Data Visualization

COMPASS Workshop

Instructors = [“Vikram Manikantan”,
“Erik Wessel”,
“Vasilis Paschalidis”]

Today’s Schedule

Tuesday

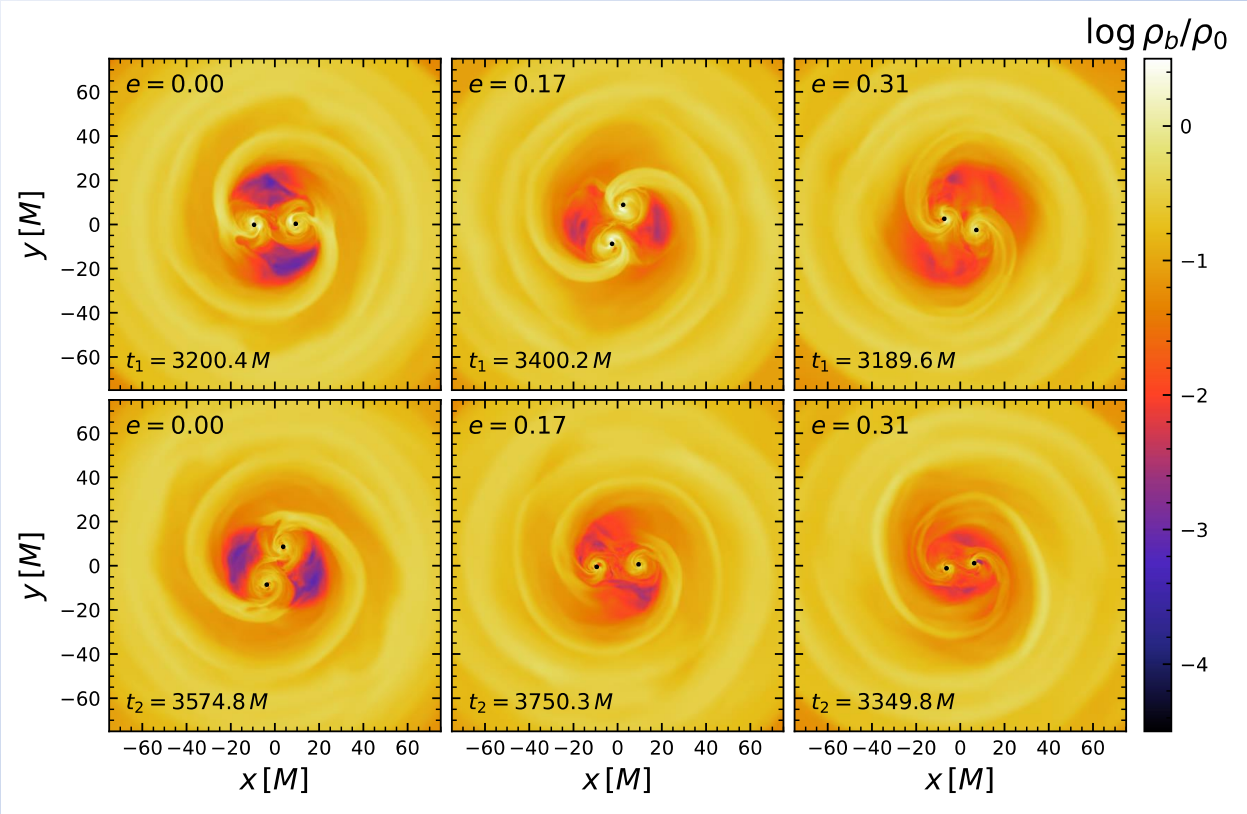
10am-10:30am: Python Loops & Functions

10:30am-noon: Matplotlib/Numpy

noon-1pm: Lunch (provided)

1pm-2pm: Overleaf/Latex hands-on experience

Contents



Google is your friend…

Today's Notebook

tinyurl.com/compassnumpy

Data Visualization

Using `numpy` and `matplotlib`

Contents

Python Packages

`Numpy` and `Matplotlib` are python packages. A python package is a library of code written by somebody else that will make your life easier (usually).

In this case, Numpy (pronounced num-pie) is a package that provides support for large arrays and high level mathematical functions (amongst other things).

Matplotlib, as you might have deduced from its name, is a library for data visualization with *lots* of functionality.

```
[1] # to be able to use these functions, we have to use the code below.

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

Python Packages

- A collection of functions and code ‘packaged’ together
- Sometimes called a ‘library’ of code
- Different packages have different purposes

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

we talked about functions

```
def calc_quadratic()
```

a function is like a book

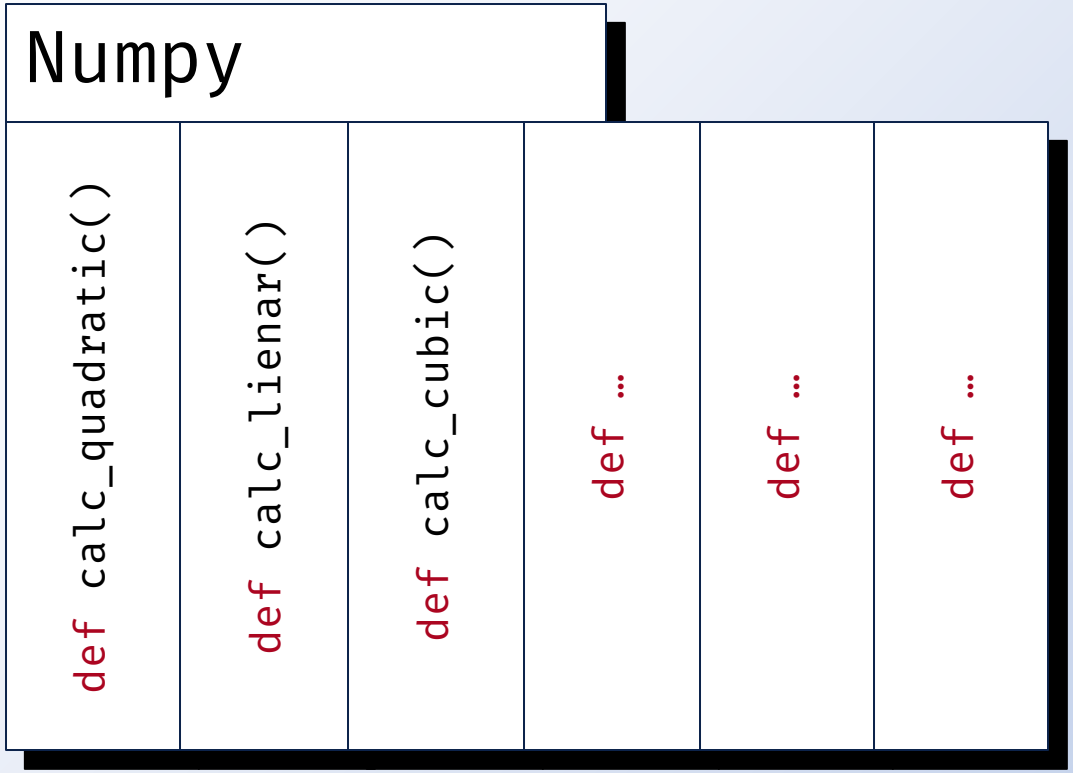
```
def calc_quadratic()
```

a package is a library of functions

```
def calc_quadratic()
def calc_lienar()
def calc_cubic()
def …
def …
def …
```

and many more…

I can name my package



Numpy

- High-level math functions
- Support for large-arrays
- Fundamental for science

```
>>> import numpy as np
>>> np.sin(0)
0
>>> np.cos(0)
1
>>> np.array([1,2,3])
[1 2 3]
```

Aside: Documentation

- Description of what code accomplishes in Human-readable language
- More descriptive than commenting
- Example of commenting:

```
# this is a comment:
# this code takes the
# variables a and b and adds
# them together

>>> a = 5
>>> b = 3
>>> print(“a + b = ”, a+b)
a + b = 8
```

Example of Documentation

- Describes a function by its <u>inputs</u> and <u>outputs</u>
- Neatly organized for future users and developers
- [Online Example]

```
def multiply(a,b):
"""
inputs:
  a: double, first number to be
multiplied
  b: double, second number
returns:
   double: product of inputs
"""
return a*b
```

Numpy Arrays

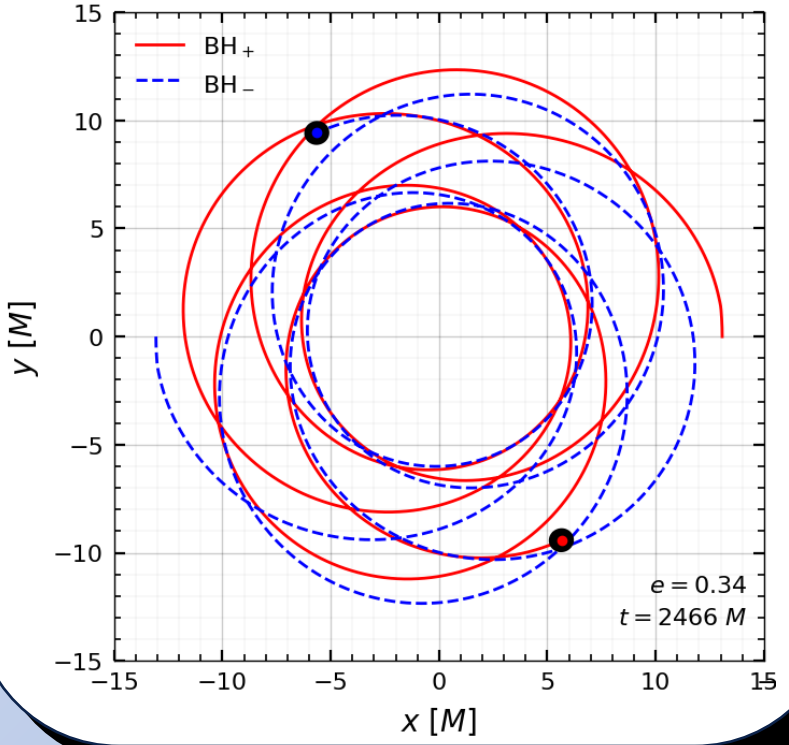
- Like the arrays we looked at yesterday but more powerful
- We can treat arrays like a number, and do **arithmetic** on them
- Let’s look at the notebook…

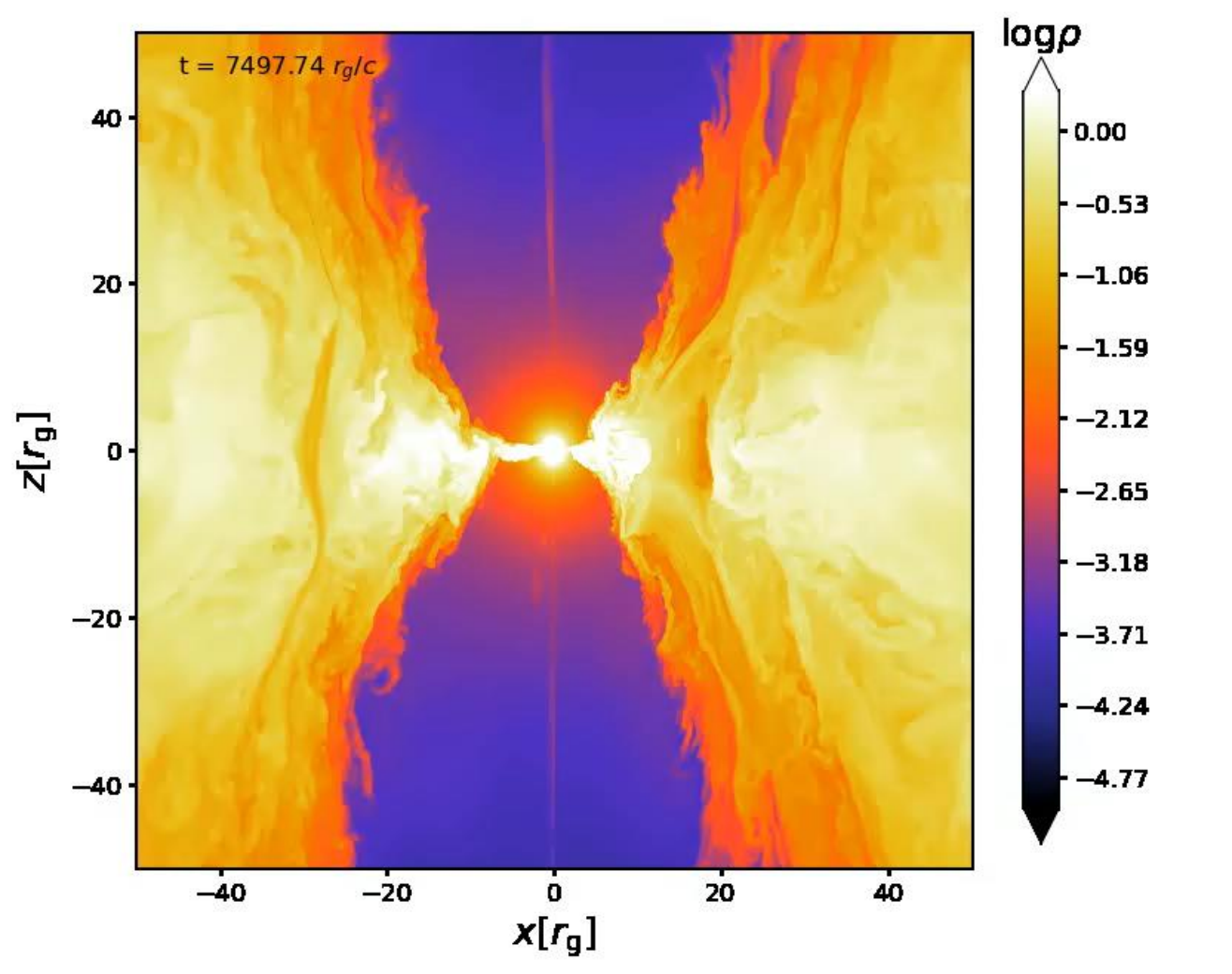
```
>>> import numpy as np
>>> arr = np.array([1,2,3])
>>> print(arr)
[1 2 3]
>>>
>>> arr*5
[5 10 15]
>>>
```

Matplotlib

- Library for creating visualizations
- Static, animated, and interactive
- Very comprehensive

```
>>> import matplotlib.pyplot as plt
>>> plt.plot(x1, y1, color = 'blue')
```

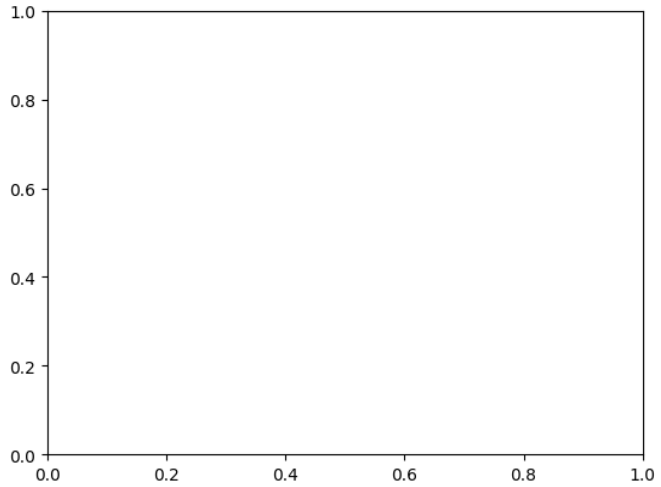



t = 7497.74 r_g/c
$\log\rho$
0.00
−0.53
−1.06
−1.59
−2.12
−2.65
−3.18
−3.71
−4.24
−4.77
40
20
0
−20
−40
$z[r_g]$
−40
−20
0
20
40
$x[r_g]$

Creating the Graph

- First, import the package
- ax stands for ‘Axes’. This is what you will be drawing your graph ‘on’

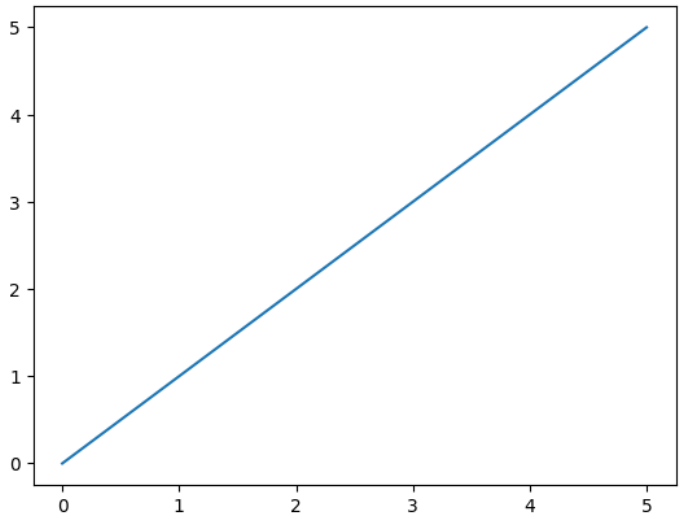
```
>>> import matplotlib.pyplot as plt
>>> fig, ax = plt.subplots()
```



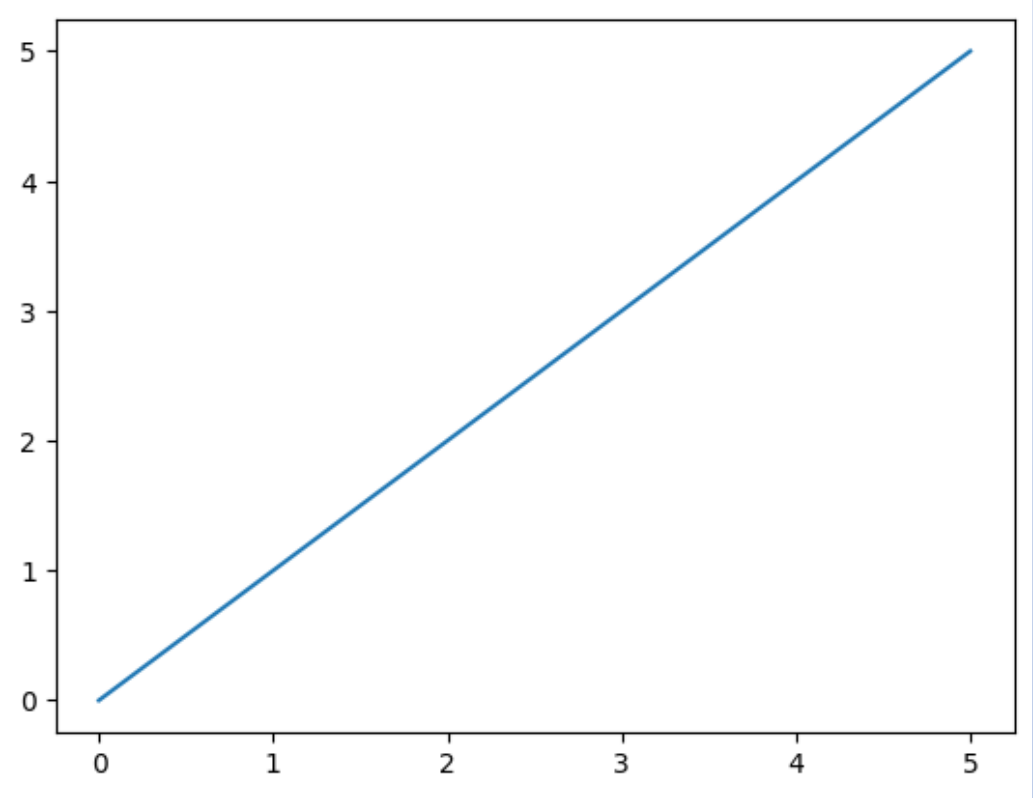
Adding Lines

- Now we need something to plot
- Make our own data

```
>>> data_x = [0, 1, 2, 3, 4]
>>> data_y = [0, 1, 2, 3, 4]
>>> ax.plot(data_x, data_y)
```

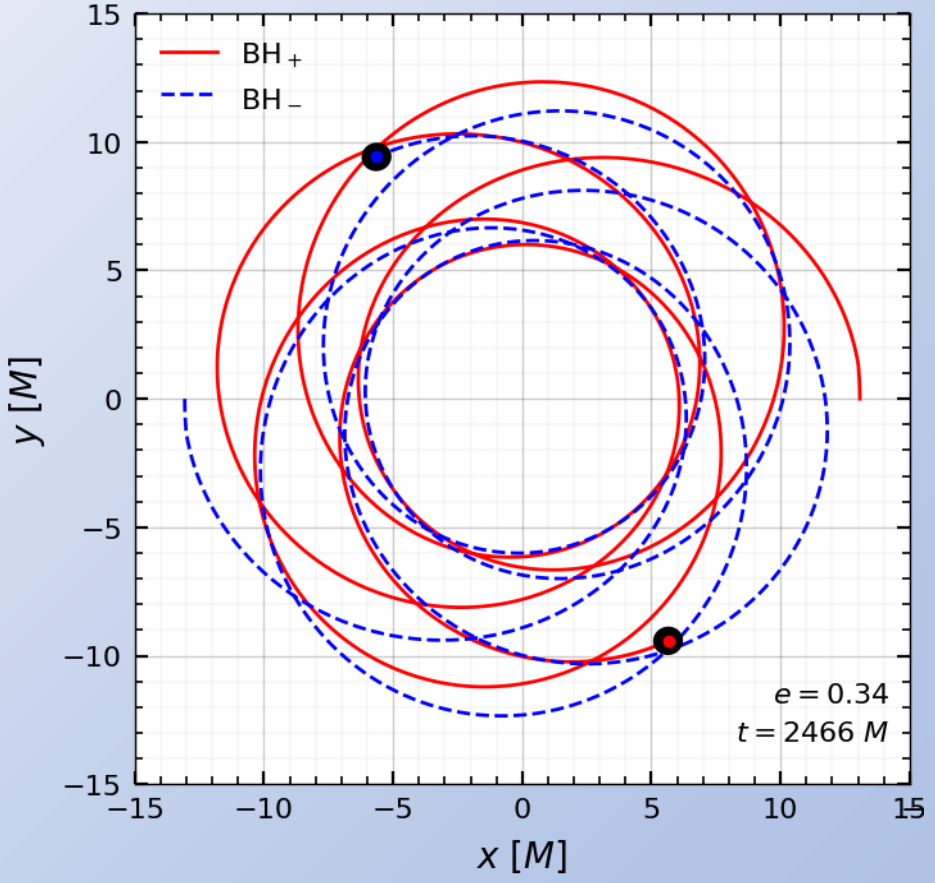


How do we make this better?



Additional Elements

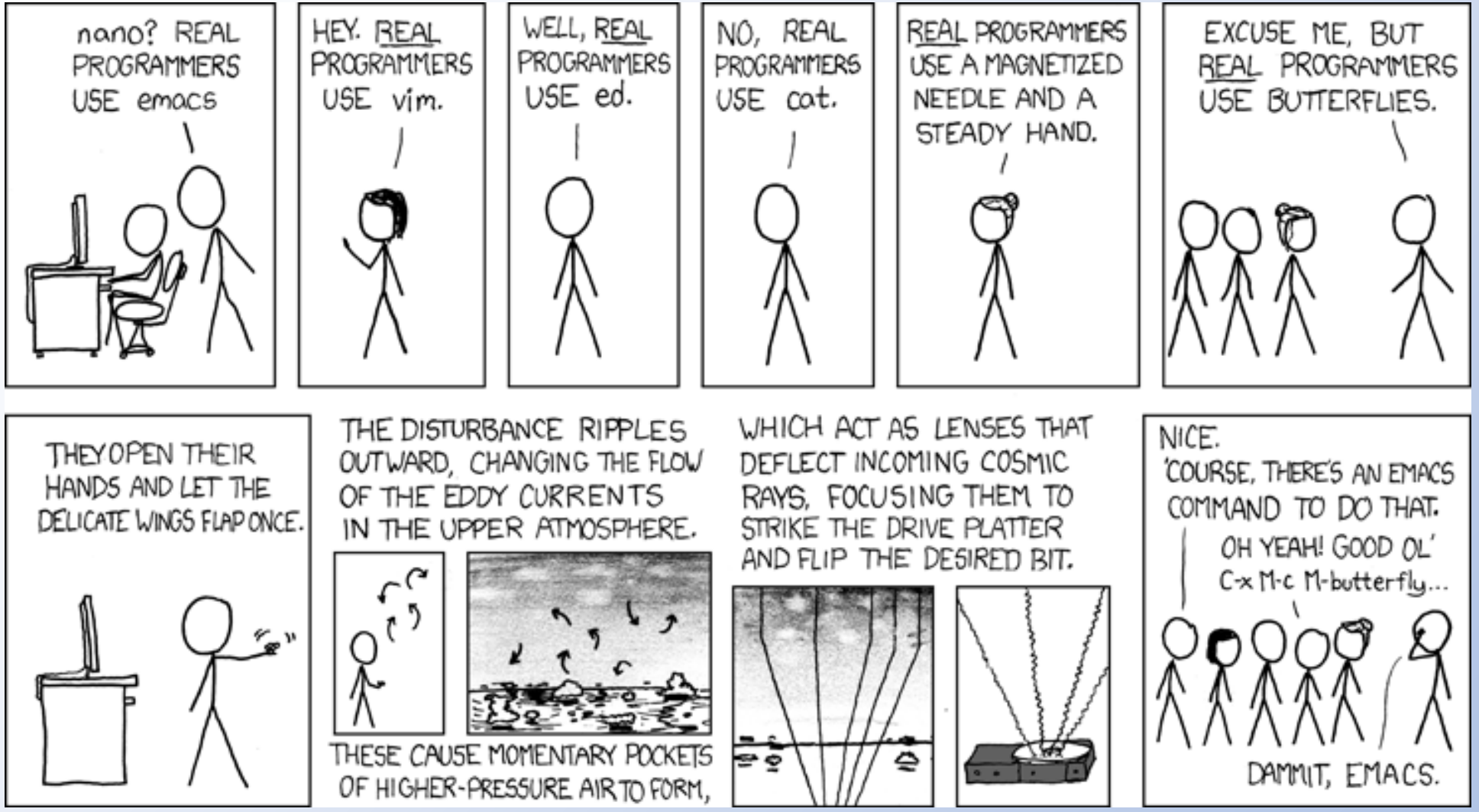
- Title
- Axes labels
- Legends
- Colors
- Linestyles
- Grid?
- …and many more

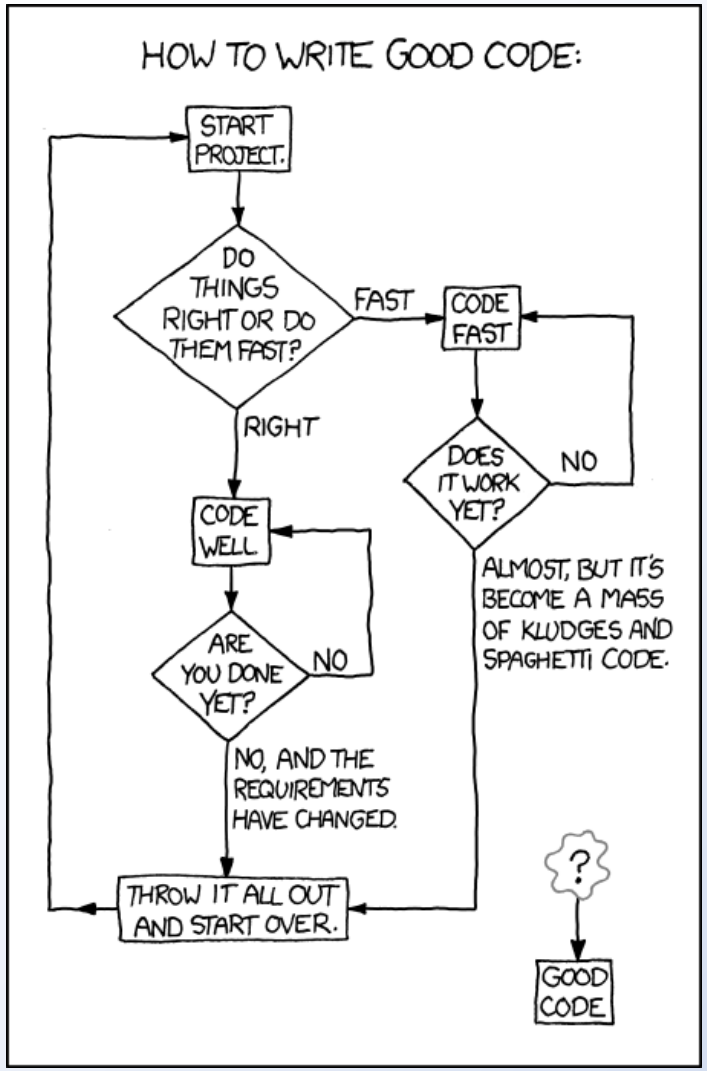


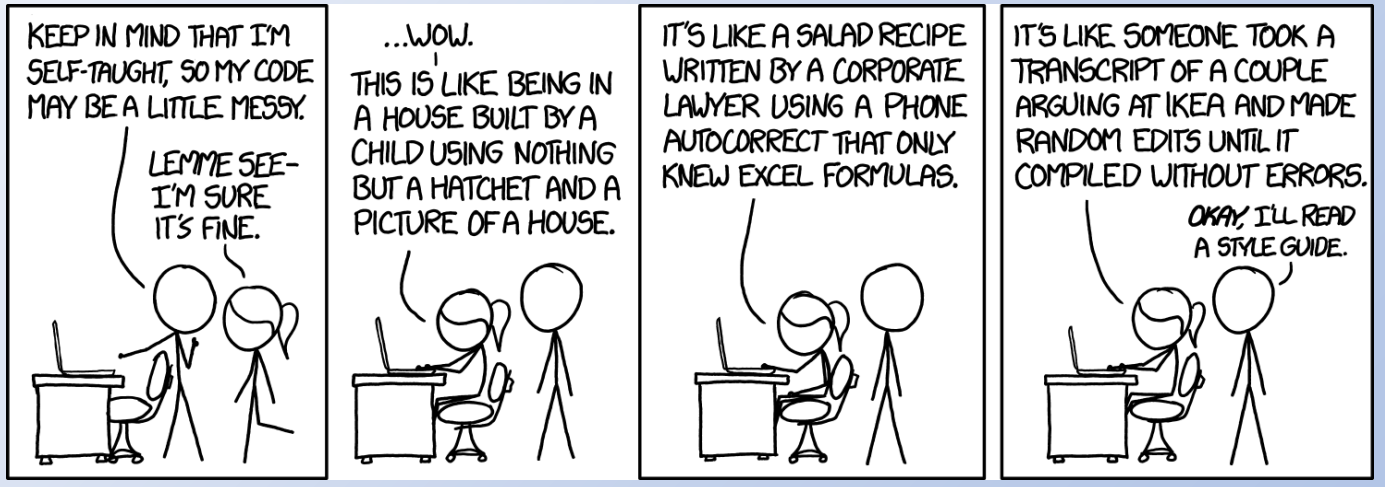
Generating Data

- You can create filled arrays with numpy
- numpy.arange(start, end, step)
- numpy.linspace(start, end, number of elements)

```
>>> import numpy as np
>>> np.arange(0,10,2)
[0 2 4 6 8]
>>>
>>> np.linspace(0, 10, 6)
[0 2 4 6 8 10]
>>>
```


nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.
EXCUSE ME, BUT REAL PROGRAMMERS USE BUTTERFLIES.
THEY OPEN THEIR HANDS AND LET THE DELICATE WINGS FLAP ONCE.
THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.
THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,
WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.
NICE.
'COURSE, THERE'S AN EMACS COMMAND TO DO THAT.
OH YEAH! GOOD OL' C-x M-c M-butterfly...
DAMMIT, EMACS.


HOW TO WRITE GOOD CODE:
START PROJECT.
DO THINGS RIGHT OR DO THEM FAST?
FAST
CODE FAST
RIGHT
DOES IT WORK YET?
NO
CODE WELL
ALMOST, BUT IT'S BECOME A MASS OF KLUDGES AND SPAGHETTI CODE.
ARE YOU DONE YET?
NO
NO, AND THE REQUIREMENTS HAVE CHANGED.
THROW IT ALL OUT AND START OVER.
?
GOOD CODE


KEEP IN MIND THAT I'M SELF-TAUGHT, SO MY CODE MAY BE A LITTLE MESSY.
LEMME SEE- I'M SURE IT'S FINE.
...WOW.
THIS IS LIKE BEING IN A HOUSE BUILT BY A CHILD USING NOTHING BUT A HATCHET AND A PICTURE OF A HOUSE.
IT'S LIKE A SALAD RECIPE WRITTEN BY A CORPORATE LAWYER USING A PHONE AUTOCORRECT THAT ONLY KNEW EXCEL FORMULAS.
IT'S LIKE SOMEONE TOOK A TRANSCRIPT OF A COUPLE ARGUING AT IKEA AND MADE RANDOM EDITS UNTIL IT COMPILED WITHOUT ERRORS.
OKAY, I'LL READ A STYLE GUIDE.