

# matplotlib - 2D and 3D plotting in Python

J.R. Johansson (robert@riken.jp) <http://dml.riken.jp/~rob/>

The latest version of this [IPython notebook](http://ipython.org/notebook.html) (<http://ipython.org/notebook.html>) lecture is available at <http://github.com/jrjohansson/scientific-python-lectures> (<http://github.com/jrjohansson/scientific-python-lectures>).

The other notebooks in this lecture series are indexed at <http://jrjohansson.github.io> (<http://jrjohansson.github.io>).

```
In [1]: # This line configures matplotlib to show figures embedded in the notebook,  
# instead of opening a new window for each figure. More about that later.  
# If you are using an old version of IPython, try using '%pylab inline' instead.  
%matplotlib inline
```

## Introduction

Matplotlib is an excellent 2D and 3D graphics library for generating scientific figures. Some of the many advantages of this library include:

- Easy to get started
- Support for  $LaTeX$  formatted labels and texts
- Great control of every element in a figure, including figure size and DPI.
- High-quality output in many formats, including PNG, PDF, SVG, EPS, and PGF.
- GUI for interactively exploring figures *and* support for headless generation of figure files (useful for batch jobs).

One of the of the key features of matplotlib that I would like to emphasize, and that I think makes matplotlib highly suitable for generating figures for scientific publications is that all aspects of the figure can be controlled *programmatically*. This is important for reproducibility and convenient when one needs to regenerate the figure with updated data or change its appearance.

More information at the Matplotlib web page: <http://matplotlib.org/>

To get started using Matplotlib in a Python program, either include the symbols from the `pylab` module (the easy way):

```
In [2]: from pylab import *
```

or import the `matplotlib.pyplot` module under the name `plt` (the tidy way):

```
In [3]: import matplotlib.pyplot as plt
```

## MATLAB-like API

The easiest way to get started with plotting using matplotlib is often to use the MATLAB-like API provided by matplotlib.

It is designed to be compatible with MATLAB's plotting functions, so it is easy to get started with if you are familiar with MATLAB.

To use this API from matplotlib, we need to include the symbols in the `pylab` module:

```
In [4]: from pylab import *
```

## Example

A simple figure with MATLAB-like plotting API: