



INSTALLING PYTHON AND ITS BASIC TOOLS

The Open Distribution System Simulator™ (OpenDSS)

Celso Rocha and Paulo Radatz
OpenDSS Users

Table of Contents

- TABLE OF CONTENTS 3**
- ABSTRACT 4**
- INSTALLATION 5**
 - OPENDSS DOWNLOAD AND INSTALLATION 5
- PYTHON DOWNLOAD AND INSTALLATION-FIRST APPROACH 6**
 - RUNNING THE EXAMPLE *DSSVPLOT.PY* 6
 - DOWNLOAD *WIN32COM.CLIENT* 7
 - DOWNLOAD AND INSTALL NUMPY 8
 - DOWNLOAD AND INSTALL PYLAB 9
 - INSTALL MATPLOTLIB 10
 - CHANGE THE PATH NAME IN *DSSVPLOT.PY* 10
 - RUN *DSSVPLOT.PY* 11
- PYTHON DOWNLOAD AND INSTALLATION-ALTERNATIVE APPROACH 13**
 - RUNNING THE EXAMPLE *DSSVPLOT.PY* 13
 - DOWNLOAD *WIN32COM.CLIENT* 13
 - DOWNLOAD AND INSTALL MATPLOTLIB 14
 - DOWNLOAD AND INSTALL NUMPY 14
- WHERE CAN WE WRITE OUR CODE? 15**

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.](#)

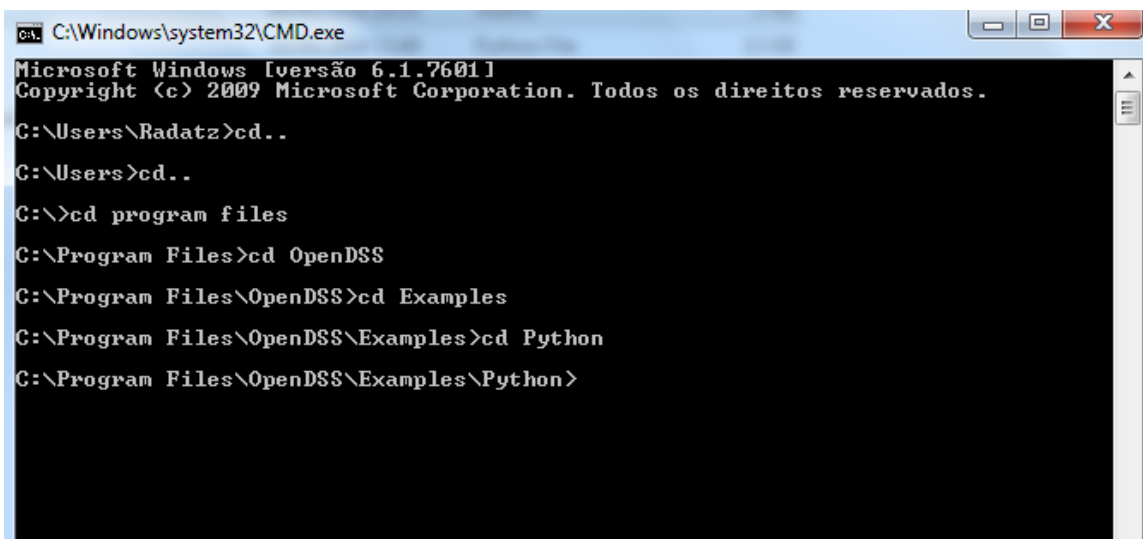
Python download and installation-First Approach

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 *DSSVPLLOT.PY*

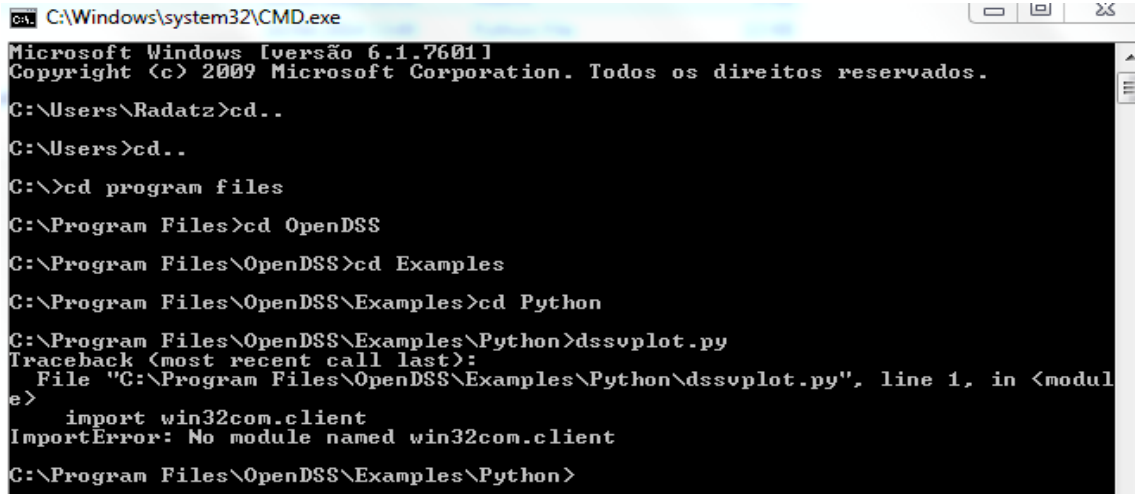
1. Open the CMD window and type the address where the example *dssvplot.py* is located. If you didn't change the default settings during the OpenDSS installation, it should be at *C:\Program Files\OpenDSS\Examples\Python*. The Figure 1 shows the CMD opened at the correct address.
 - a. To open the CMD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



```
cmd C:\Windows\system32\CMD.exe
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.
C:\Users\Radatz>cd..
C:\Users>cd..
C:\>cd program files
C:\Program Files>cd OpenDSS
C:\Program Files\OpenDSS>cd Examples
C:\Program Files\OpenDSS\Examples>cd Python
C:\Program Files\OpenDSS\Examples\Python>
```

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**”.



```
C:\Windows\system32\CMD.exe
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Radatz>cd..
C:\Users>cd..
C:\>cd program files
C:\Program Files>cd OpenDSS
C:\Program Files\OpenDSS>cd Examples
C:\Program Files\OpenDSS\Examples>cd Python
C:\Program Files\OpenDSS\Examples\Python>dssvplot.py
Traceback (most recent call last):
  File "C:\Program Files\OpenDSS\Examples\Python\dssvplot.py", line 1, in <module>
    import win32com.client
ImportError: No module named win32com.client

C:\Program Files\OpenDSS\Examples\Python>
```

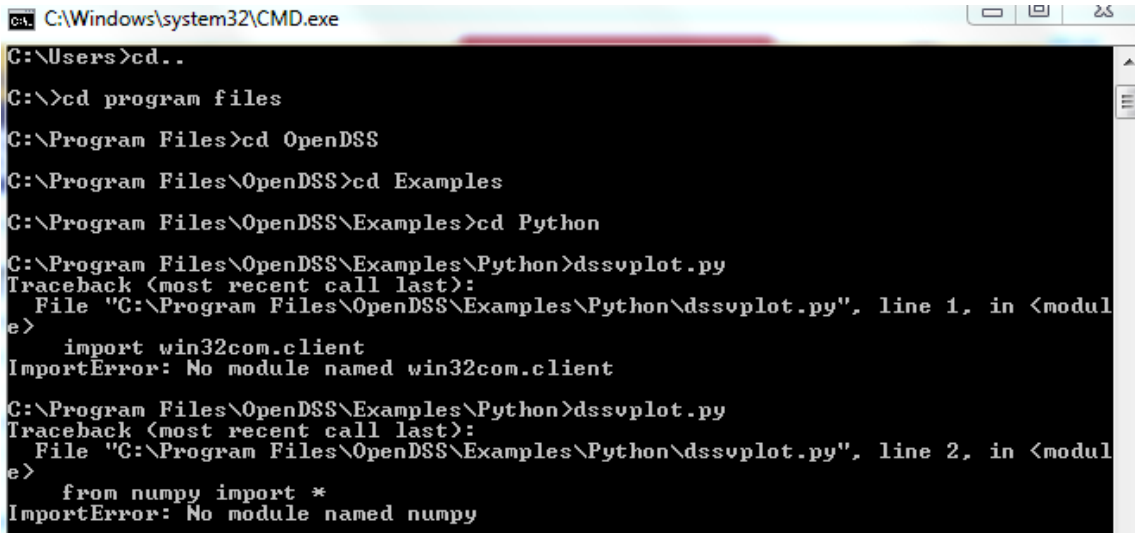
Figure 2

```
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.

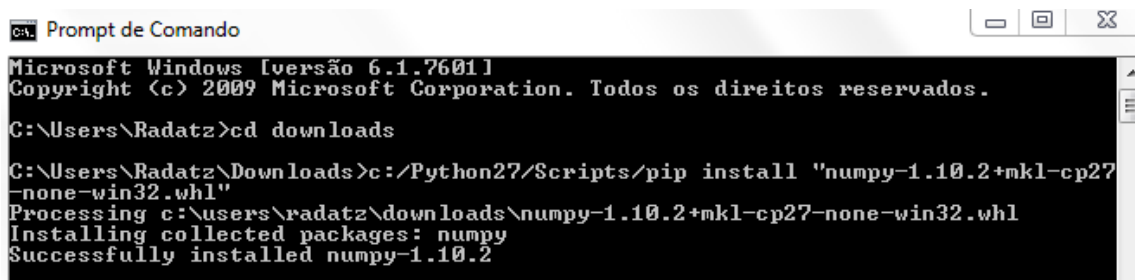


```
C:\Windows\system32\CMD.exe
G:\Users>cd..
C:\>cd program files
C:\Program Files>cd OpenDSS
C:\Program Files\OpenDSS>cd Examples
C:\Program Files\OpenDSS\Examples>cd Python
C:\Program Files\OpenDSS\Examples\Python>dssvplot.py
Traceback (most recent call last):
  File "C:\Program Files\OpenDSS\Examples\Python\dssvplot.py", line 1, in <module>
    import win32com.client
ImportError: No module named win32com.client
C:\Program Files\OpenDSS\Examples\Python>dssvplot.py
Traceback (most recent call last):
  File "C:\Program Files\OpenDSS\Examples\Python\dssvplot.py", line 2, in <module>
    from numpy import *
ImportError: No 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.



```
Prompt de Comando
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.
C:\Users\Radatz>cd downloads
C:\Users\Radatz\Downloads>c:/Python27/Scripts/pip install "numpy-1.10.2+mkl-cp27-none-win32.whl"
Processing c:\users\radatz\downloads\numpy-1.10.2+mkl-cp27-none-win32.whl
Installing collected packages: numpy
Successfully installed numpy-1.10.2
```

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*.


```
C:\Windows\system32\cmd.exe
C:\>cd program files
C:\Program Files>cd opendss
C:\Program Files\OpenDSS>cd examples
C:\Program Files\OpenDSS\Examples>cd python
C:\Program Files\OpenDSS\Examples\Python>dssvplot.py
Traceback (most recent call last):
  File "C:\Program Files\OpenDSS\Examples\Python\dssvplot.py", line 3, in <module>
    from pylab import *
ImportError: No module named pylab
C:\Program Files\OpenDSS\Examples\Python>
```

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.

```
C:\Python27\Scripts>pip install pylab
```

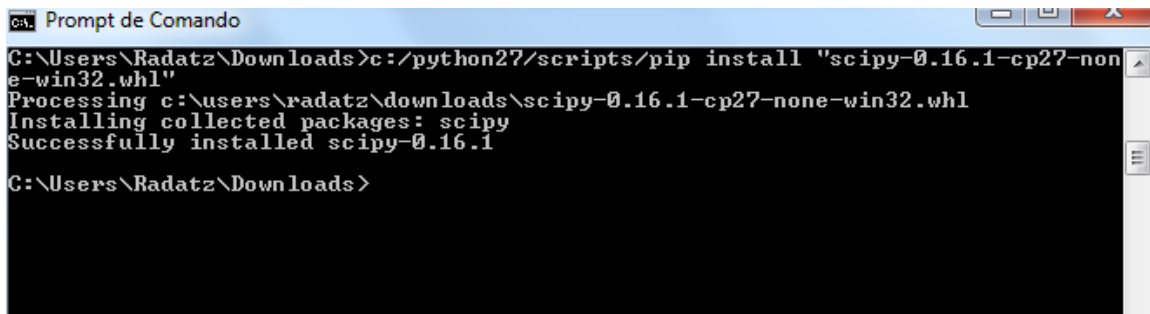
Figure 7

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.

```
Collecting scipy (from pylab)
  Downloading scipy-0.16.1.tar.gz (12.2MB)
    100% |#####| 12.2MB 12kB/s
Collecting jinja2 (from pylab)
  Downloading Jinja2-2.8-py2.py3-none-any.whl (263kB)
    100% |#####| 266kB 217kB/s
Collecting sympy (from pylab)
  Downloading sympy-0.7.6.1.tar.gz (6.4MB)
    100% |#####| 6.4MB 21kB/s
Collecting ipython (from pylab)
  Downloading ipython-4.0.1-py2-none-any.whl (730kB)
    100% |#####| 733kB 153kB/s
Collecting scikit-image (from pylab)
  Downloading scikit-image-0.11.3.tar.gz (18.6MB)
    100% |#####| 18.6MB 8.7kB/s
Complete output from command python setup.py egg_info:
Traceback (most recent call last):
  File "<string>", line 20, in <module>
  File "c:\users\dozeni~1\appdata\local\temp\pip-build-iol_dy\scikit-image\setup.py", line 77, in <module>
    import scipy
ImportError: No module named scipy
-----
```

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"*.



```
ca. Prompt de Comando
C:\Users\Radatz\Downloads>c:/python27/scripts/pip install "scipy-0.16.1-cp27-none-win32.whl"
Processing c:\users\radatz\downloads\scipy-0.16.1-cp27-none-win32.whl
Installing collected packages: scipy
Successfully installed scipy-0.16.1
C:\Users\Radatz\Downloads>
```

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.

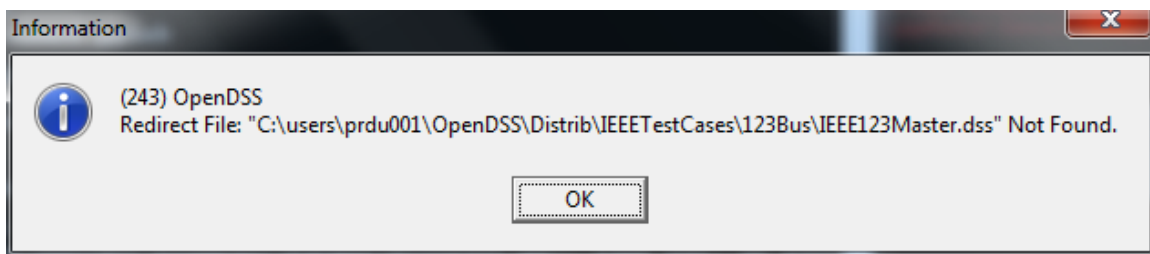


Figure 10

CHANGE THE PATH NAME IN DSSVPLLOT.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 *DSSVPLLOT.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.

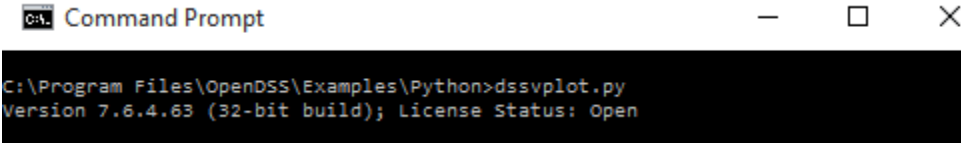


Figure 11

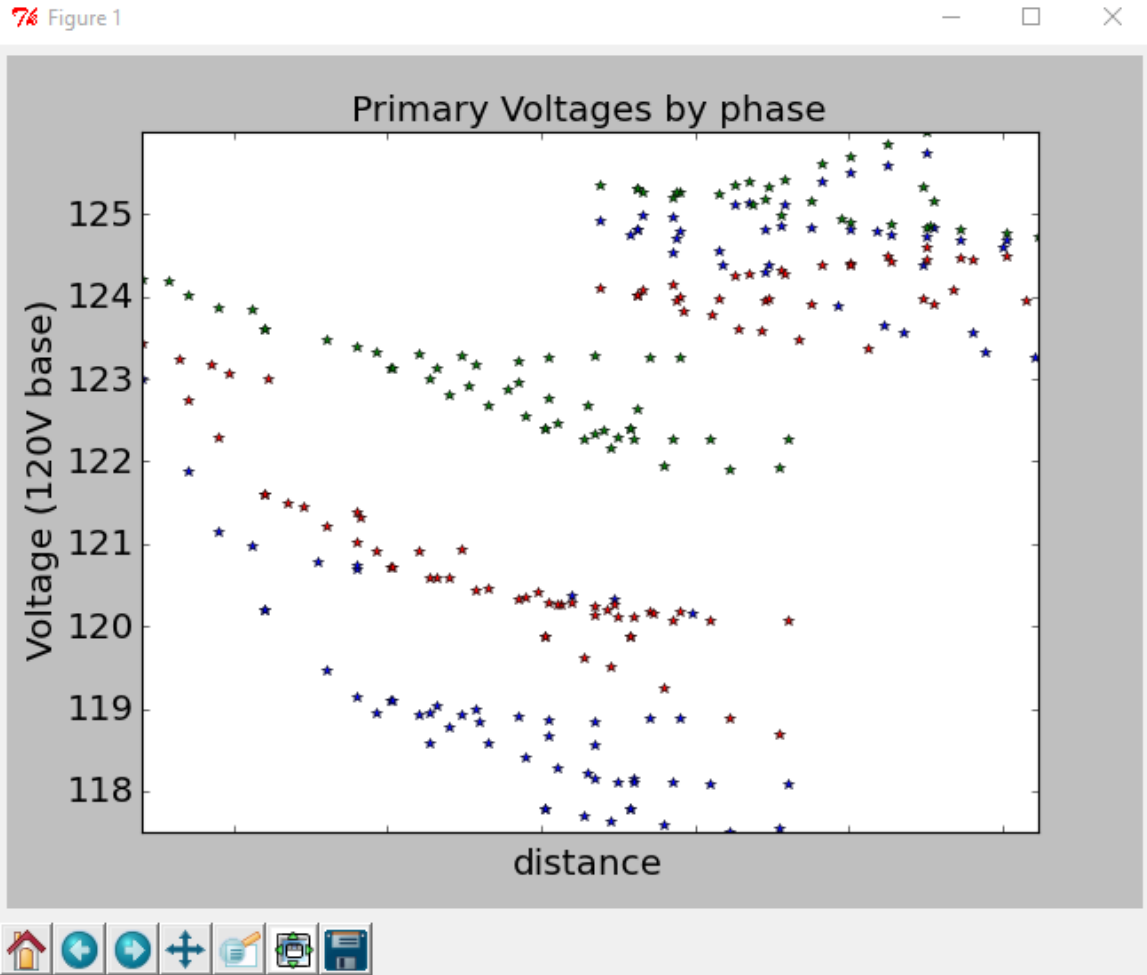


Figure 12

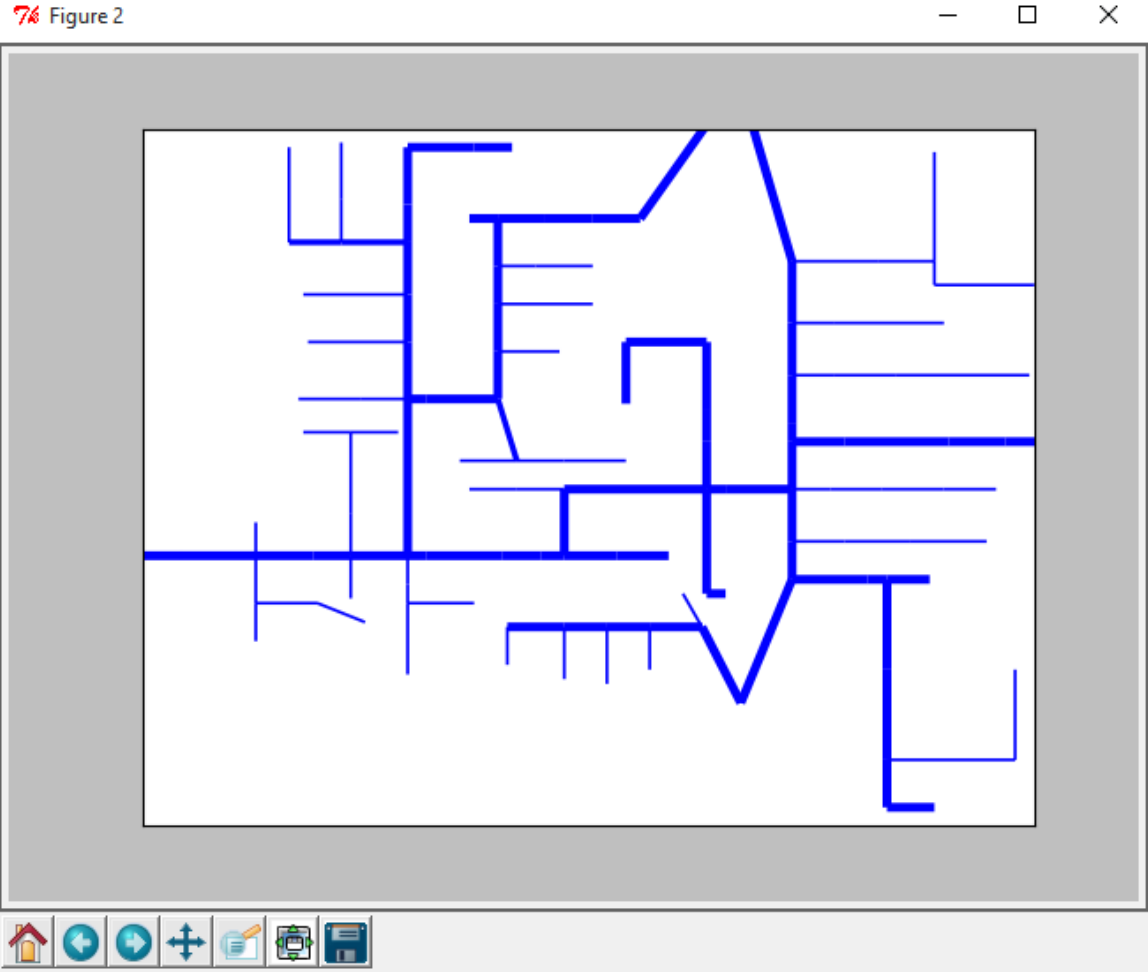


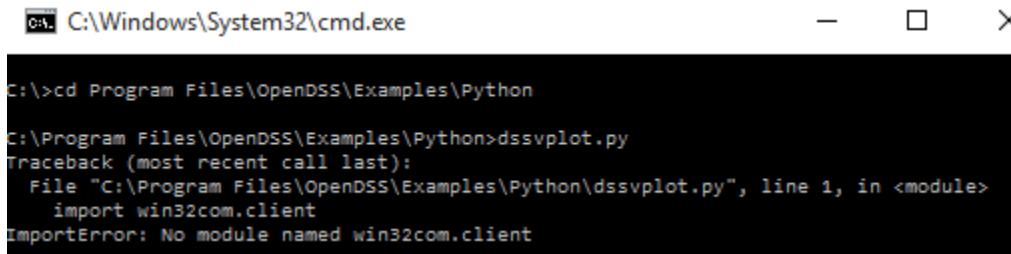
Figure 13

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 *DSSVPLLOT.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.



```
C:\Windows\System32\cmd.exe

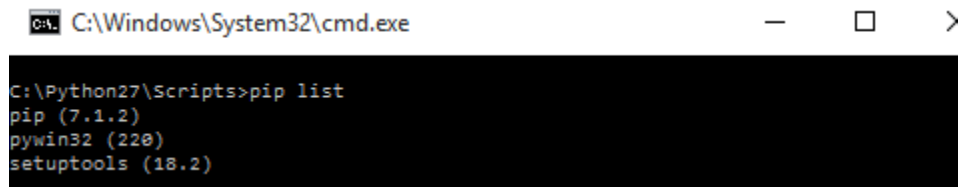
C:\>cd Program Files\OpenDSS\Examples\Python
C:\Program Files\OpenDSS\Examples\Python>dssvplot.py
Traceback (most recent call last):
  File "C:\Program Files\OpenDSS\Examples\Python\dssvplot.py", line 1, in <module>
    import win32com.client
ImportError: No module named win32com.client
```

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.

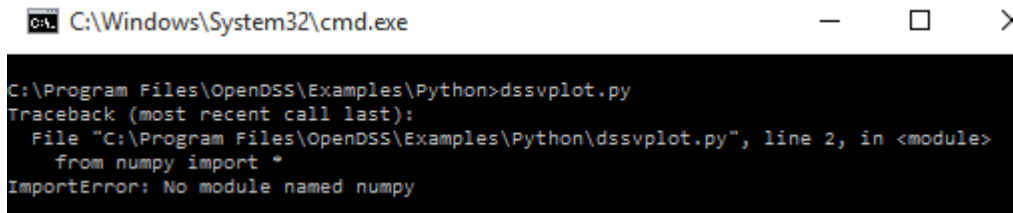


```
C:\Windows\System32\cmd.exe

C:\Python27\Scripts>pip list
pip (7.1.2)
pywin32 (228)
setuptools (18.2)
```

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.



```
C:\Windows\System32\cmd.exe

C:\Program Files\OpenDSS\Examples\Python>dssvplot.py
Traceback (most recent call last):
  File "C:\Program Files\OpenDSS\Examples\Python\dssvplot.py", line 2, in <module>
    from numpy import *
ImportError: No 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.

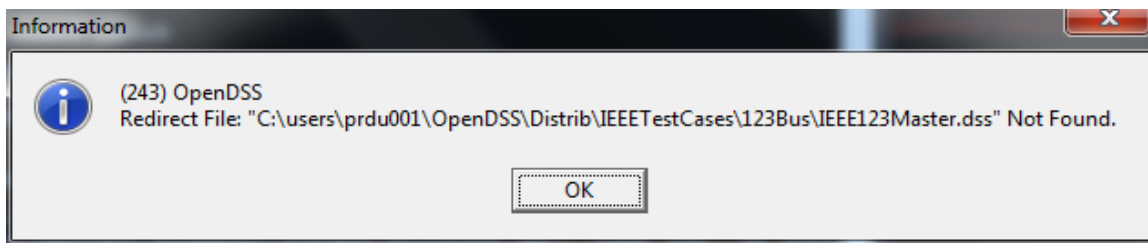


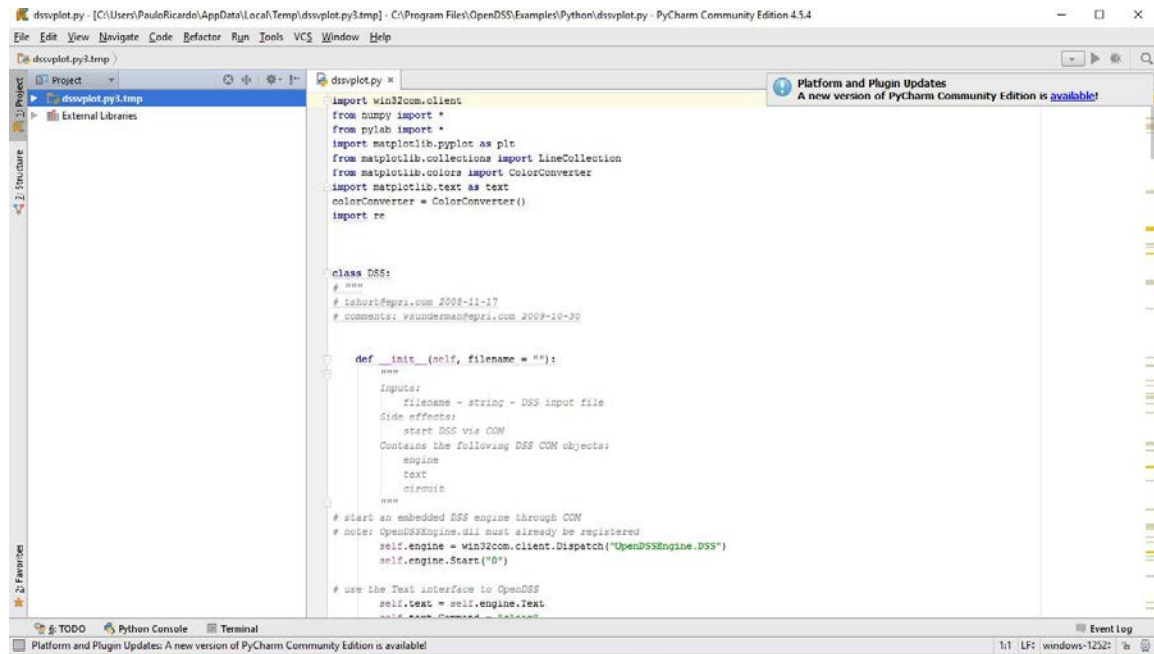
Figure 17

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.



```
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

class DSS:
    # ...
    # Lehart@epri.com 2008-11-17
    # comments: vsunderman@epri.com 2009-10-30

    def __init__(self, filename = ""):
        """
        Inputs:
            filename - string - DSS input file
        Side effects:
            start DSS via COM
            Contains the following DSS COM objects:
                engine
                text
                circuit
        """
        # start an embedded DSS engine through COM
        # note: OpenDSSEngine.dll must already be registered
        self.engine = win32com.client.Dispatch("OpenDSSEngine.DSS")
        self.engine.Start("0")

        # use the Text interface to OpenDSS
        self.text = self.engine.Text
        self.com.Command = self.com
```

Figure 18