Matplotlib
A python plotting suite

R. David
david@unistra.fr
Direction Informatique
28/09/2011
Outline

- What is Matplotlib?
- Matplotlib basics
- Useful extensions
  - Images
  - Basemap
  - Going 3D
- Conclusion
What is Matplotlib?

From [http://matplotlib.sourceforge.net/](http://matplotlib.sourceforge.net/): *matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.*

Not only:

- for plotting: contains numerical functions mimicking matlab (useful in interactive environments)
- can be used for animations
- for 2D: has some *extensions* for 3D
- a library: rather a suite. It has serveral interfaces
What is matplotlib

- As several python Numerics package, it relies on numpy
- Several others dependencies :
  - Output formats ⇒ *renderers* :
    - png library with high-quality anti-graining (Agg)
    - SVG
    - GDK (Gimp)
  - combined with Graphical user interfaces ⇒ *backends* :
    - QT, TK, GTK, macosX
- Several namespaces for the same functions
What is matplotlib

Examples (from the website)

9 lines of code

46 lines

91 lines

8 lines
Outline

- What is Matplotlib?
- Matplotlib basics
- Matplotlib as an interactive tool
- Useful extensions
  - Images
  - Basemap
  - Going 3D
- Conclusion
Matplotlib basics

- Matplotlib basics functions are included in the pyplot sub-package:
  
  ```python
  import matplotlib.pyplot as plt
  ```

- pyplot is designed to look like matlab

- basic command: `plot`

- plots lines with optionally markers

- markers and linestyle are documented using the `help(plt.plot)` statement

- The plot command applies to the last plot you modified (stateful)
Matplotlib basics

- Plot example:

  ```python
  plt.plot([1, 2, 3, 4])
  ```

- Defaults behaviour:
  - Line style: blue, continuous
  - X data: from 0, 1, 2, 3, computed from the number of values to plot

- Without defaults:

  ```python
  plt.plot([0, 1, 2, 3], [1, 2, 3, 4], 'b-')
  ```
The plot function returns an variable with type Lines2D, which can then be modified.

```python
l = plt.plot([1, 2, 3, 4])
# Python style
plt.setp(line, color='r', linewidth=2.0)
# Matlab style
plt.setp(line, 'color', 'r', 'linewidth', 2.0)
```

Same function, different argument style.
There are several other functions available in the plt namespace

List available using `help (plt.plotting)`:

- autocorrelation,
- contour
- histograms
- pie charts
- plotfile

Some of the functions make useful computations on data

Matplotlib uses numpy arrays
Up to now, the basic workflow was as follows:

```python
import matplotlib.pyplot as plt
plt.plot(something)
```

In order to use this in a (text) program, you have to:

- use a backend without a gui
- add a `savefig` command

The program rewrites into:

```python
import matplotlib as mp
mp.use('agg')  # mp.use(): matplotlib configuration
import matplotlib.pyplot as plt
plt.plot(something)
plt.savefig('something.png')
```
Interacting with Matplotlib

- Onclick (zoom)
- Default GUI
- ipython
  - autocorrelation,
  - barbs
  - contour
  - histograms
  - pie charts
  - plotfile
- Some of the functions make useful computations on data
- List available using help (plt.plotting)
Up to now, the basic workflow was as follows:

```python
import matplotlib.pyplot as plt
plt.plot(something)
```

In order to use this in a (text) program, you have to:
- use a backend without a gui
- add a `savefig` command

The program rewrites into:

```python
import matplotlib as mp
mp.use('agg')  # mp.use() : matplotlib configuration
import matplotlib.pyplot as plt
plt.plot(something)
plt.savefig('something.png')
```
In interactive mode, plot commands spawn a window:

```
import matplotlib.pyplot + matplotlib.mlab
from matplotlib.pylab import *
```

In interactive mode, it is common to import a `pylab` module.

```python
Pylab = matplotlib.pyplot + matplotlib.mlab
```

```python
from matplotlib.pylab import *
```

*brings everything in the same namespace*
Matplotlib is integrated in several python scientific environments providing editing and interactive features:

- spyder, an IDE with matlab-like features, [http://packages.python.org/spyder/](http://packages.python.org/spyder/)

The `-pylab` flag of ipython imports `*` from `matplotlib.pylab` at interpreter startup, thus bringing a lot of useful functions in the `__main__` namespace

`ipython -pylab` is the recommended way of using ipython for exploration

`pydev` dans Eclipse
The `matplotlib.mlab` module provides 10 functions with Matlab names, implemented on top of NumPy, with Matlab names.

It provides other helper functions to deal with text-files containing numerical values or some math functions needed by the authors of Matplotlib.

`mlab` is included when using `pylab`.

The module can be used with `mpl_toolkits.exceltools`: record arrays can be written out to Excel files with the `rec2excel` method.
Outline

- What is Matplotlib?
- Matplotlib basics
- Useful extensions
  - Images
  - Basemap
  - Going 3D
- Conclusion
Outline

- What is Matplotlib?
- Matplotlib basics
- Useful extensions
  - Images
  - Basemap
  - Going 3D
- Conclusion
As working on images is a common task, there are helpers in matplotlib. These helpers rely on the Python Imaging Library (PIL), [http://www.pythonware.com/products/pil/](http://www.pythonware.com/products/pil/). The functions are contained in the matplotlib.pyplot module. They work on images as 3D (R,G,B) arrays of real numbers in [0:1] and images are stored as numpy arrays.
Displaying an image

```python
import matplotlib.pyplot as plt
plt.imshow(plt.imread("ifremer.jpg"))
```

Surprising, isn't it (not a bug in the slide)?
Useful extensions - Images

- More functions:
  - changing the colormap
  - rotating, resizing image
  - interpolating on pixels

- Note that the imaging facilities are a subpart of the pyplot package, tightly coupled with PIL for dealing with formats other than PNG
Outline

- What is Matplotlib?
- Matplotlib basics
- Useful extensions
  - Images
  - Basemap
  - Going 3D
- Conclusion
Similar to matlab mapping toolbox

Transforms coordinate of map projections into data than can be plotted by matplotlib (pre-processing)

Contains political boundaries, rivers, shorelines, taken from the GMT suite

Connected to standard formats:
- OpenDAP for data exchange
- NetCDF (via pure python library)
- Shapefiles

Written by Jeffrey Whitaker
(http://www.esrl.noaa.gov/psd/people/jeffrey.s.whitaker/)
Create a projection (29 available):

```python
from mpl_toolkits.basemap import Basemap

# setup lambert azimuthal equal area basemap.
# lat_ts is latitude of true scale.
# lon_0, lat_0 is central point.
m = Basemap(width=12000000, height=8000000,
            resolution='l', projection='laea',
            lat_ts=50, lat_0=50, lon_0=-107.)
```

Plot data (with coordinates) on the projection
29 Available projections can be listed accessing online help (65 in Matlab)

```python
import mpl_toolkits.basemap
print basemap.supported_projections
```

- Azimuthal, Polyconic, Gnomonic, Mollweide, Transverse, North-Polar Lambert, Gall, Miller, Mercator, Stereographic, North-Polar Stereographic, Geostationary
- Near-Sided, van der Grinten, Lambert Azimuthal Equal Area, McBryde-Thomas, Sinusoidal, South-Polar, Lambert, North-Polar Azimuthal Equidistant, Equidistant, Cylindrical, Oblique Albers, South-Polar, Orthographic, Cassini-Soldner, South-Polar, Robinson
The Basemap object instances contains several functions, including:

- (filled) contour drawing
- geographic drawings: coastlines, boundaries, great circles, rivers, parallels, meridians...
- `is_land(x, y)`: `x, y` are in projection coordinates, returns True/False
- reading and plotting shapefiles
- computing earth areas in the shadow at a given time
- making nice plots: adding images as map background
Using Nasa Blue Marble as a background:

```python
import mpl_toolkits.basemap
b = mpl_toolkits.basemap.Basemap()
b.bluemarble()
```
Adding political boundaries

```python
b.drawcountries()
```
The shapefile interface is:

```python
b.readshapefile("landeareas", "lands", drawbounds=True, zorder=None, linewidth=0.5, color='k', antialiased=1, ax=None)
```

- `shapefile`: basename of the 3 components of the file
- `name`: (string) name of the attribute that will be added to the Basemap. This attribute holds the data of the shapefile

Land boundaries on the Antarctic
The data held by the "lands" attribute of the Basemap instance is made of a list of vertices:

```python
print b.lands
[(-96.36414655167492, 68.47130097246468),
 (-96.70338706497459, 68.48849366276391),
 (-97.14621337170688, 68.57308500473383),...]
```

- a list of attributes of the shapes:

```python
print b.lands_info

{'RINGNUM': 1, 'Name': 'Greenland', 'SHAPENUM': 4}
{'RINGNUM': 1, 'Name': 'Australia', 'SHAPENUM': 5}
```

- and a list of (low-level) geoslib polygons:
The data hold by the "lands" attribute of the Basemap instance is made of a list of vertices

```
print b.lands
[(-96.36414655167492, 68.47130097246468),
 (-96.70338706497459, 68.48849366276391),
 (-97.14621337170688, 68.57308500473383),...]
```

a list of attributes of the shapes

```
print b.lands_info
{'RINGNUM': 1, 'Name': 'Greenland', 'SHAPENUM': 4}
{'RINGNUM': 1, 'Name': 'Australia', 'SHAPENUM': 5}
```

and a list of (low-level) geoslib polygons
Outline

- What is Matplotlib?
- Matplotlib basics
- Useful extensions
  - Images
  - Basemap
  - Going 3D
- Conclusion
Basic matplotlib handles only plotting 2D datasets \( x,y \)

Plotting of \( x,y,z \) relies on the mplot3d toolkit, computing 2D projections of 3D data

It works by adding a 3d projection axe to the figure (since matplotlib 1.0.0)

3D plotting methods are available from the \texttt{Axes3D} instance

Gallery:
Get a 3D axe:

```python
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig=plt.figure()
axes3d=fig.gca(projection='3d')
axes3d=fig.gca(projection='3d')
```

From this instance, choose between:

```python
bar3D, contour3D, plot3D, plot_surface,
plot_wireframe, scatter3D, text2D, text3D,
```
Each plotting function uses 3 arrays as an input

The shape of these arrays depend on the plot

For 3D line plots, bar plots, scatter plots, X,Y,Z are 1D array

length = number of points to be plotted
For wireframe plots, surface plots, X, Y are 2D coordinates matrices, Z is a 2D array containing the values.
Outline

- What is Matplotlib?
- Matplotlib basics
- Useful extensions
  - Images
  - Basemap
  - Going 3D
- Conclusion
Matplotlib:

- is the python package to keep in mind when it comes to plotting
- tries to mimic matlab
- is tightly coupled to numpy and other top python libraries
- is integrated in ipython, pythonxy and other IDE