkl-cp27-none-win32.whl* refers to a 32bit version and *numpy-1.10.2+mkl-cp27-none-win_amd64.whl* to a 64bit version. You should download the one corresponding to your Python version.

2. Open a CMD window, go to the address where the *.whl* file is saved (in this example, the folder Downloads was used) and type *C:/Python27/Scripts/pip install "numpy-1.10.2+mkl-cp27-none-win32.whl"*, as can be seen in Figure 5.

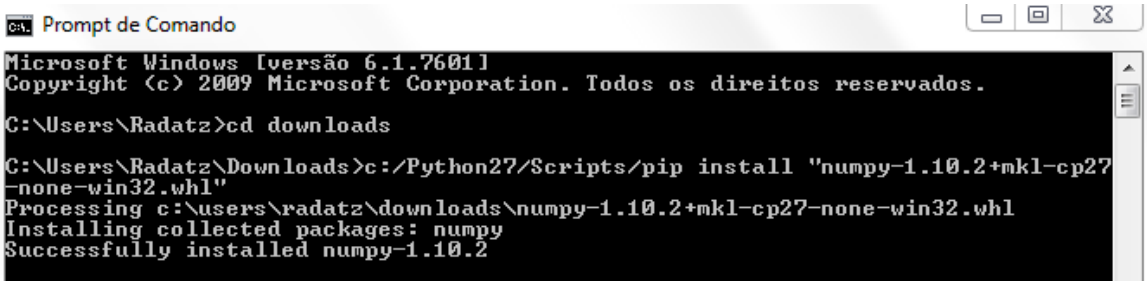

Figure 5

3. By running the example dssvplot.py again, you should get the same error as in Figure 6. Now, as you can see, the missing module is the pylab.

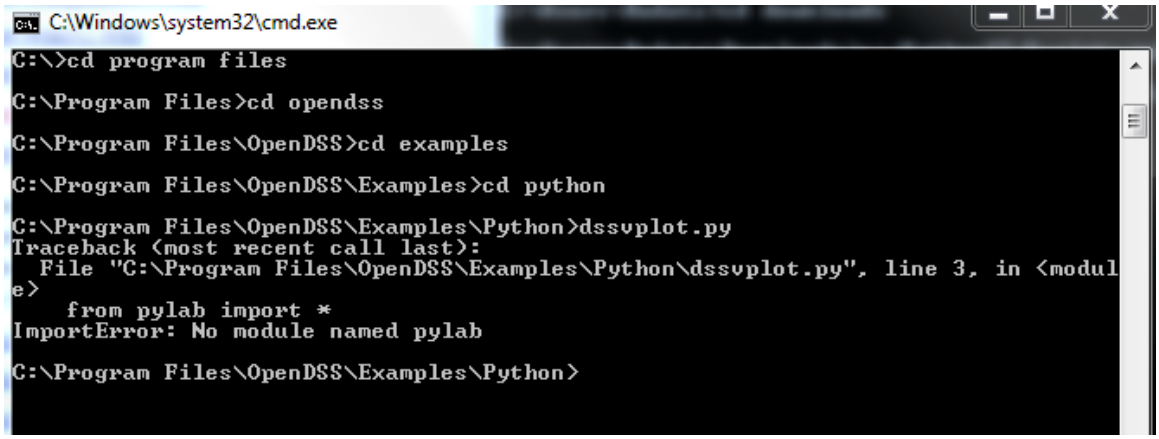

Figure 6

DOWNLOAD AND INSTALL PYLAB

1. Open the address *C:/python27/Scripts* and, in the CMD window, type *"pip install pylab",* you can see it in the Figure 7.

Figure 7

2. As a result, the missing module is the scipy, as shown in Figure 8. One way to solve this problem is by downloading and installing the scipy module.

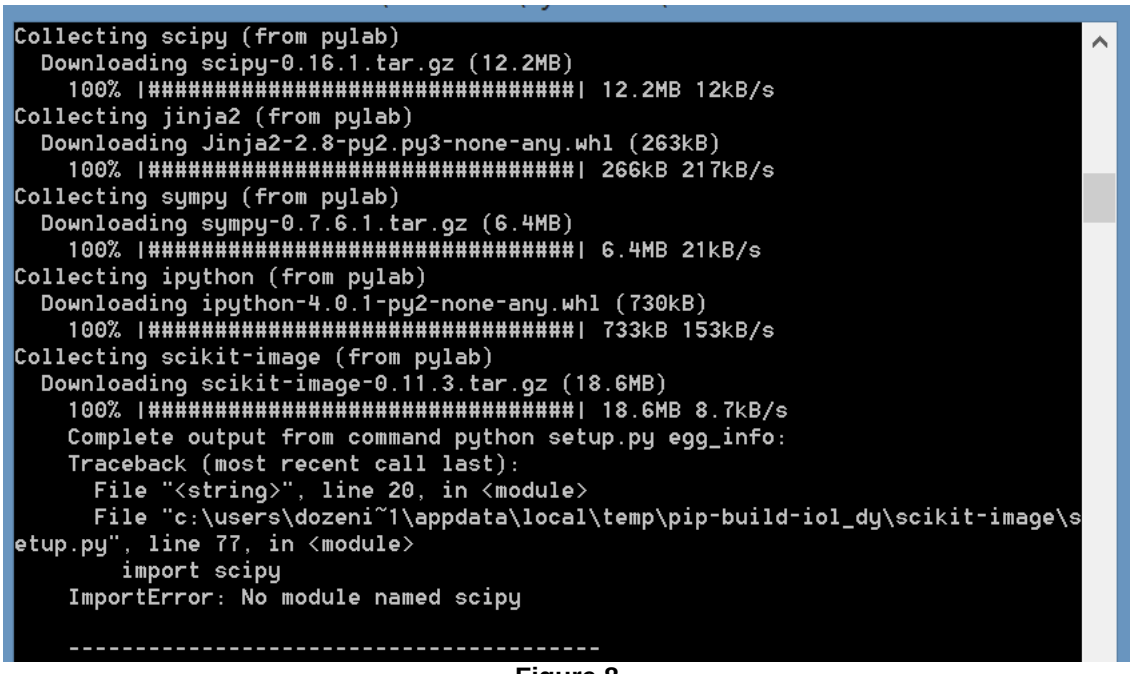

Figure 8

3. Download scipy using the following link.
   Again, notice that *scipy-0.16.1-cp27-none-win32.whl* refers to a 32bit version and *numpy-1.10.2+mkl-cp27-none-win_amd64.whl* to a 64bit version. You should download the one corresponding to your Python version.

4. Open a CMD window, go to the folder where you have saved the *.whl* file and type *C:/Python27/Scripts/pip install "scipy-0.16.1-cp27-none-win32.whl"*. The result should be similar to Figure 9.
5. Again open the address *C:/python27/Scripts* and, in the CMD window, type *"pip install pylab"*.

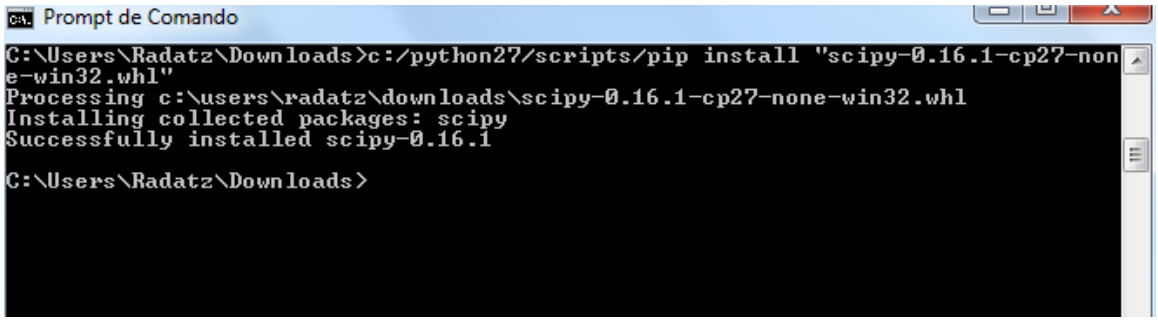

Figure 9

INSTALL MATPLOTLIB

1. Open the address *C:/python27/Scripts* at the CMD window and type *"pip install matplotlib"*.
2. Try running the example *dssvplot.py* again. You should get the error shown in the Figure 10.

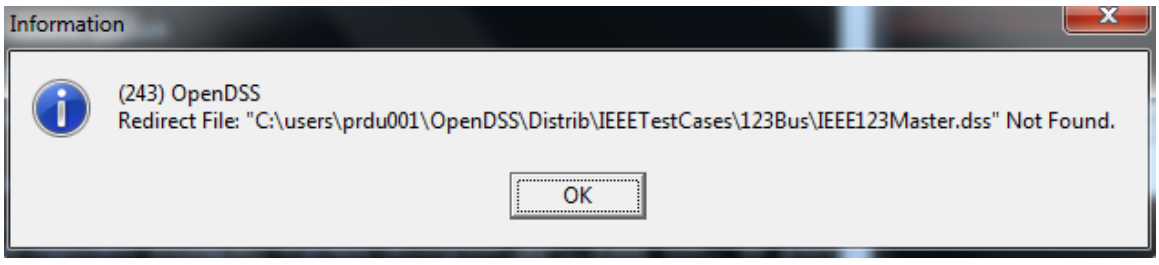

Figure 10

CHANGE THE PATH NAME IN DSSVPLOT.PY

1. The file *dssvplot.py* is probably in the following folder: *C:\program files\opendss\examples\python*. Go to the line 346 of the dssvplot.py code and change the path C:\users\prdu001\OpenDSS\Distrib\IEEETestCases\123Bus\IEEE123Master.dss for the path where your file IEEE123Master.dss is located.

RUN *DSSVPLOT.PY*

1. Open the CMD window and run *dssvplot.py*, Figure 11. You should also get a circuit map and a voltage profile plot, as you can see in Figures 12 and 13.

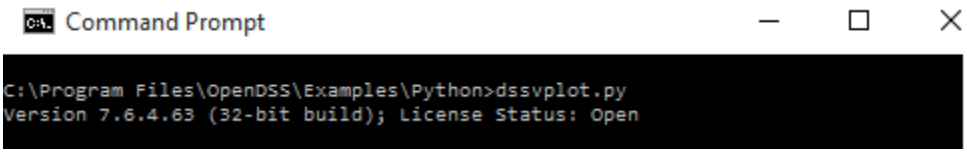

Figure 11

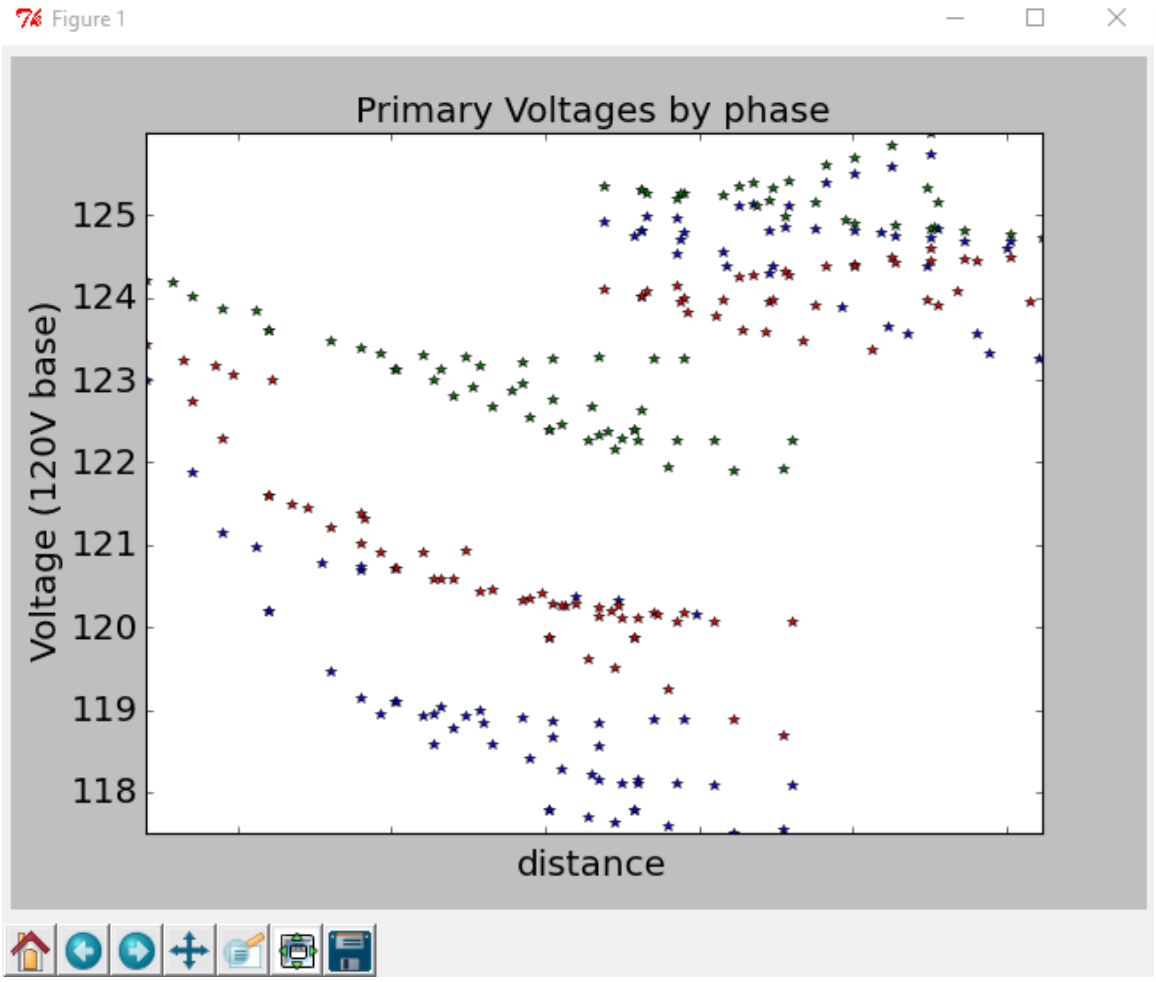

Figure 12

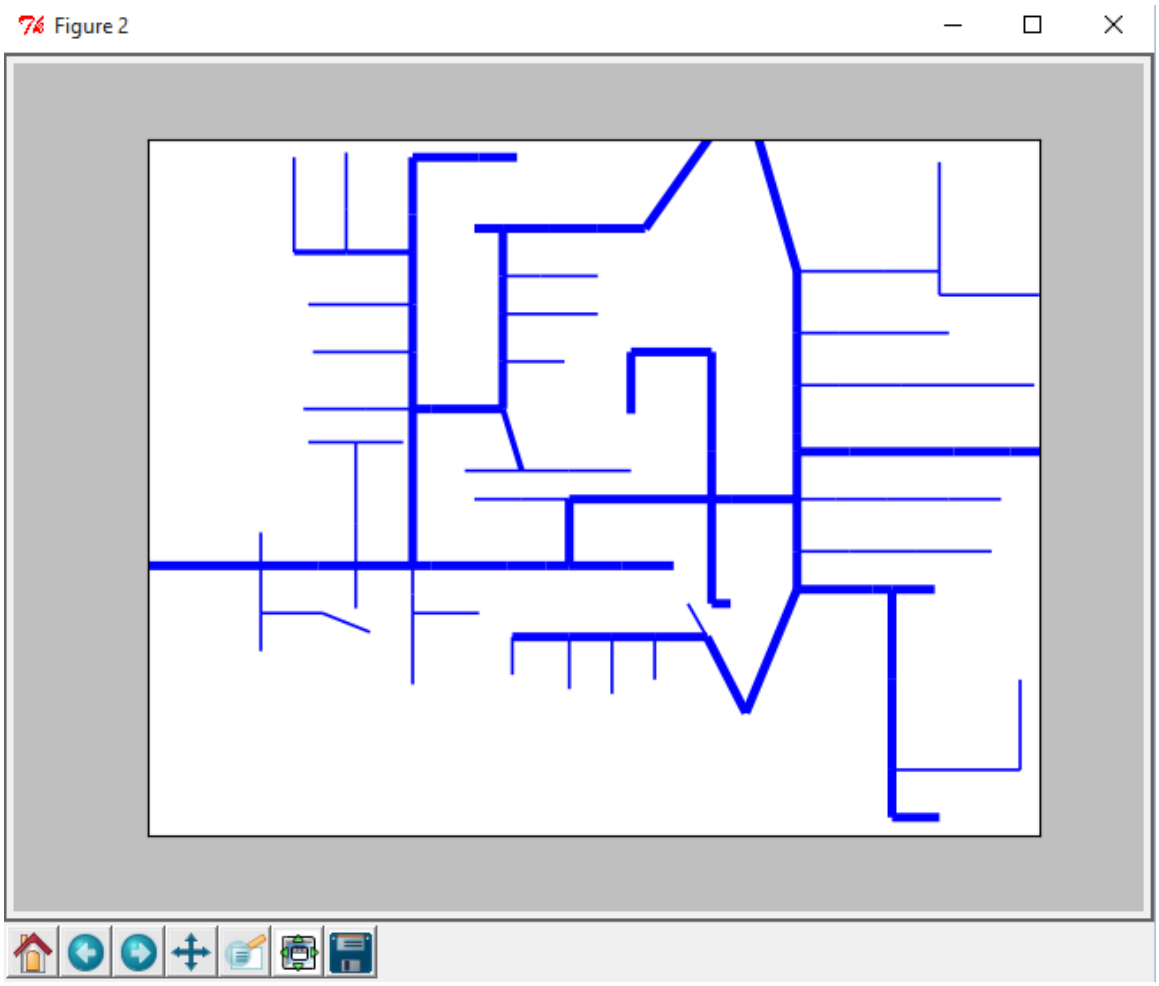

Figure 13

Python download and installation-Alternative Approach

You can also use the following approach, which is easier than the first one because you will be dealing only with .exe files.

RUNNING THE EXAMPLE *DSSVPLOT.PY*

1. Open the CMD window and type the address where the example *dssvplot.py* is located. If you didn't change the default settings during the OpenDSS installation, it should be at *C:\Program Files\OpenDSS\Examples\Python*. The first line of Figure 14 shows the CMD opened at the correct address.
    a. To open the CMD window, type CMD at your Windows Explorer or in the Search box on the Start button.

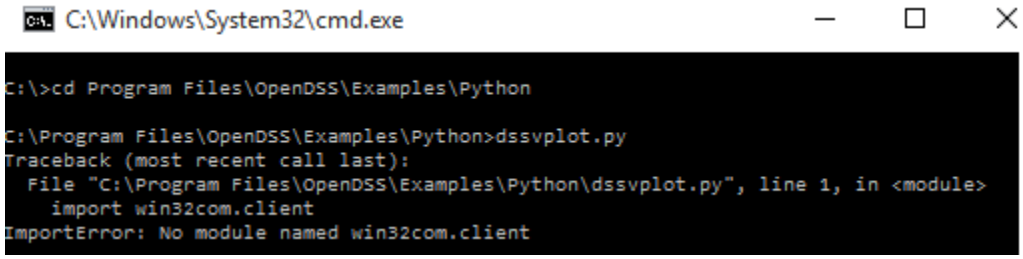

Figure 14

2. Now, you can try typing *dssvplot.py* and see what happens. It should be the same as the second line of Figure 14. As can be seen, we do not have the module named *win32com.client*. This happens due the fact that at the beginning of the code, Figure 3, the *dssvplot.py* attempts to import some modules that will be used further in the code and the first one is the win32com module.
   The use of each module is explained in the document "**Example how to control OpenDSS using Python**".

DOWNLOAD *WIN32COM.CLIENT*

1. The *win32com.client* can be downloaded here.
2. Install it by executing the exe file. You can always check if the module was properly installed by typing *pip list* once you are running the CMD in the *C:\Python27\Scripts*, as can be seen in Figure 15.

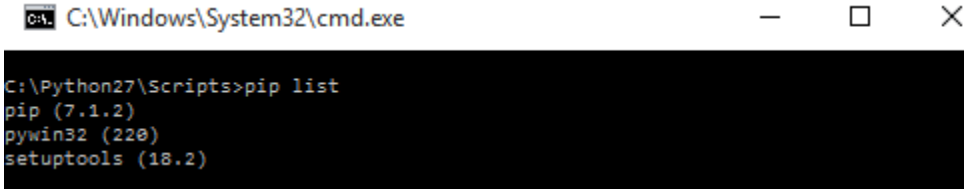

Figure 15

3. Now you can try again typing at CMD *dssvplot.py*. It should be the same as the Figure 16. As you can see, we do not have the module named numpy.

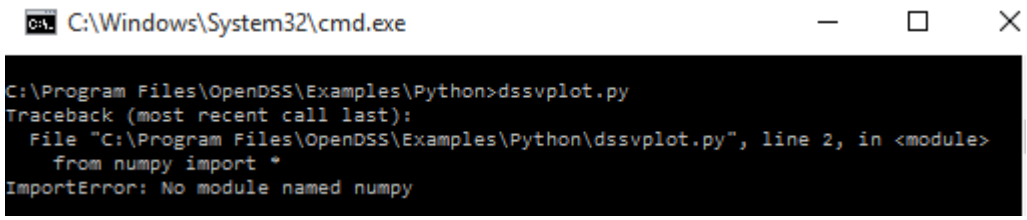

Figure 16

Download and Install Matplotlib

1. Differently from the first approach, even though you are receiving an error showing that you do not have the module numpy, first download and install the module matplotlib by executing the exe file. It can be found here. This step will avoid the installation of the modules pylab and scipy.
2. If you try typing at CMD *dssvplot.py*, you will receive the same error message.

Download and Install Numpy

1. Download numpy, which can be found here.
2. Install it by executing the exe file and run the *dssvplot.py* again. You should get the same message as the figure 17. Finally, you need to change the path name in dssvplot.py, as described in the first approach.

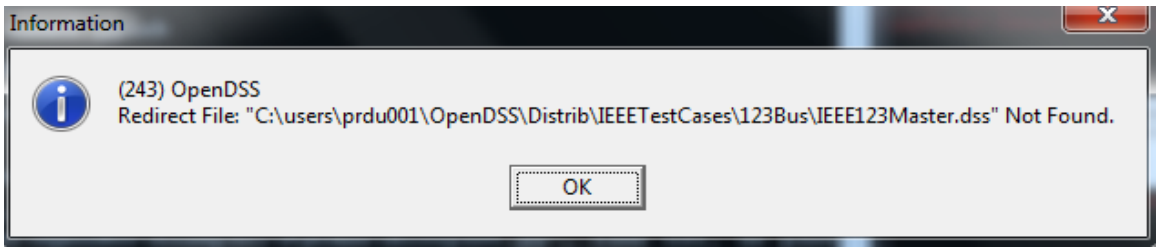

Figure 17

Where can we write our code?

We recommend the use of PyCharm which has a free version called Pycharm Community Edition. It is helpful during debugging and gives us a friendly window to code.

You can download it here.

The Figure 12 shows the code of dssvplot.py in PyCharm.

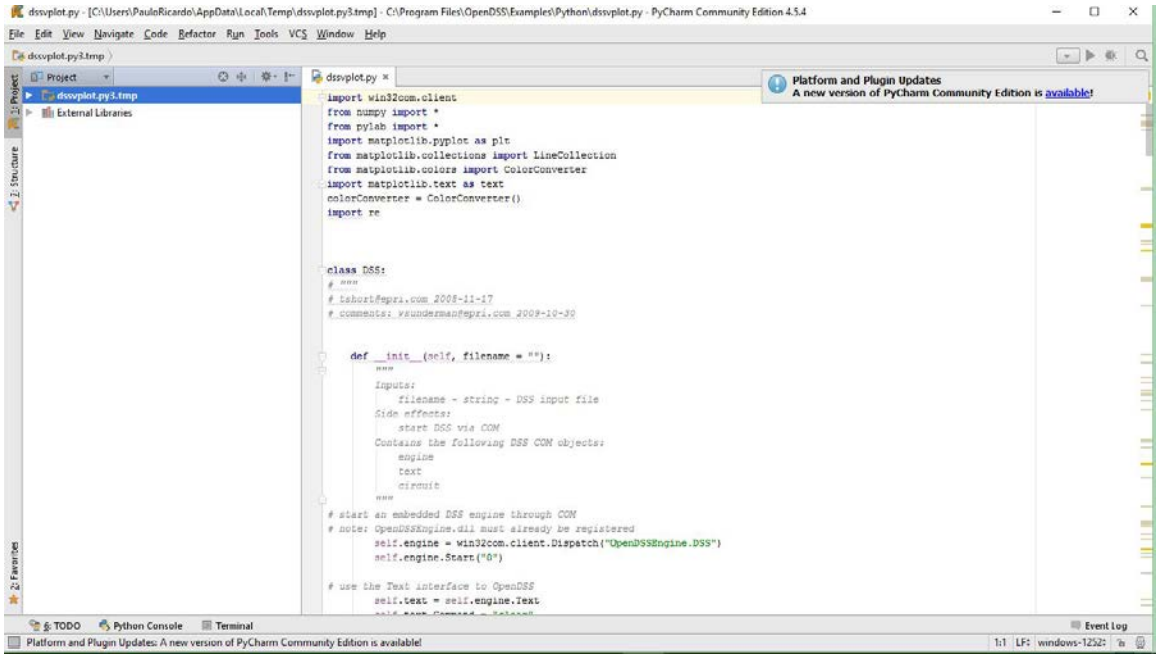

Figure 18