

Προγραμματισμός σε C++ & Python & Εφαρμογές στη Ναυπηγική & Ναυτική Μηχανολογία

ΣΝΜΜ 2019

Μάθημα 6Α: Βιβλιοθήκες NymPy, SciPy. Γραφικά

Γεώργιος Παπαλάμπρου

Επίκουρος Καθηγητής ΕΜΠ

`george.papalambrou@lme.ntua.gr`

Εργαστήριο Ναυτικής Μηχανολογίας (Κτίριο Λ)

Σχολή Ναυπηγών Μηχανολόγων Μηχανικών

Εθνικό Μετσοβίο Πολυτεχνείο

April 3, 2019

Περιεχόμενα

1 Classes

Περιεχόμενο Μαθήματος

- Εβδομάδα 1. A. Εισαγωγή. Η γλώσσα. Το περιβάλλον Linux. Command line. Python interpreter. Ιστοσελίδα μαθήματος. Βιβλιογραφία. Editors: Sublime, Spyder. B. Εισαγωγή στην γλώσσα Python. Hello World.
- Εβδομάδα 2. A. Data types. Loops. Control. B. Παραδείγματα
- Εβδομάδα 3. Functions. Modules
- Εβδομάδα 4. OOP. Classes
- Εβδομάδα 5. A. Παραδείγματα: Μέτρηση και επεξεργασία δεδομένων. B. Errors-Exceptions.
- Εβδομάδα 6.
A. Βιβλιοθήκες NymPy, SciPy. Γραφικά
B. Εφαρμογή: Neural Networks. Machine Learning. Εφαρμογή: Hardware. Πλατφόρμες. Πρωτόκολλα. Βασικό I/O

Εισαγωγή


Για σήμερα η παράδοση προέρχεται από:



- Programming and Scientific Computing in Python for Aerospace Engineers, AE Tutorial Programming Python, by Jacco Hoekstra, TU Delft


Ch. 9: Matplotlib: Plotting in Python

Ch. 11: Numpy and Scipy : Scientific Computing with Arrays and Matrices

Γραφικά: Matplotlib

<https://matplotlib.org>

...

 Search





Apply to work with matplotlib for GSOC! Take a look at our [ideas list](#) and learn how to [apply here](#) .
The 2019 SciPy John Hunter Excellence in Plotting Contest is accepting submissions until June 8th!

matplotlib

Version 3.0.3

[home](#) | [examples](#) | [tutorials](#) | [API](#) | [docs](#) »

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.



Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the [sample plots](#) and [thumbnail gallery](#).

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Installation

Visit the [Matplotlib Installation Instructions](#).

Documentation

This is the documentation for Matplotlib version 3.0.3.

To get started, read the [User's Guide](#).

Γραφικά: Matplotlib - Tutorials

Δείτε: Introductory -> Usage Guide (web)



[home](#) | [examples](#) | [tutorials](#) | [API](#) | [docs](#) » [User's Guide](#) »

[pre](#)

Tutorials ¶

This page contains more in-depth guides for using Matplotlib. It is broken up into beginner, intermediate, and advanced sections, as well as sections covering specific topics.

For shorter examples, see our [examples page](#). You can also find [external resources](#) and a [FAQ](#) in our [user guide](#).

Introductory

These tutorials cover the basics of creating visualizations with Matplotlib, as well as some best-practices in using the package effectively.



Usage Guide



Pyplot tutorial



Sample plots in
Matplotlib



Image tutorial



The Lifecycle of a
Plot



Customizing
Matplotlib with style
sheets and rcParams

Γραφικά: Matplotlib - ch. 9 - TU Delft (pdf)

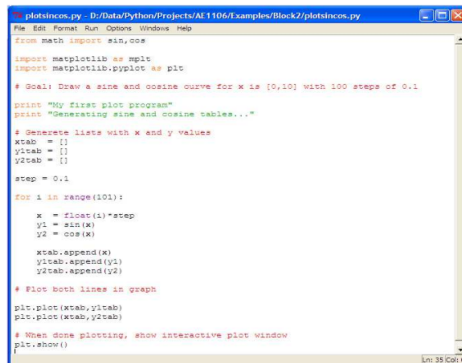
9. Matplotlib: Plotting in Python

9.1 Example: plotting sine and cosine graph

Making graphs in Python is very easy with the `matplotlib` module. A large range of types of plots can be made on screen and saved in several high-quality formats (like .eps) to be used in reports etc.

Try the example program below (or download `plotsincos.py` from blackboard). The header is the standard way to import both `matplotlib` and `matplotlib.pyplot`. It uses `import module as newname`, to create shorter names, which make it easier to use the modules.

Pyplot is a module inside a module and imported as `plt` in the code below. Then it uses a for-loop to generate different values for `x` and with this, two `y`-values using sine and cosine are calculated. Append each value to the list variables `xtab`, `y1tab` and `y2tab` and then plot these in the same figure. This figure will only become visible when a call to `plt.show()` is made. These starts a separate program with the plot and waits until the user has closed this window.



```
plotsincos.py - D:/Data/Python/Projects/AE1106/Examples/Block2/plotsincos.py
File Edit Format Run Options Windows Help
from math import sin,cos
import matplotlib as mplt
import matplotlib.pyplot as plt

# Goal: Draw a sine and cosine curve for x is [0,10] with 100 steps of 0.1
print "My first plot program"
print "Generating sine and cosine tables..."

# Generate lists with x and y values
xtab = []
y1tab = []
y2tab = []

step = 0.1

for i in range(101):
    x = float(i)*step
    y1 = sin(x)
    y2 = cos(x)

    xtab.append(x)
    y1tab.append(y1)
    y2tab.append(y2)

# Plot both lines in graph
plt.plot(xtab,y1tab)
plt.plot(xtab,y2tab)

# When done plotting, show interactive plot window
plt.show()
```

Μαθηματικά: SciPy, Numpy, ...

Εδώ υπάρχουν: NumPy, SciPy, Matplotlib, pandas ...



The screenshot shows the SciPy.org website in a web browser. The address bar displays "https://scipy.org". The header features the SciPy.org logo and the text "Sponsored by ENTHOUGHT". Below the header, there are five circular icons with labels: "Install" (a blue circle with a green arrow pointing down), "Getting Started" (a yellow circle with a white 'S' and a green arrow), "Documentation" (a blue circle with a white 'S' and a stack of books), "Report Bugs" (a blue circle with a white 'S' and a red bug), and "Blogs" (an orange square with a white RSS symbol). Below these icons, a paragraph states: "SciPy (pronounced 'Sigh Pie') is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages:". This is followed by a grid of six packages, each with an icon and a description: NumPy (Base N-dimensional array package), SciPy library (Fundamental library for scientific computing), Matplotlib (Comprehensive 2D Plotting), IPython (Enhanced Interactive Console), Sympy (Symbolic mathematics), and pandas (Data structures & analysis). At the bottom center, there is a blue button labeled "More information...".

SciPy.org

Sponsored by ENTHOUGHT

Install Getting Started Documentation Report Bugs Blogs

SciPy (pronounced "Sigh Pie") is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages:

NumPy
Base N-dimensional array package

SciPy library
Fundamental library for scientific computing

Matplotlib
Comprehensive 2D Plotting

IPython
Enhanced Interactive Console

Sympy
Symbolic mathematics

pandas
Data structures & analysis

More information...

Μαθηματικά: SciPy, Numpy, ...

https://www.scipy.org/install.html

SciPy.org

Powered by
ENTHOUGHT

SciPy.org

Installing packages

These are general instructions for installing packages in the SciPy ecosystem.

Scientific Python distributions

For many users, especially on Windows, the easiest way to begin is to download one of these Python distributions, which include all the key packages:

- **Anaconda:** A free distribution of Python with scientific packages. Supports Linux, Windows and Mac.
- **Enthought Canopy:** The free and commercial versions include the core scientific packages. Supports Linux, Windows and Mac.
- **Python(x,y):** A free distribution including scientific packages, based around the [Spyder IDE](#). Windows and Ubuntu; Py2 only.
- **WinPython:** Another free distribution including scientific packages and the Spyder IDE. Windows only, but more actively maintained and supports the latest Python 3 versions.
- **Pyzo:** A free distribution based on Anaconda and the IEP interactive development environment. Supports Linux, Windows and Mac.

Installing via pip

Most major projects upload official packages to the [Python Package index](#). They can be installed on most operating systems using Python's standard [pip](#) package manager.

Note that you need to have Python and pip already installed on your system.

You can install packages via commands such as:

```
python -m pip install --user numpy scipy matplotlib ipython jupyter pandas sympy nose
```

We recommend using an *user* install, using the `--user` flag to pip (note: do not use `sudo pip`, which can cause problems). This

About SciPy

Getting Started

Documentation

Install

Bug Reports

Codes of Conduct

SciPy Conferences

Topical Software

Citing

Cookbook

Blogs

NumFOCUS

CORE PACKAGES:

[Numpy](#)

[SciPy library](#)

[Matplotlib](#)

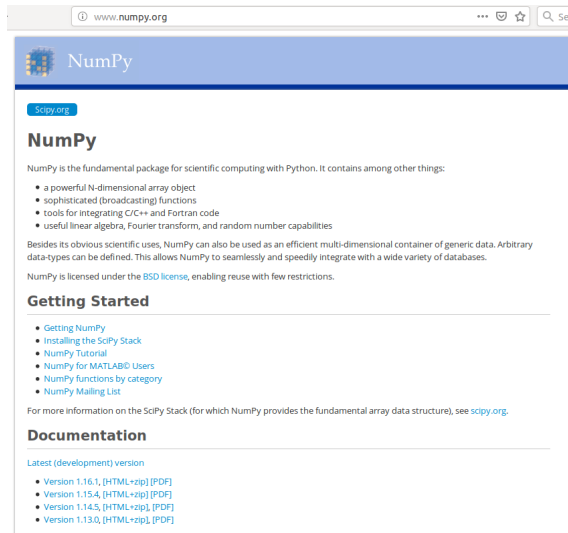
[IPython](#)

[SymPy](#)

[Pandas](#)

Arrays: Numpy, ...

Δείτε: Getting Started > NumPy Tutorial (web)

A screenshot of the NumPy website. The browser address bar shows 'www.numpy.org'. The website has a blue header with the NumPy logo and name. Below the header, there is a link to 'Scipy.org'. The main heading is 'NumPy'. A paragraph describes NumPy as the fundamental package for scientific computing with Python, listing features like N-dimensional array objects, broadcasting functions, and integration with C/C++ and Fortran. It also mentions its use as a multi-dimensional container for generic data. A section titled 'Getting Started' provides links to 'Getting NumPy', 'Installing the SciPy Stack', 'NumPy Tutorial', 'NumPy for MATLAB® Users', 'NumPy functions by category', and 'NumPy Mailing List'. Another section titled 'Documentation' lists links for the latest (development) version and previous versions (1.16.1, 1.15.4, 1.14.5, 1.13.0) in HTML+zip and PDF formats.

www.numpy.org

NumPy

[Scipy.org](#)

NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

NumPy is licensed under the [BSD license](#), enabling reuse with few restrictions.

Getting Started

- [Getting NumPy](#)
- [Installing the SciPy Stack](#)
- [NumPy Tutorial](#)
- [NumPy for MATLAB® Users](#)
- [NumPy functions by category](#)
- [NumPy Mailing List](#)

For more information on the SciPy Stack (for which NumPy provides the fundamental array data structure), see [scipy.org](#).

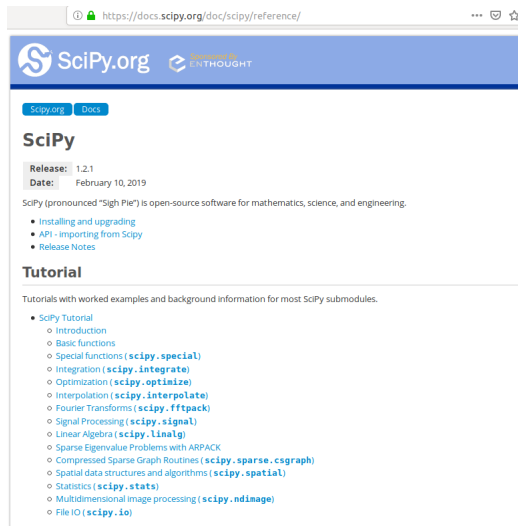
Documentation

[Latest \(development\) version](#)

- [Version 1.16.1, \[HTML+zip\], \[PDF\]](#)
- [Version 1.15.4, \[HTML+zip\], \[PDF\]](#)
- [Version 1.14.5, \[HTML+zip\], \[PDF\]](#)
- [Version 1.13.0, \[HTML+zip\], \[PDF\]](#)

Μαθηματικά: SciPy, ...

Δείτε: Tutorial (web)



The screenshot shows a web browser window with the URL <https://docs.scipy.org/doc/scipy/reference/>. The page header features the SciPy.org logo and the ENTHOUGHT logo. Below the header, there are tabs for "Scipy.org" and "Docs". The main heading is "SciPy". Underneath, it states "Release: 1.2.1" and "Date: February 10, 2019". A paragraph describes SciPy as open-source software for mathematics, science, and engineering. A list of links includes "Installing and upgrading", "API - importing from Scipy", and "Release Notes". A section titled "Tutorial" follows, with a description: "Tutorials with worked examples and background information for most SciPy submodules." Below this, a list of topics is provided, including Introduction, Basic functions, Special functions, Integration, Optimization, Interpolation, Fourier Transforms, Signal Processing, Linear Algebra, Sparse Eigenvalue Problems, Compressed Sparse Graph Routines, Spatial data structures, Statistics, Multidimensional image processing, and File IO.

SciPy.org ENTHOUGHT

Scipy.org Docs

SciPy

Release: 1.2.1
Date: February 10, 2019

SciPy (pronounced "Sigh Pie") is open-source software for mathematics, science, and engineering.

- [Installing and upgrading](#)
- [API - importing from Scipy](#)
- [Release Notes](#)

Tutorial

Tutorials with worked examples and background information for most SciPy submodules.

- SciPy Tutorial
 - Introduction
 - Basic functions
 - Special functions (`scipy.special`)
 - Integration (`scipy.integrate`)
 - Optimization (`scipy.optimize`)
 - Interpolation (`scipy.interpolate`)
 - Fourier Transforms (`scipy.fftpack`)
 - Signal Processing (`scipy.signal`)
 - Linear Algebra (`scipy.linalg`)
 - Sparse Eigenvalue Problems with ARPACK
 - Compressed Sparse Graph Routines (`scipy.sparse.csgraph`)
 - Spatial data structures and algorithms (`scipy.spatial`)
 - Statistics (`scipy.stats`)
 - Multidimensional image processing (`scipy.ndimage`)
 - File IO (`scipy.io`)

Numpy and Scipy : - ch. 11 - TU Delft (pdf)

11. Numpy and Scipy : Scientific Computing with Arrays and Matrices

11.1 Numpy, Scipy

The modules Numpy and Scipy have provided users of Python with an enormous range of engineering and scientific computing tools. Many of this has been inspired by the functionality of MATLAB and its provided toolboxes. The syntax and names of functions are often identical.

Python with Numpy and Scipy are more capable than Matlab. Python is better in handling strings, reading files, working with very large projects and with large datasets. Python can also be used in an object oriented programming way. Both Spyder and the iPy Notebook provide a very user-friendly environment for scientists. A more general difference are the extra possibilities, which are provided by a full-featured general purpose programming language like Python. And, often more importantly, MATLAB is very expensive and many applications require extra toolboxes, which are in turn also very expensive. This also hinders sharing tools as well as quickly using source code from the internet community: often you can only use the downloaded bits after purchasing the required toolboxes.

With Numpy and Scipy, Python has surpassed MATLAB in terms of functionality. The modules are available for free (as are all Python modules). They are developed, maintained, expanded and used by a large academic community, mainly in the US and Europe.

Numpy forms the foundation for Scipy, Matplotlib and many other modules: it provides the array- and matrix-types as well as linear algebra functions as extension to Python. Scipy adds a whole range of sometimes very dedicated scientific computing and engineering functions. Thanks to Numpy and Scipy, Python has become the default language of choice for scientific computing, according to IEEE and many others. Let's explore the capabilities of these modules, even though we can only scratch the surface in this course.

To use these modules, they need to be imported. It has become standard practice to rename them when importing them. Numpy becomes "np" and Scipy becomes "sp". This means we will often see the following header in our scientific applications. In this course, we assume you have imported the modules as follows:

```
import numpy as np
import scipy as sp
import matplotlib as mpl
import matplotlib.pyplot as plt
```

In the Numpy and Scipy documentation it is often even assumed that you have imported everything using `from numpy import *`. So do not forget to type "np." before the Numpy functions and "sp." before the Scipy functions, even though you don't see this in the Numpy and Scipy documentation. Most Numpy functions can also be used as if they were a part of Scipy, with the "sp." prefix.