Radio Astronomy Bootcamp
March 2023
Day 2

Intro to Python & Fourier Transforms

Akshatha K Vydula & Amy Zhao

Arizona State University

loco.lab.asu.edu

What is Python?

A reptile?

A programming language?

What's a programming language?

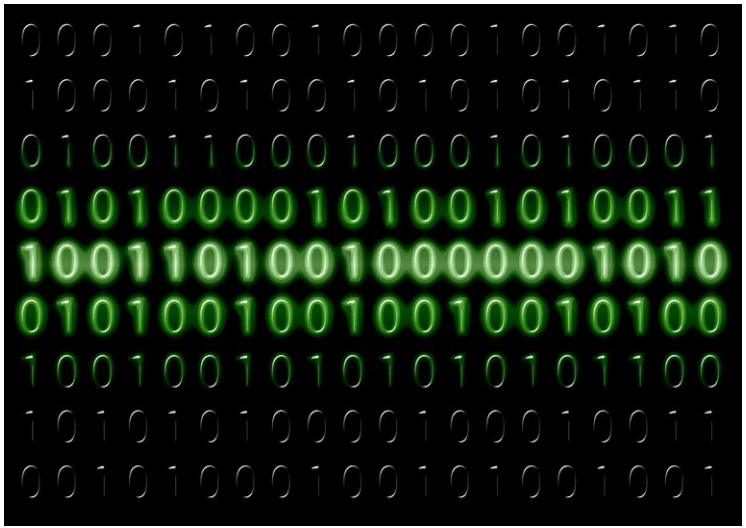
A Computer understandable language.

Our computers can't understand English, Spanish, Russian or French!

They only understand zeros and ones.

They are dumb!

Multiplicative effect of one billion people doing simple dumb calculations.



But

We don’t understand zeros and ones!

What do we do?

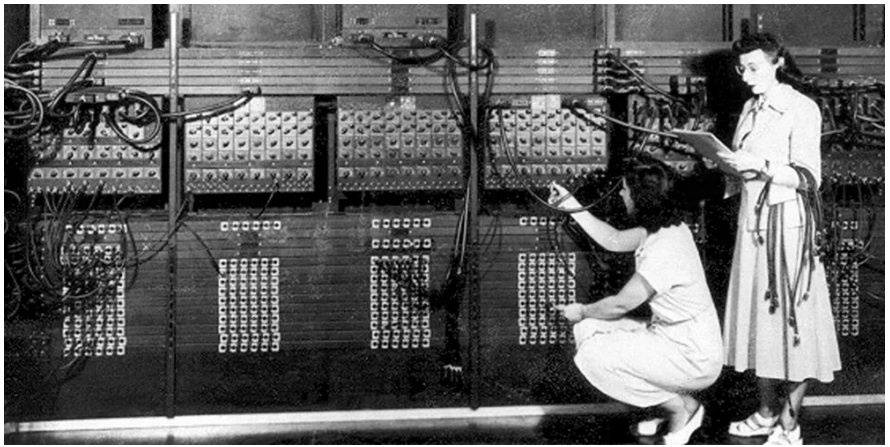
Find a common ground.

So what's python anyways?

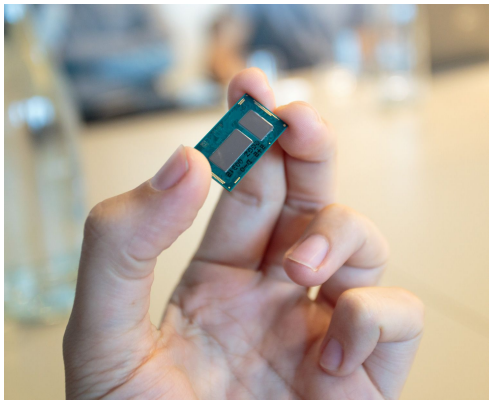
It is a high-level "english-like" language to talk to computer.

Why do we need python?

Humans need calculator and computer is the best one we've got.



Electronic Numerical Integrator and Computer, 1946



Today's computer

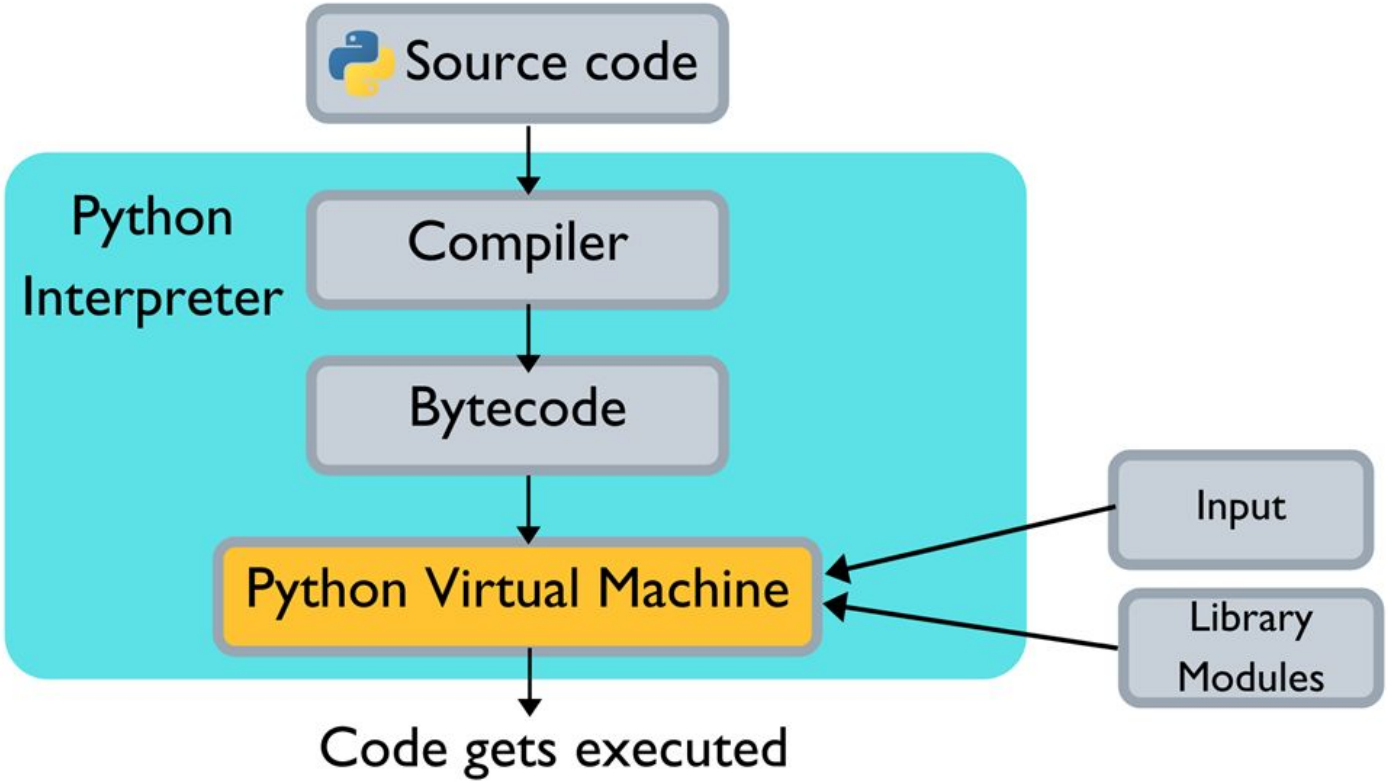
Why do we need programming?

$$p(\boldsymbol{\theta}|y_1,\ldots,y_n) = \frac{\frac{1}{(\sigma\sqrt{2\pi})^n}\exp\left[-\frac{1}{2}\sum_{i=1}^{n}\left(\frac{y_i-\mu(d_i)}{\sigma}\right)^2\right]\frac{1}{\sigma_\tau\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{\tau-\mu_\tau}{\sigma_\tau}\right)^2\right]\frac{1}{(b_{r_0}-0)}\frac{1}{(b_\sigma-0)}}{\int_{-\infty}^{\infty}\int_0^{b_{r_0}}\int_0^{b_\sigma}\frac{1}{(\sigma\sqrt{2\pi})^n}\exp\left[-\frac{1}{2}\sum_{i=1}^{n}\left(\frac{y_i-\mu(d_i)}{\sigma}\right)^2\right]\frac{d\tau}{\sigma_\tau\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{\tau-\mu_\tau}{\sigma_\tau}\right)^2\right]\frac{dr_0}{(b_{r_0}-0)}\frac{d\sigma}{(b_\sigma-0)}}$$

This is a posterior distribution of a problem I work on : very complex calculations + repeat calculations for 100,000 times.

Not a practical hand calculation ⇒ outsource the calculation to a computer.

So how does it work?



Source: C# Corner

What are libraries?

They are literally libraries in the programming world.

Ex: numpy, scipy, astropy, matplotlib etc

Every character has a meaning in programming

Ex: # is used for comments, = is used for assigning values to a variable

- Comments are statements ignored by the computer
- Used to make your code readable for humans

```
# Jane is my best friend
x = 10
# Austin is always late to the class
austin = 0
```

Computer ignores this

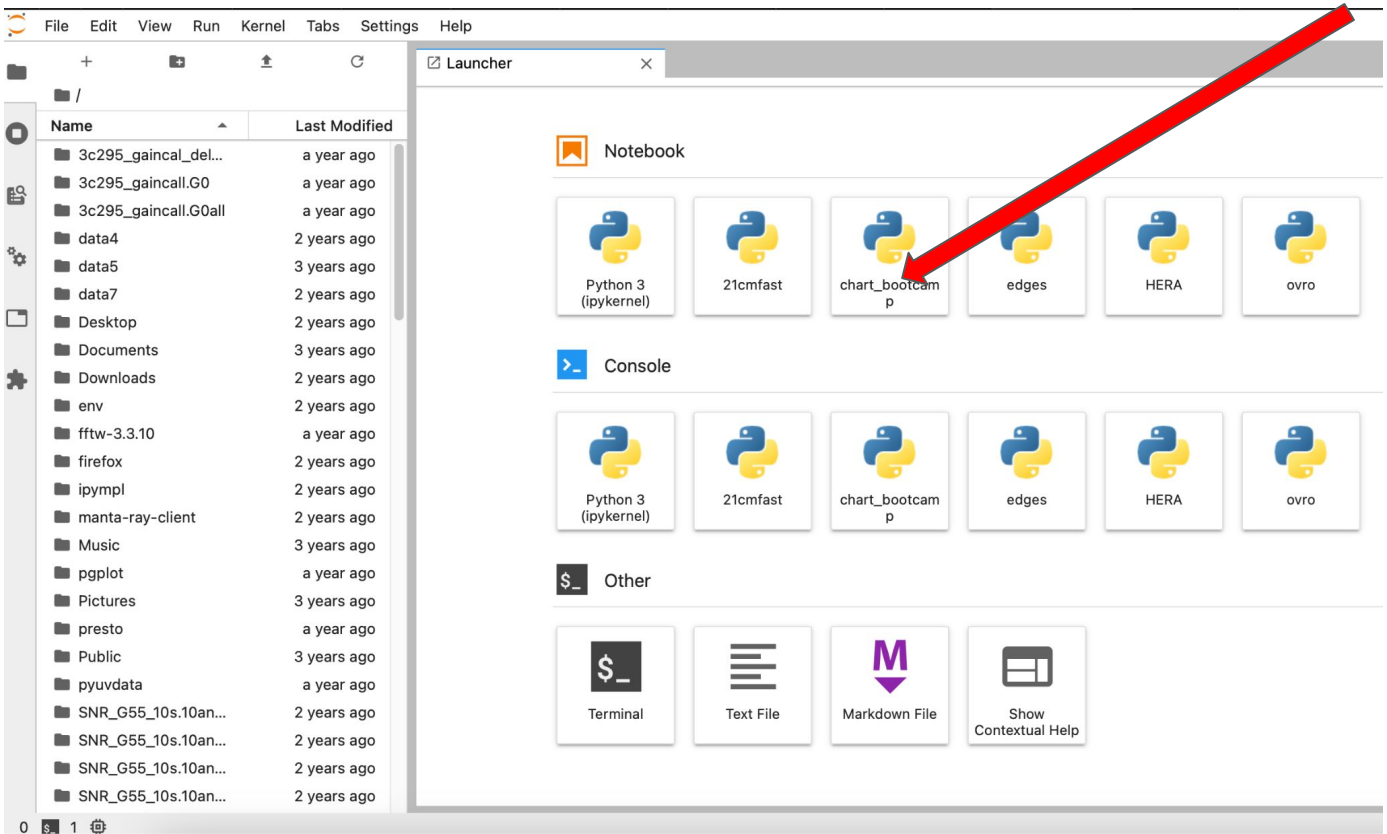
Computer does not ignore this

Computer ignores this

Computer does not ignore this

Let’s do some coding!

https://enterprise.sese.asu.edu/jupyter/ , Use your username and password.



Jupyter notebook : It is an interpreter

- Jupyter Notebooks are a Python based environment for programming
- Notebooks break down the program into cells which you can edit and run one at a time
- Compiler: Translator that also looks for syntax errors.
- Interpreter: Translates one line at a time

What is a compiler?

Translator that also looks for syntax errors.

What is an interpreter?

Translates one line at a time

What is a Bytecode?

Instructions in zeros and ones

Variables

Buckets to hold values for a calculation.

They can have types depending on type of the number or values

x =10
y=20
x=y

What's x?
What's y?

int 16 bits int 32 bits int 64 bits

Defining Variables

- Create variables by writing the name, followed by an “=” sign, and the value

```
d = 1.5
```

- Try to be descriptive with your variable names
- You may wish to add a comment after defining your variable

```
d = 1.5        #initial distance from the car (in m)
```

Assigning values

```
[2]: x = 10

[3]: y = 20

[4]: print(x)
     10

[5]: print(y)
     20

[6]: y=x

[7]: print(x)
     10

[8]: print(y)
     10
```

If Statements

- If statements allow you to have the code do something based on a logic statement
- Say we want the code to print “Hello” if k is greater than j and print “Goodbye” if not. (left)
- We can also use more than 2 conditions (right)

```
if k > j:
   print(“Hello”)
else:
   print(“Goodbye”)
```

```
if k > j:
   print(“Hello”)
elif:
   print(“Goodbye”)
else:
  print(“k and j are equal”)
```

If Statements (Logic statements)

```
--
```

[9]:
```
if(x>5):
    print('x is greater than 5')
else:
    print('x is lesser than 5')
```

```
x is greater than 5
```

[10]:
```
if(x>15):
    print('x is greater than 15')
else:
    print('x is lesser than 15')
```

```
x is lesser than 15
```

Operations

- Let's perform some simple calculations.
- You can change your values by addition,

```
x = y+50        #Just an example. Don't write this.
```

- or multiplication,

```
x = y*2             #Just an example. Don't write this.
```

- You can perform operations of multiple variables,

```
z = x/y      #calculate time to travel distance d1
```

- To see what you're calculating, you can use a print statement,

```
print z
```

List and Arrays: Sets of Numbers

- A list is a set of numbers, such as **[1,2,3,4,5]**
- You can create a list by explicitly listing its components

```
v2 = [100,120,140,160,180,200]
```

- You can also create an empty list with empty brackets

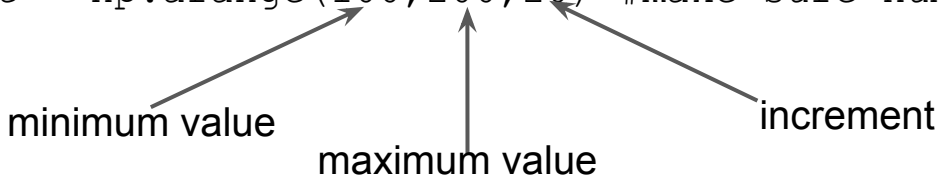
```
t1_list = []                    #empty time list
```

- And add individual values to the list later

```
t1_list.append(t1)              #append this list when you solve
for t1
```

- Arrays are a different way to store numbers and can be defined by minimum and maximum values

```
v2_list = np.arange(100,200,20) #make sure numpy was imported
```



Loops

- Set of commands which repeats until some end condition occurs
- Ex: Go to the University everyday until you graduate

```
for day in day_list:
   if(graduation == True):
       go_to_class = False
       exit
    else:
       go_to_class = True
```

```
for v2 in v2_list
    #conditional statement

      #calculate t1
      #save the value of t1
   in your list (append
   stmt)
      print("v2 = ", v2)
```

Include Packages

- Scripts used to accomplish a common task
- Pre-written by other users to make things easier

```
import numpy as np
#Importing package that lets python do some math functions

import matplotlib
#Python package that handles plotting graphs and figures

import matplotlib.pyplot as plt
# This is to let us just say 'plt' when we plot
```

- These packages are pre-installed and ready to use in Jupyter hub

Plotting

- matplotlib lets you make a plot from data

```
    plt.plot(x_array, y_array) #creates a
plot of x-array and y-array
    plt.show() #displays the plot created on
   the previous line called plt
```

- Use this code to make changes to your plot before plt.show()

```
        plt.title('Plot Title')
#Title, xlabel and ylabel create labels
        plt.xlabel('X-axis label')  #that
are added to your plots
        plt.ylabel('Y-axis label')
```

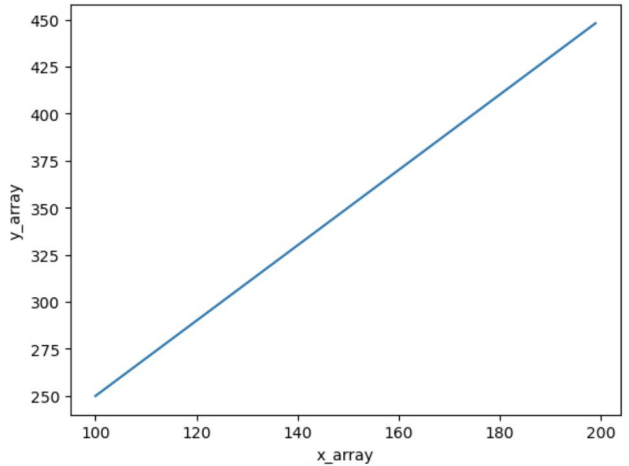
```
[11]: import numpy as np
      import matplotlib.pyplot as plt

[16]: x_array = np.arange(100, 200,1)
      y_array = 2*x_array + 50

[17]: plt.figure()

      plt.plot(x_array, y_array)
      plt.ylabel('y_array')
      plt.xlabel('x_array')

[17]: Text(0.5, 0, 'x_array')
```



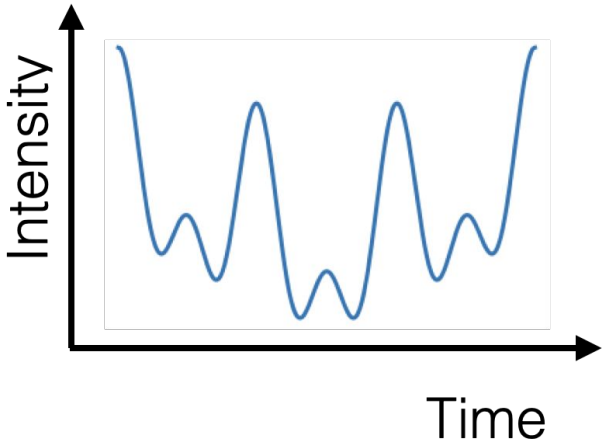
Radio Astronomy Bootcamp
March 2023
Day 3

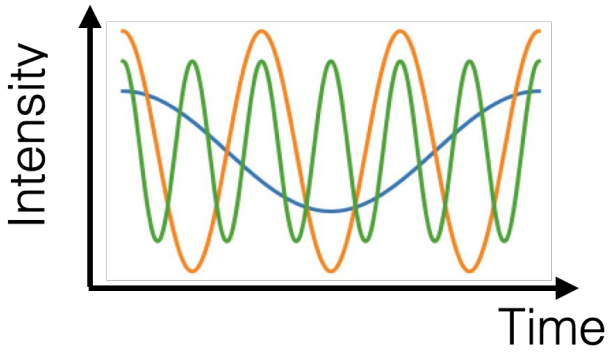
Introduction to Fourier Analysis

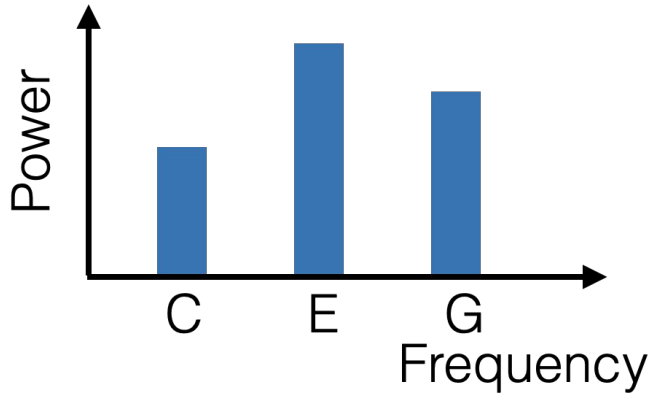
Contributions from: Amy Zhao, Akshatha Vydula, Stephen Murray, HERA Boot Camp materials

Open Jupyter Hub: /data/ra_bootcamp/Day2_FourierAnalysis.ipynb
Change kernel: chart_bootcamp

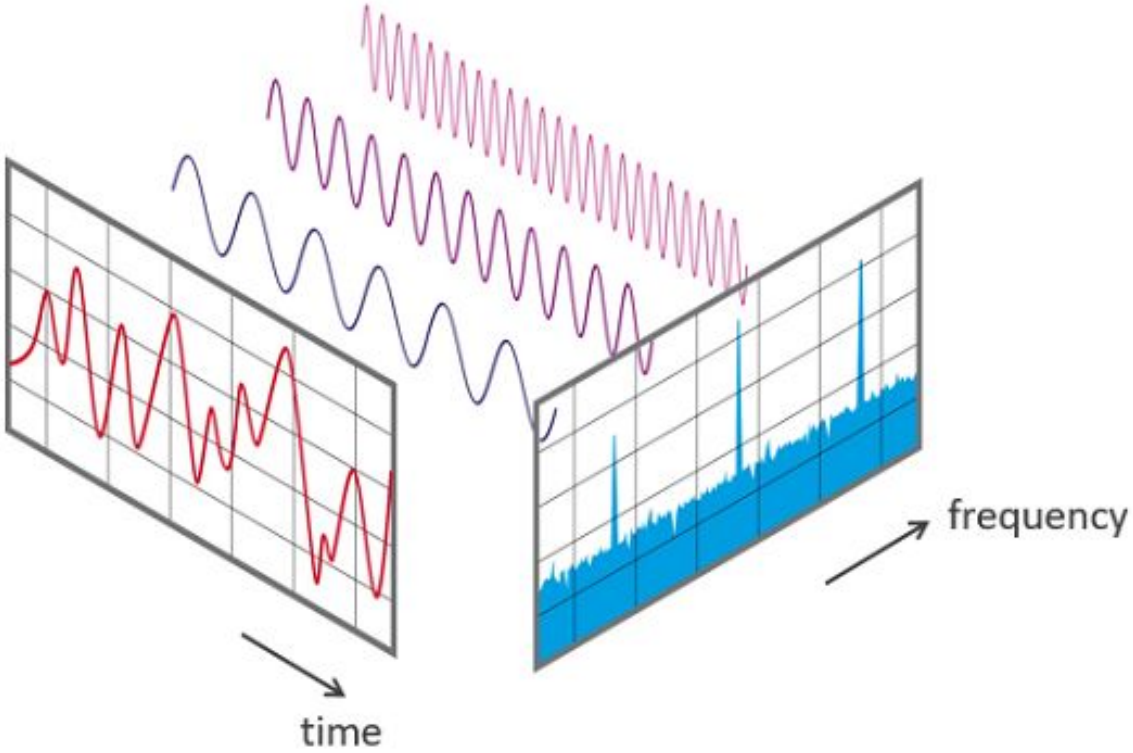
Fourier Transform







Fourier Transform to Spectra



https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft

Fourier Series

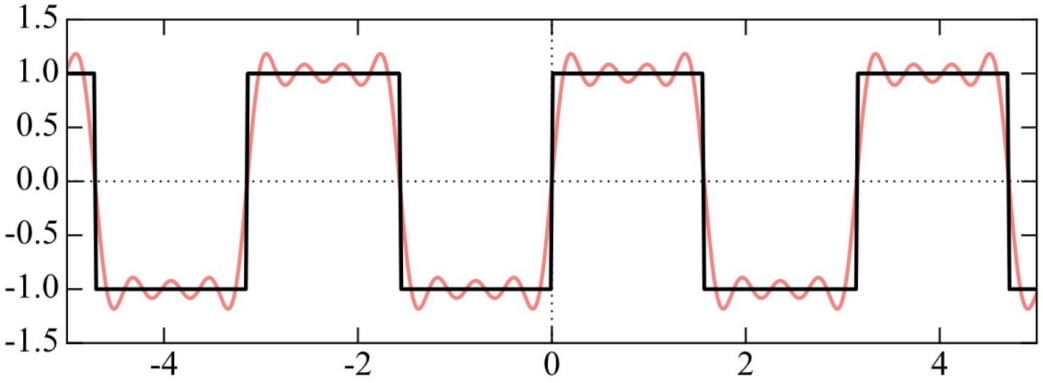
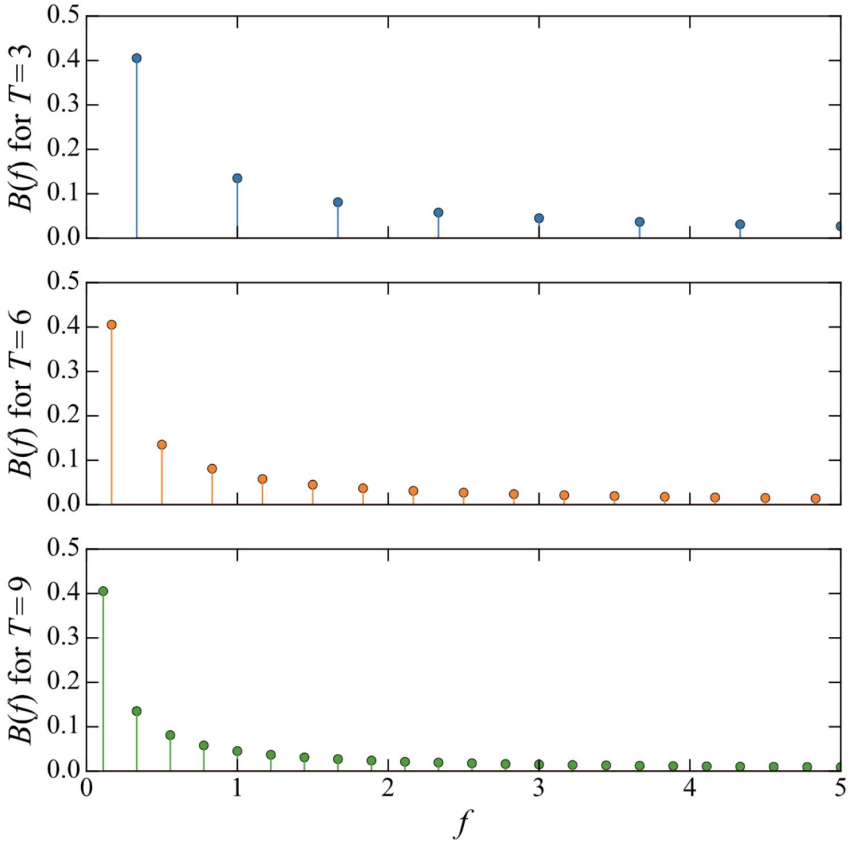


Figure 1: A square wave (black) approximated by a sum of cosines (red).

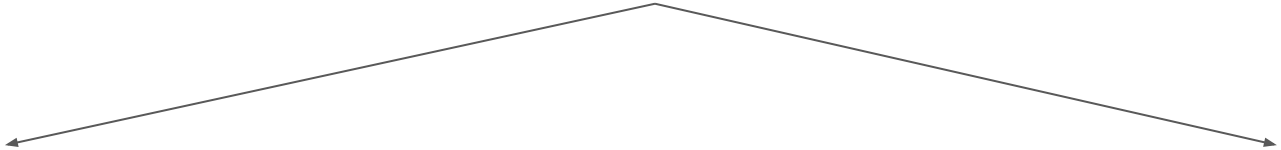
Let's build some intuition for the Fourier Series:

https://phet.colorado.edu/sims/html/fourier-making-waves/latest/fourier-making-waves_en.html

Fourier Transform



Fourier Transform



Continuous Fourier Transform (CFT)

- For continuous functions
- Mathematical construct
- Used in modeling (no real world applications)

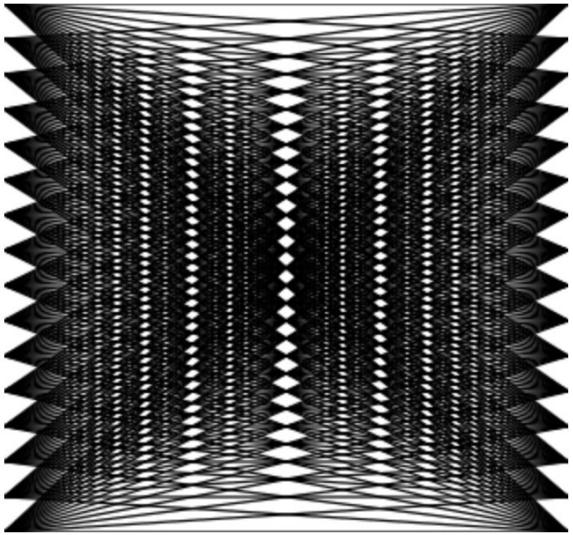
Discrete Fourier transform (DFT)

- For discrete functions
- Experimental data are samples – DFT
- In computers – int/float values

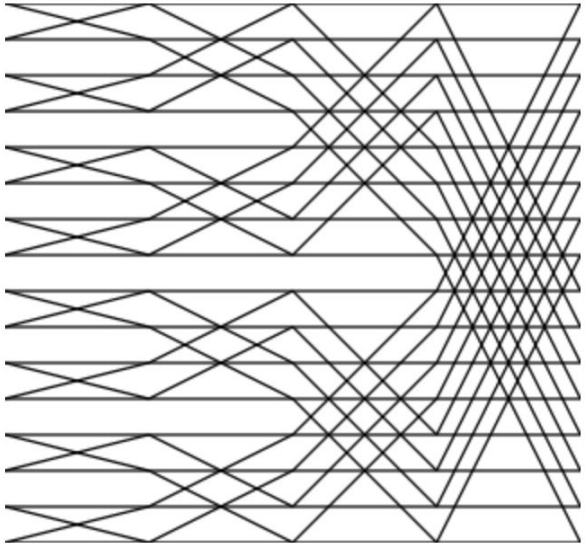
Fast Fourier Transform

- Efficient technique to calculate DFT
- Computations are faster compared to DFT definition
- “Divide and Conquer” → compute DFT of smaller chunks of data and combine them ⇒ lower number of computations
- Most commonly used numerical tool to compute DFT of real signals

DFT, size 16

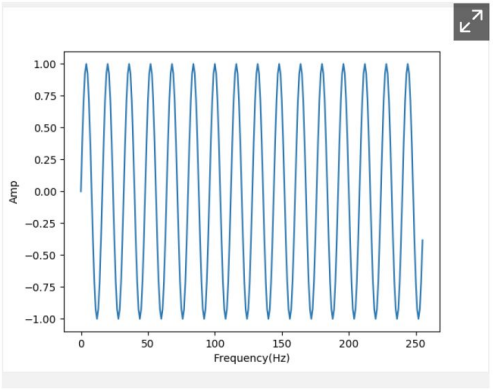


FFT, size 16

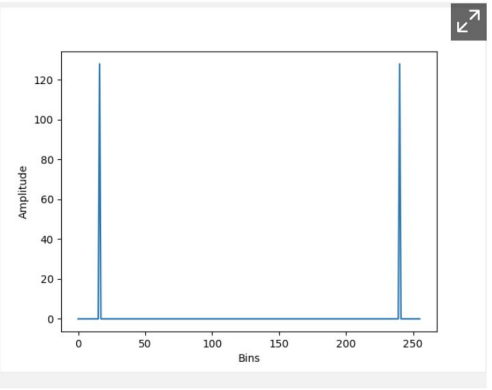


Source: Stack Exchange

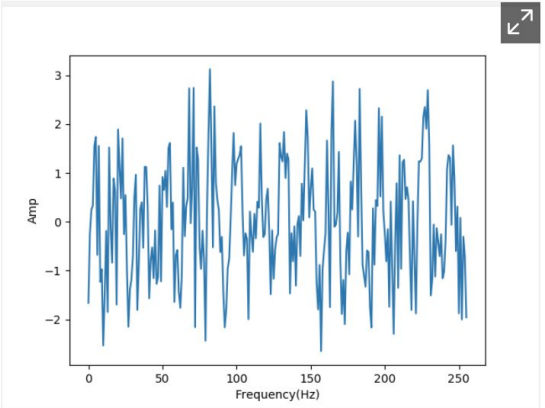
Fourier Transform (with and without noise)



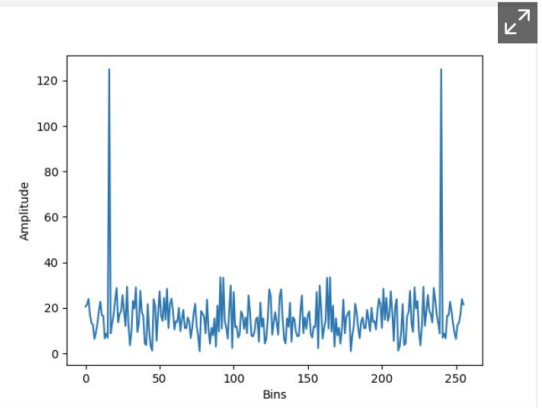
a) Pure Sine wave



b) FFT of Pure Sine wave



a) Noisy Sine wave



b) FFT of Noisy Sine wave