Plots
PyLab

• Plotting is one of Python’s greatest strengths
  – It is very easy to plot data in a visually appealing format
  – Also very easy to customize your plots and change how they look

• Plotting is done using the PyLab module
  – Basically contains a lot of Matlab plotting functions
  – Matlab is a quick-prototyping language mostly used by engineers
  – In Matlab inspecting and manipulating data is key, so its plotting functionality is unmatched by other languages
  – Unlike Matlab, Python is an actual programming language but it also has Matlab’s plotting capabilities
Installing PyLab

- To get `pylab`, you will need to install `numpy` and `matplotlib` modules
- Depending on your OS, different procedures are followed
- First make sure you have a new version of Python
- Python installs packages using a tool called PIP
- On Mac/Linux type in your terminal:
  
  ```
  pip install --user numpy
  pip install --user matplotlib
  ```

- On Windows type in your terminal:
  
  ```
  python --user pip install --user numpy
  python --user pip install --user matplotlib
  ```

- Both assume that python was added to your path
- Might also need to install `setuptools`
Basic structure of plotting code

• First, you need to create a figure
  – This basically creates the window of your figure
  – Example (assuming import pylab): pylab.figure(1)

• Then call the plot function
  – Example: pylab.plot(myList1, myList2)
  – myList1 and myList2 are lists of numbers of the same size (could be other collections as well)
  – myList1 contains x-coordinates and myList2 contains y-coordinates

• Finally, call show to display the figure
  – Example: pylab.show()
  – You will not see anything before you call show!

• IDLE example: simplePlot.py
Can also have multiple figures

- Repeat procedure from last slide but put a different number in each figure
- You should only call `show` once at the end
- IDLE example: `multipleFigures.py`
Can also save figure to file

- Instead of `show`, call `savefig`
  - Example: `pylab.savefig("myFig.png")`
- It has to be in png format
- Will only save the last figure
  - PyLab has a notion of current figure
  - The last figure that you executed `pylab.figure` on
- IDLE example: `saveFig.py`
Figure attributes

• You have control over many figure attributes
• Can change the title, axis labels, font size, axis numbers, scale, etc.
  – Explore the examples for different options I used
• Can also have multiple subplots in one figure
  – Use the `subplot(x, y, cur)` function
  – `x, y` are the dimensions of your subplot matrix
  – E.g., if `x = 2, y = 3`, then you have 6 subplots, arranged in 2 rows of 3
  – `cur` specifies which subplot you are working on right now
• IDLE example: principal.py