Programming and Simulation II

Amy Lien

Goddard Space Flight Center

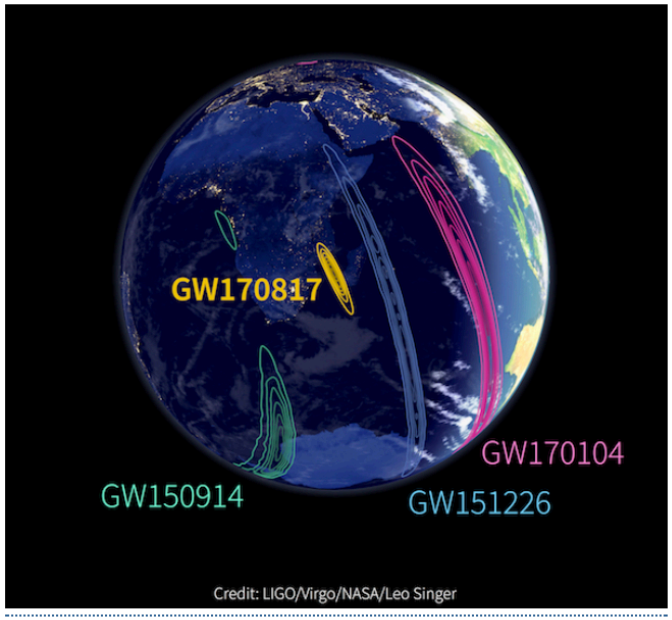
ASTR 288C, Lecture 7, 2017/10/16

Mid-term grade

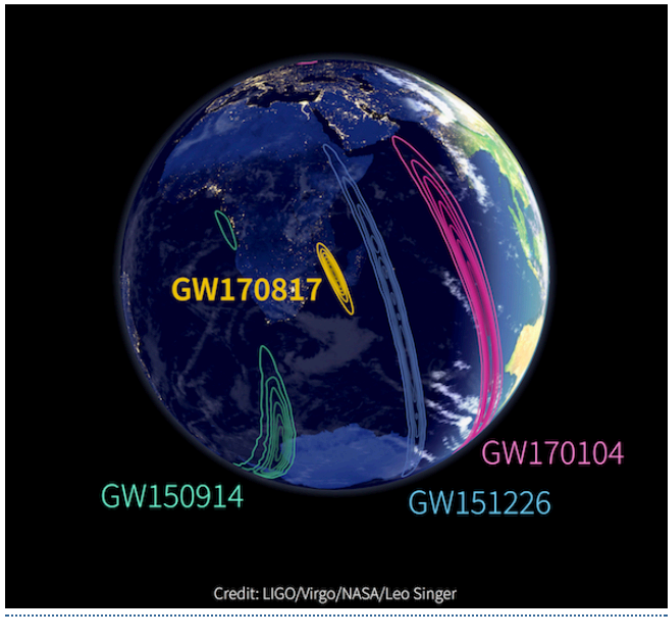
- Include all the grades so far
  - (Lab*15/60) + (Homework*45/60)
- If you think your grade isn't correct, calculate it yourself and contact the instructor ASAP.

97 <= grede <= 100	A+
93 <= grede < 97	A
90 <= grede < 93	A-
87 <= grede < 90	B+
83 <= grede < 87	B
80 <= grede < 83	B-
77 <= grede < 80	C+
73 <= grede < 77	C
70 <= grede < 73	C-
67 <= grede < 70	D+
63 <= grede < 67	D
60 <= grede < 63	D-

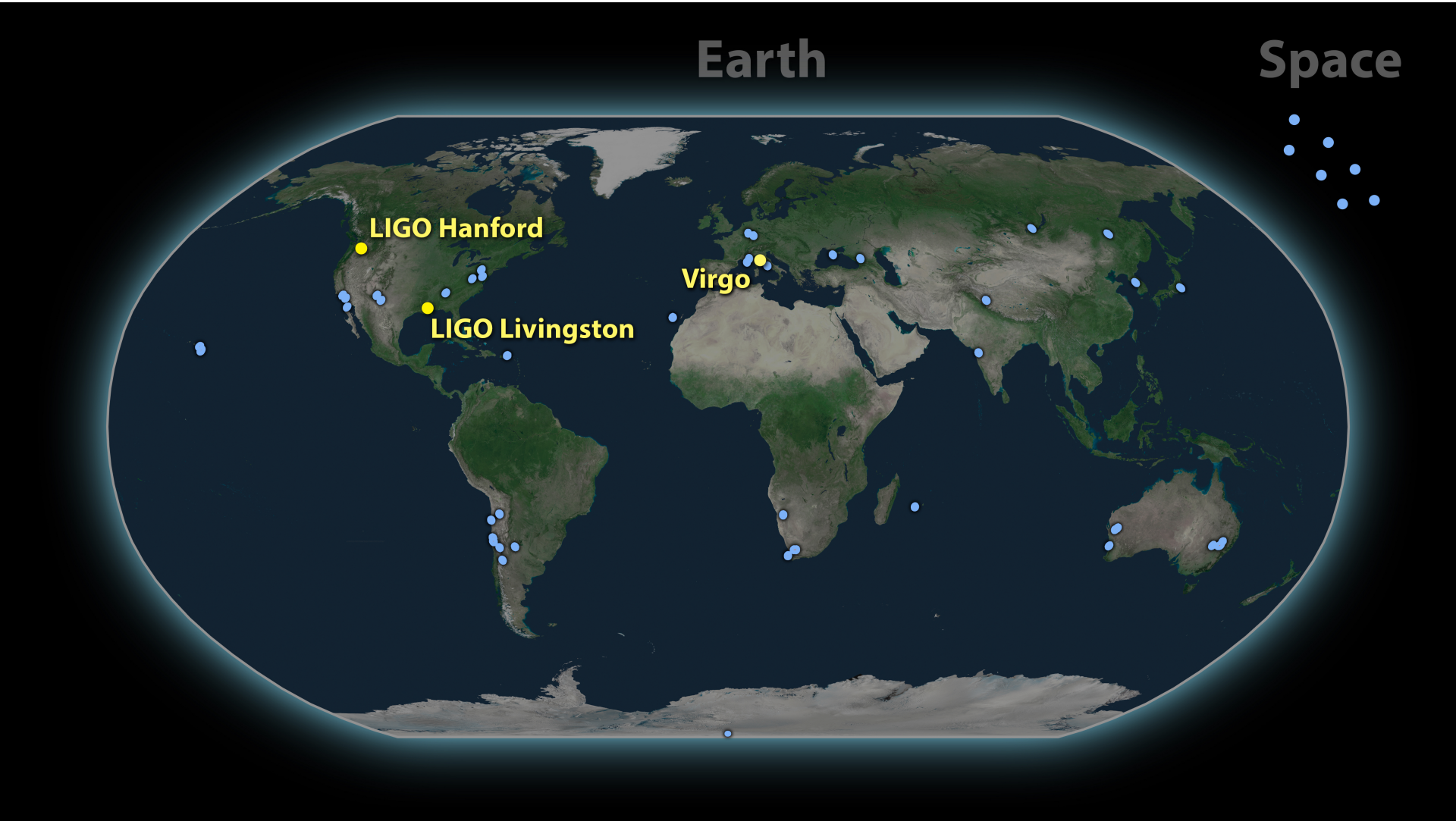
Gravitational Waves with E&M Counterparts



Gravitational Waves with E&M Counterparts

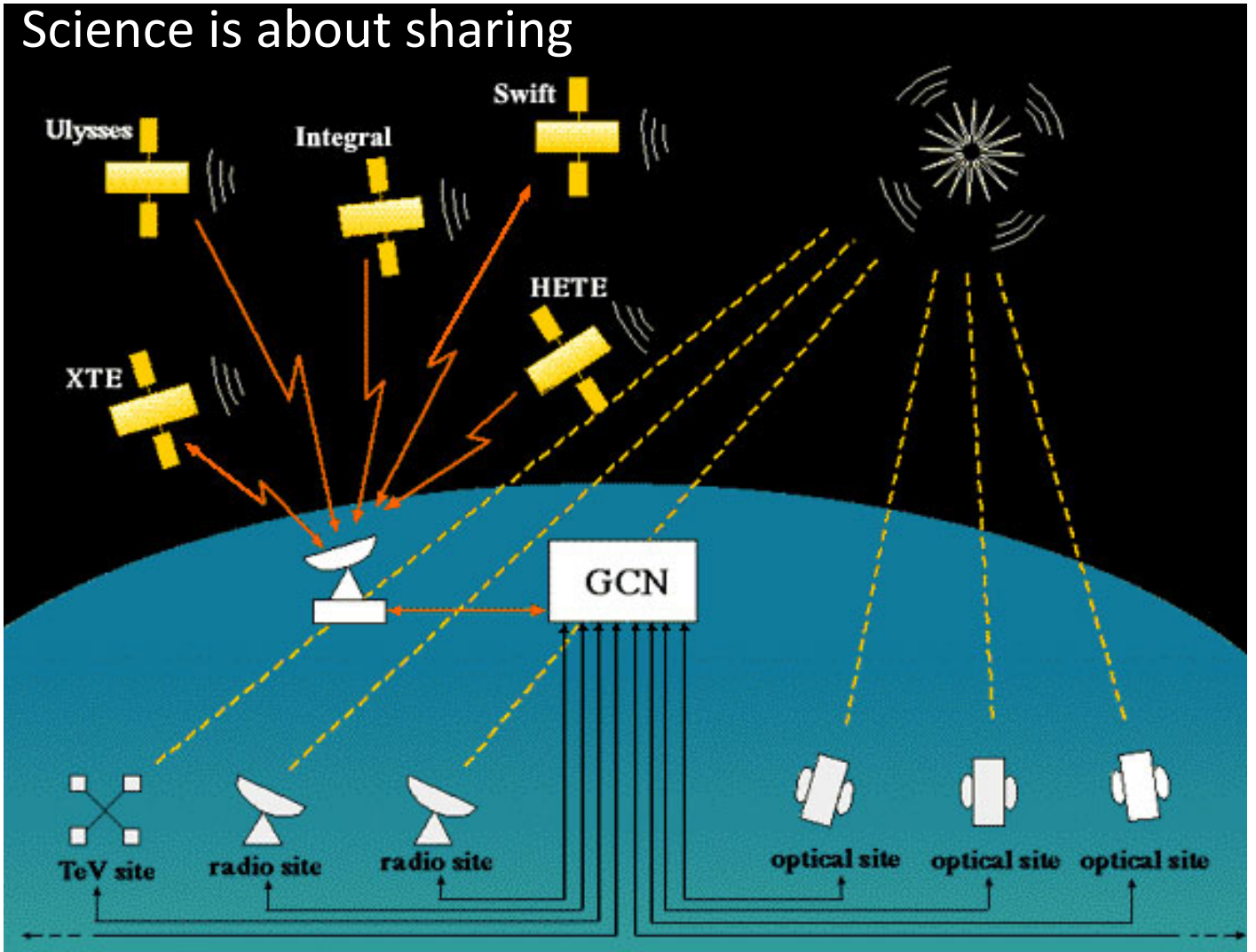


Gravitational Waves with E&M Counterparts

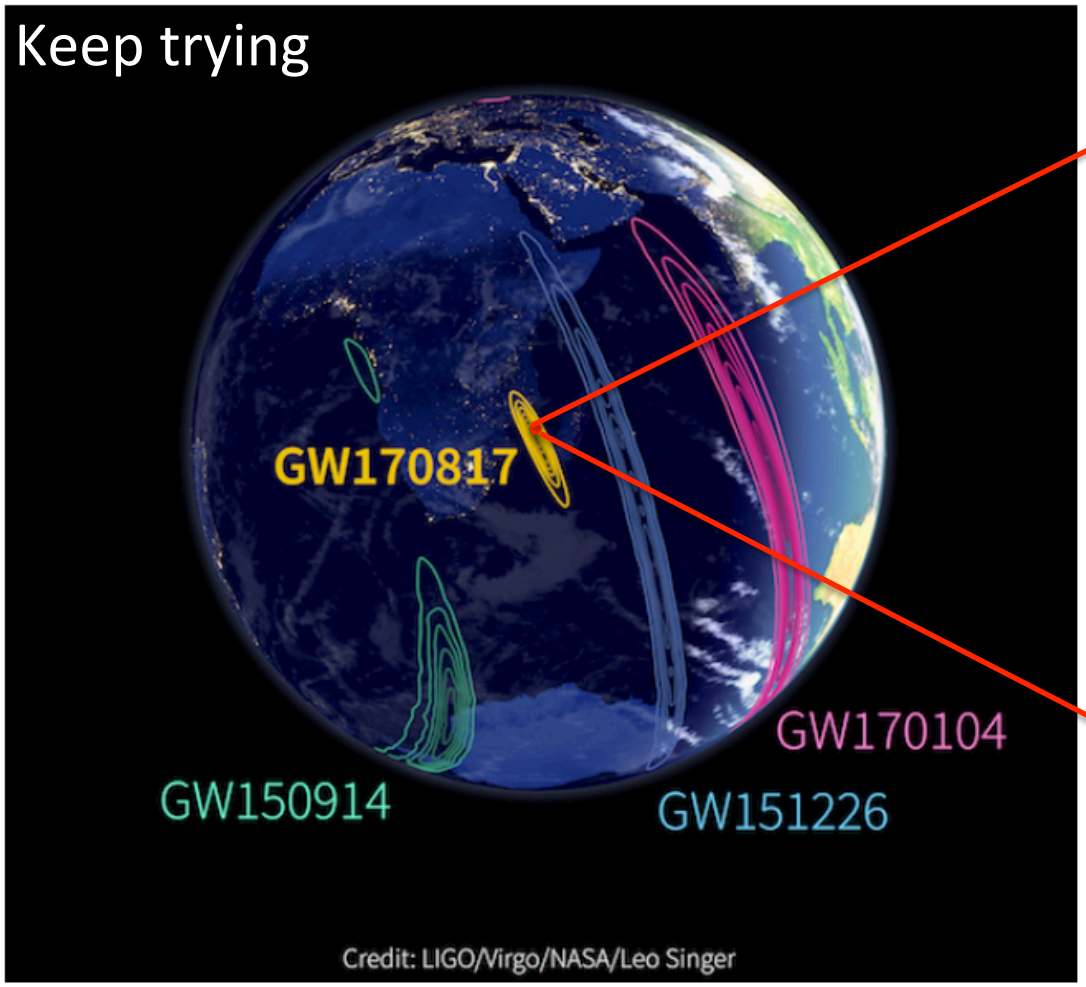


Gravitational Waves with E&M Counterparts

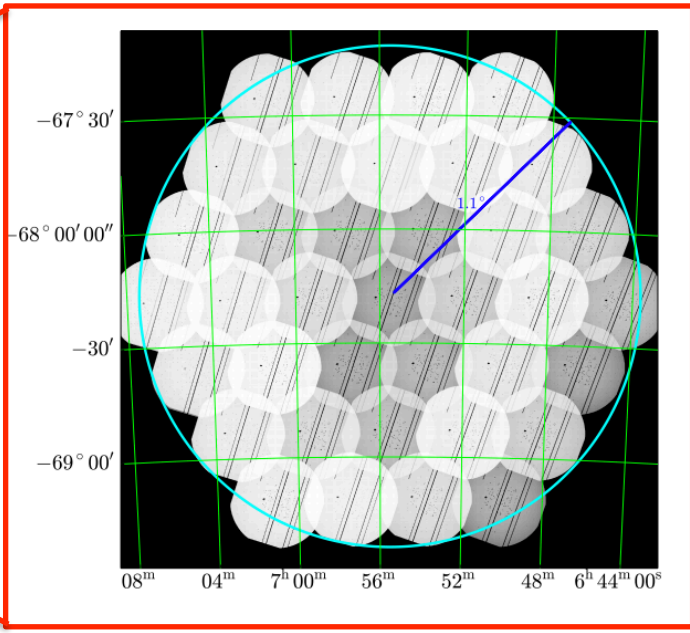
Gravitational Waves with E&M Counterparts



Gravitational Waves with E&M Counterparts



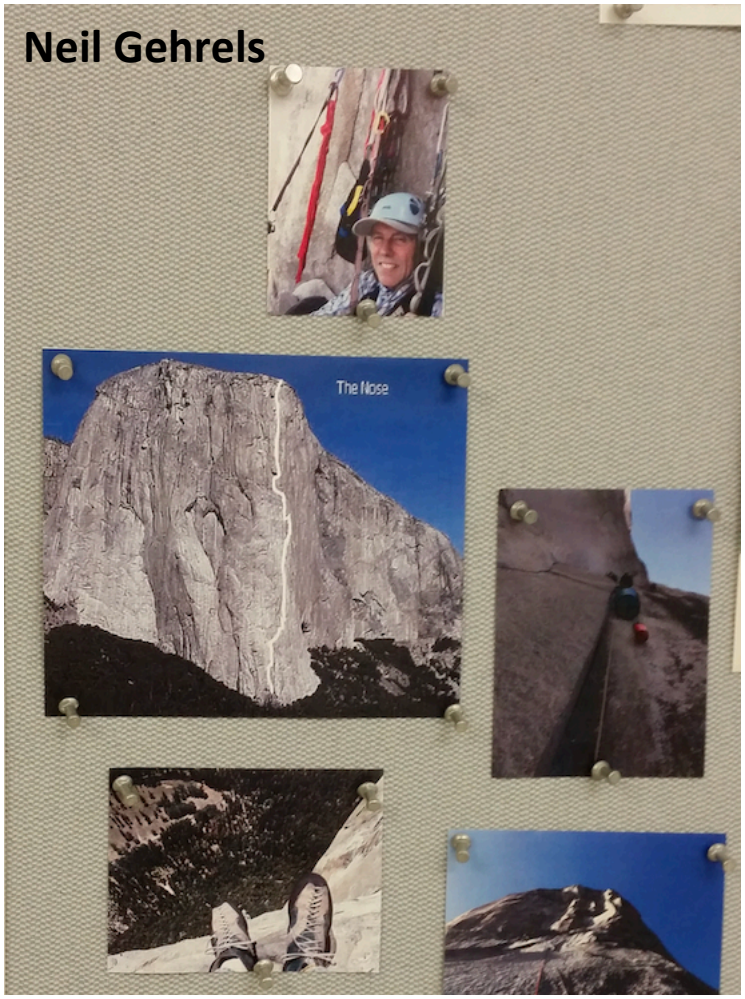
Swift XRT followup



37 tiles/observations

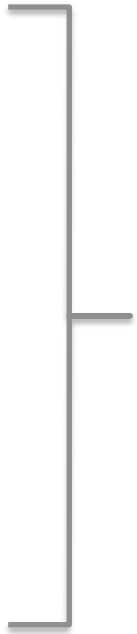
Evans et al. (2016)

Gravitational Waves with E&M Counterparts



Outline
Python I and II

- Basic structure
  - Print “Hello World!”
- For loop
- If statement
- Input and output (open files, save files)
- Logical elements (and, or, ==)
- Array (1d, 2d)
- Function
- Plot
- Managing FITS file in python



You can find the example python codes at /n/ursa/A288C/alien/python_template

A special note for Python coding

- For float number, use 2.0 instead of 2
- Python will read 2 as an integer, which might give you some hard-to-find bugs in division.

Function

- Example code: function_example.py

```
## Definine a function called "example_func",
## which has a variable x
## and return the value of y
def example_func(x):
        y = x*x
        return y

## Print out the value of example_func at x=2.0
print example_func(2.0)
```

- Output:

```
ursa% python function_example.py
4.0
```

Integration

- Example code: integration.py

```
## Import relevant packages
import sys
import math
from scipy.integrate import quad ## this is the package for integration

## Define a function
def example_func(x):
        y = x*x
        return y

## Integrating example_func from 0 to 3
result = quad(example_func, 0.0, 3.0)

## Print out the results,
## which is an array with the 0th element to be the integrated value
## and the 1st element to be the numerical error
print result
```

- Output

```
ursa% python integration.py
/usr/lib64/python2.6/site-packages/numpy/oldnumeric/__init__.py:11: ModuleDe
precationWarning: The oldnumeric module will be dropped in Numpy 1.9
  warnings.warn(_msg, ModuleDeprecationWarning)
(9.0000000000000018, 9.9920072216264114e-14)
```

Integration

- Example code: integration.py

```
## Import relevant packages
import sys
import math
from scipy.integrate import quad ## this is the package for integration

## Define a function
def example_func(x):
        y = x*x
        return y

## Integrating example_func from 0 to 3
result = quad(example_func, 0.0, 3.0)

## Print out the results,
## which is an array with the 0th element to be the integrated value
## and the 1st element to be the numerical error
print result
```

- Output

```
ursa% python integration.py
/Library/Python/2.7/site-packages/scipy/integrate/__init__.py:11: ModuleDe
precated: oldnumeric has been dropped in Numpy 1.9
  warnings.warn(_msg, ModuleDeprecationWarning)
(9.0000000000000018, 9.9920072216264114e-14)
```

Result of the integration

Numerical error

Integration

- A list of different integration packages can be found at

https://docs.scipy.org/doc/scipy/reference/tutorial/integrate.html

2-d array

- 2-d array

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}$$

2-d array

- 2-d array

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}$$

- Python's array (array inside array)

$$\begin{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 2 & 3 \end{bmatrix} \end{bmatrix}$$

2-d array

- Example code: array_2d.py

```
## First, create an empty array
test_array_2d = []



## Loop through the i-th element
for i in range(0,11):
        ## Fill in the i and j number in the current row of the array
        test_array_2d.append([])
        test_array_2d[i].append(i)
        test_array_2d[i].append(1)

## Print out the 2-d array
print test_array_2d
```

- Output

```
ursa% python array_2d.py
[[0, 1], [1, 1], [2, 1], [3, 1], [4, 1], [5, 1], [6, 1], [7, 1], [8, 1], [9, 1], [10, 1]]
```

2-d array

- Output

```
[ [0, 1],
  [1, 1],
  [2, 1],
  [3, 1],
  [4, 1],
  [5, 1],
  [6, 1],
  [7, 1],
  [8, 1],
  [9, 1],
  [10, 1] ]
```

FITS file

- Example code: fits_file_example.py

```
from astropy.io import fits ## This is the package that handles FITS file

## Open the FITS file
lc_fits = fits.open('sw00767284000b_1chan_1s_sort.lc')

## Read in data from the 1st extension
## as an array called lc_fits_data
## (the 1st extension is the light curve in this case)
lc_fits_data = lc_fits[1].data

## Go through the rows in lc_fits_data
## to print out the content
for i in range(len(lc_fits_data)):
        time = lc_fits_data[i][0] ## the 0th element of row i
        rate = lc_fits_data[i][1] ## the 1st element of row i

        print time, rate
~
```

Plotting

- Example code: plot_example.py

```
from matplotlib import pyplot as plt ## This is the package for plotting

## Open the file to plot the data
f_input = open('lc_example.txt','r')

## Go through each line in the file
## and append the time and rate numbers to arrays
time_array = []
rate_array = []
for line in f_input.readlines():
        ## Skip the comment line that begins with #
        if '#' not in line[0]:
                ## Separate columns using 'space'
                column = line.split()
                ## Assign column numbers to easy-to-read names
                time = float(column[0])
                rate = float(column[1])

                ## Append time and rate to arrays
                time_array.append(time)
                rate_array.append(rate)

## Close input file
f_input.close

## Plot rate_array vs time_array
## (time in x axis and rate in y axis)
plt.plot(time_array,rate_array,color='r', linestyle='-', label='Light Curve')
```

Plotting

- Example code: plot_example.py

```
from matplotlib import pyplot as plt ## This is the package for plotting

## Open the file to plot the data
f_input = open('lc_example.txt','r')

## Go through each line in the file
## and append the time and rate numbers to arrays
time_array = []
rate_array = []
for line in f_input.readlines():
        ## Skip the comment line that begins with #
        if '#' not in line[0]:
                ## Separate columns using 'space'
                column = line.split()
                ## Assign column numbers to easy-to-read names
                time = float(column[0])
                rate = float(column[1])

                ## Append time and rate to arrays
                time_array.append(time)
                rate_array.append(rate)

## Close input file
f_input.close

## Plot rate_array vs time_array
## (time in x axis and rate in y axis)
plt.plot(time_array,rate_array,color='r', linestyle='-', label='Light Curve')
```

Value to be plotted need to be in float

Plot error bars

- Example code: plot_errorbar.py

```
from matplotlib import pyplot as plt ## This is the package for plotting

## Open the file to plot the data
f_input = open('lc_example.txt','r')

## Go through each line in the file
## and append the time and rate numbers to arrays
time_array = []
rate_array = []
error_array = []
for line in f_input.readlines():
        ## Skip the comment line that begins with #
        if '#' not in line[0]:
                ## Separate columns using 'space'
                column = line.split()
                ## Assign column numbers to easy-to-read names
                time = float(column[0])
                rate = float(column[1])
                error = float(column[2])

                ## Append time and rate to arrays
                time_array.append(time)
                rate_array.append(rate)
                error_array.append(error)

## Close input file
f_input.close

## Plot rate_array vs time_array, with error bars for the rate.
## (time in x axis and rate in y axis)
plt.errorbar(time_array,rate_array,yerr=error_array,color='r', linestyle='-', label='Light Curve')
```

Plot histogram
(A.K.A. bar plot or binned plot)

- Example code: plot_histogram.py

```
from matplotlib import pyplot as plt

## Set up bin size
bin_size = 0.1 ## bin size
value_min = -1.0 ## value of the minimum bin
value_max = 1.0 ## value of the maximum bin

#creat bins array
bin_size_array=[]
i_max = int((value_max-value_min)/bin_size)
for i in range(0,i_max):
        value_point = i*bin_size + value_min
        bin_size_array.append(value_point)

## Make an empty array to add in counts
count = []
for i in range(0,i_max):
        count.append(0.0)

f_input = open('lc_example.txt','r')

for line in f_input.readlines():
        if '#' not in line[0]:
                column = line.split()
                error = float(column[1])

                ## go through each element in the binned array
                ## put the rate to the corresponding bin
                for i in range(0,i_max):
                        i_start = value_min + i*bin_size
                        i_stop = value_min + (i+1)*bin_size
                        if(i_start<=error<i_stop):
                                count[i] += 1

## Making bar plot
plt.bar(bin_size_array, count, bin_size, color='b', label='Histogram of light curve count')
```