

```
In [2]: # These are a special "magic commands" that gives us access to a most of the  
# mathematical functions and graphing capabilities we will need.  
%matplotlib notebook  
%pylab
```

Using matplotlib backend: nbAgg

Populating the interactive namespace from numpy and matplotlib

Introduction to Jupyter notebooks

Many (most?) of our programming assignments will be done in Jupyter Notebooks.

The name "Jupyter" is an mash-up of three common programming languages - **Julia**, **Python**, and **R**. All three languages can be used in a "notebook".

What is a notebook?

- A notebook is an interactive platform in which you can enter text (including LaTeX), code fragments and plots in a single document.
- Notebooks are run through your web browser. The actual "behind the scenes code" for a notebook is human-readable JSON code.
- These slides were created using a Jupyter notebook!
- This is an example of a "markdown" cell.
- This is another cell

Why use Python/Jupyter?

The Jupyter Project (of which Jupyter Notebooks are a part) is an open source project millions of users and contributors worldwide.

- **Python** is now the most widely used computing language in data science and machine learning.
- **Jupyter Notebooks** allow users to mix narrative text with executable code, equations and visualization to create a living document that supports reproducibility and enables exploratory learning.

I know Matlab. Do I now have to learn Python?

The short answer : My sample codes will all be in Python.

All homework will be turned in using Python and Jupyter notebooks (explained below).

The good news is that if you are fairly comfortable with Matlab, you should find that using Python will be reasonably straight forward. We will "pre-install" a set of commands that are very similar to Matlab commands. You will also find that learning Python is a valuable computing skill.

What is a notebook?

A Jupyter notebook (or just a "notebook" for this course) is a web-based document (typically hosted locally) that consists text, code, numerical results, plots, hyperlinks, JPEG images and so on.

- Notebooks are launched from the command line, and navigated using your web browser. So all you need to run a notebook is a suitable laptop/desktop, and the Jupyter/Python code base.
- *Instructions for installing Anaconda Python are at the bottom of this note book, if you'd like to jump ahead.*

Below, we go through some basic ideas in using a notebook.

Notebook mechanics : Magic commands

To get access to a wide variety of commands, many of which will be familiar to you from Matlab, you will want to include the following two commands at the top of every new notebook. Commands that start with `%` or `%%` are called "Magic commands".

```
%matplotlib inline
%pylab
```

Scroll up to the top of this notebook to see these two magic commands in action. Note that they should not be indented (they are indented in this cell so they are formatted correctly).

These commands will allow us to use commands familiar to many of you who have learned Matlab.

Notebook mechanics : Input cells

There are two kinds of notebook cells used for input :

- **Markdown** cells contain narrative text and mathematical equations, expressed using LaTeX.
 - Markdown is a simple formatting language that allows you to use **bold** face font, *italics*, and inline code fragments like this one `print("Hello, world")` .
 - Equations can be entered using Latex
 - Sections can be highlighted to add structure to your documents
 - Numbered and bulleted lists can be created
 - Hyperlinks can be provided to link to sections within a document, or to external links on the web.
 - Include links to images, videos and so ...
- **Code** cells are used to input executable code.

To render text and equations, and to run executable code, hit "shift-return" from the cell.

The text for this slide is entered into a **Markdown** cell.

Input cells : Markdown cells

This cell is a markdown cell which can be used to enter text and equations, like this one:

$$f(x) = \frac{\sin(x)}{x^2 + 1}, \quad x \in \mathcal{R}$$

Or inline equations $x = 3.5$, $y = f(x)$.

Render the text and equations in Markdown cells using "shift-return".

Part of this course is learning how to provide narrative text so that others can follow your work.

Input cells : Code cells

Code cells contain actual code (Python, in our case) and can be executed by hitting "shift-return".

Code cells are indicated by `In[XX]` , where `XX` is the counter for the order in which code in cells is executed. This counter is updated each time you execute code in a cell.

The first time you execute your code, the brackets `[]` will be empty.

```
In [5]: # **This cell** is a "code" cell and can be used to enter code. To run  
        the code, hit "shift-return"  
  
        # Set a variable named 'x' to a value  
        x = 3.5  
  
        # evaluate the function at the value 'x'  
        y = sin(x)/(x**2 + 1)  
  
        # Display the value of y  
        y
```

Out[5]: -0.026474205863367534

Notebook mechanics : Output cells

We have already seen that the results of our code fragments above appear in *output* cells, indicated by Out[XX] , where the XX matches the code cell counter that produces the results.

Another type of output is a plot or graph. Below, we create a plot of the function we defined above.

Plotting $y = f(x)$

To create a plot of our function, we first need to define the domain over which we will plot our function. Let's plot our function over the interval $x \in [-2\pi, 2\pi]$. Our new problem is then,

$$f(x) = \frac{\sin(x)}{x^2 + 1}, \quad x \in [-2\pi, 2\pi]$$

To create our plot, we need to do several things :

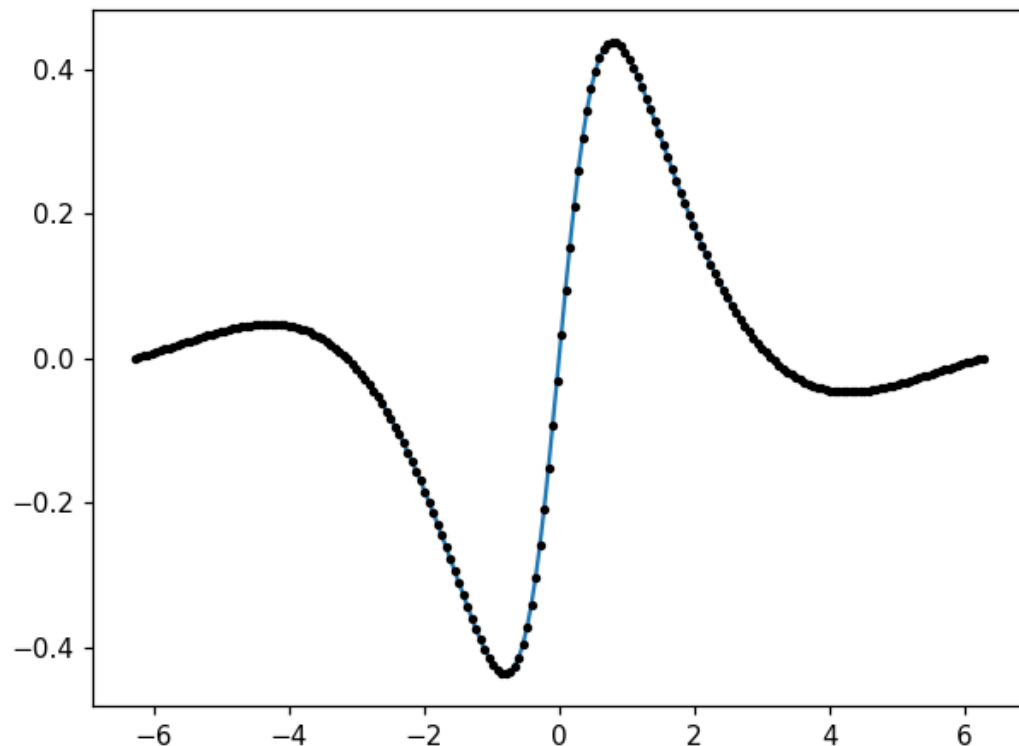
- Create a sequence of values $x_1, x_2, \dots, x_i, \dots, x_N$ at which to evaluate our function. The size of N will determine how smooth our plot is.
- Evaluate our function at each term x_i to get a corresponding sequence of values $y_i = f(x_i)$.
- Create a smooth graph by "connecting the dots" $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_N, y_N)$ in order that they appear in our sequence.

```
In [4]: figure(1)
        clf()

        # Create a sequence of 200 x values equally spaced in the interval [-2*pi, 2*pi]
        N = 200
        x = linspace(-2*pi, 2*pi,N);

        # evaluate the function at the sequence of x values
        y = sin(x)/(x**2 + 1)

        # "connect the dots"
        plot(x,y)
        plot(x,y, 'k.', markersize=5)
```



```
Out[4]: [<matplotlib.lines.Line2D at 0x7fa0f082ae10>]
```

We used two specialized commands :

- The keyword `pi` , which is just $\pi = 3.14159\dots$
- The use of the command `linspace(a,b,N)` which creates a sequence of N equally spaced points in the interval $[a, b]$.
- Notice that the plot is interactive. By moving your mouse in the plot window, the (x,y) values will appear. We can also use interactive buttons to zoom into the plot, "pan", or save the plot to a PNG file.
- Notice that as you zoom in, the plot becomes less smooth.

How do I set up notebooks on my laptop?

The easiest way to install the Jupyter notebook functionality on your notebook is to install Anaconda Python.

Here are links to an excellent set of tutorials for getting the Anaconda Python installed on your system.

[Installation for Windows 10 \(https://www.youtube.com/watch?v=5mDYijMfSzs\)](https://www.youtube.com/watch?v=5mDYijMfSzs)

[Installation for Linux \(https://www.youtube.com/watch?v=DY0DB_NwEu0&t=3s\)](https://www.youtube.com/watch?v=DY0DB_NwEu0&t=3s)

[Installation for OSX \(https://www.youtube.com/watch?v=daVgEXjv6DE&t=10s\)](https://www.youtube.com/watch?v=daVgEXjv6DE&t=10s)

The site will explain exactly what Anaconda is, how to download it, and install it, and how to launch a Jupyter notebook from your own laptop.

The video is about two years old, so some version numbers may not be current.

- Install Python 3.X (the most current release of Python).
- If you do not have "administration privileges" on your laptop/desktop, install for "you only".
- The videos are done by the same person, so it might be useful to watch each video, since the narrator includes slightly different comments in each video.
- We will not be using "JupyterLab", "Spyder" or Visual Studio code. You are of course free to install these, but we are mainly interested in the Jupyter notebooks.

Homework Assignment #1

1. Install Anaconda Python on your personal laptop.
2. Download this notebook from the Canvas website and make sure you can run it.
 - Use "shift-enter" to run each cell individually, or
 - Run the entire notebook in one step by using the "kernel" menu. Choose "kernel -> Restart & Run All"
3. See notebook "Number_system" for a few more homework problems.

Please have 1) and 2) done by Thursday, and try 3). If you run into trouble, we can go over questions on Thursday.