



PyQB

Monga

Matplotlib

Graphical commands

OO plotting

Programming in Python¹

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia

`mattia.monga@unimi.it`

Academic year 2020/21, II semester



PyQB

Monga

Matplotlib

Graphical commands

OO plotting

Lecture XII: Matplotlib



When you have arrays with many data it is useful to have a way to display them graphically.

- The most popular is `matplotlib`
<https://matplotlib.org/>
- Many other graphical frameworks (e.g., `seaborn`) based on it
- Many, many possibilities to tune your graphics! It's hard to master every detail.
- Be careful: it can be used with two different styles.
 - 1 The (preferred) object-oriented way: clean and rational, but a bit more verbose
 - 2 The procedural way: mostly useful only for “throw-away” scripts, but for this reason more common in the examples you can find online

PyQB

Monga

Matplotlib

Graphical commands

OO plotting

Graphical output is an operating system service



PyQt

Monga

Matplotlib

Graphical commands

OO plotting

- Output is a service provided by the operating system: *textual* output is very standardized even across different platform, **graphics is not so stable**
- When you deal with graphical programs: expect installation headaches, portability glitches, etc.



A real world example

On the Win10 platform where I try the programs:

- the current `matplotlib` library doesn't work: it installs, but when used it complains about missing DLLs (Dynamic libraries shared among the programs)
- according to the "Internet" ... the problem can be solved by installing Visual Studio (!): I didn't try...
- The last version I found working properly is 3.3.1: I will stick to that (thanks to the virtual environments this is not a problem)
- On other platforms (Linux, Mac): no problems, but better if we use all the same version

BTW, it depends on `numpy`

PyQB

Monga

Matplotlib

Graphical commands

OO plotting



PyQB

Monga

Matplotlib

Graphical commands

OO plotting

- You need always two objects: a Figure and a Axes
- plotting happens on axes, framed in a figure
- very flexible: you can add plots on the same axis, or you can have many axes collected in a single figure

Basic example



PyQB

Monga

Matplotlib

Graphical commands

OO plotting

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

x = np.linspace(-2*np.pi, 2*np.pi, 100)

fig, ax = plt.subplots()

ax.plot(x, np.sin(x))

fig.show()
```

Many different types of charts



PyQB

Monga

Matplotlib

Graphical commands

OO plotting

If `ax` is a `Axes`

- Scatter-plots `ax.scatter`
- Bar-plots `ax.bar`
- Histograms `ax.hist`
- 2D `ax.imshow`



PyQB

Monga

Matplotlib

Graphical commands

OO plotting

- add labels, legends, titles
- add a grid
- combine multiple plots on the same axis
- combine multiple axes on the same figure

Save your pictures!



PyQB

Monga

Matplotlib

Graphical commands

OO plotting

A Figure can be saved in a file with `savefig`. You should keep in mind the difference between:

- bitmap formats (`png` `jpg` ...): the file is matrix of pixels
- vector formats (`svg` `pdf` ...): the file is a set of instructions to reproduce the picture, less portable but it can be magnified