Python Tutorial

SHRIKANT MEHRE

Windows Installation Procedure

Download the anaconda installer

http://energy.iitkgp.ac.in/node/27/ [Accessed on 18 January 2018]

https://www.anaconda.com/download/

Windows Installation Procedure

- Double click the installer to launch
- Click Next
- Read the licensing terms and click I Agree
- Select an install for “Just Me” unless you’re installing for all users and click Next
- Select a destination folder to install Anaconda and click Next
- Choose whether to add Anaconda to your PATH environment variable
- Choose whether to register Anaconda as your default Python 3.6

Windows Installation Procedure

- Click Install
- Click Next
- Click Finish

- From your Windows Start menu, search for ‘Spyder’ and click on the Spyder Icon

Other IDEs:

PyCharm

Sublime Text

Mathematical Operations

```
>>> 2 + 2
4
>>> 50 - 5*6
20
>>> (50 - 5.0*6) / 4
5.0
>>> 8 / 5.0
1.6
```

Mathematical Operations

```
>>> 17 / 3  # int / int -> int
5
>>> 17 / 3.0  # int / float -> float
5.666666666666667
>>> 17 // 3.0  # explicit floor division discards the fractional part
5.0
>>> 17 % 3  # the % operator returns the remainder of the division
2
```

Lists

```
>>> squares = [1, 4, 9, 16, 25]
>>> squares
[1, 4, 9, 16, 25]


>>> squares[0]  # indexing returns the item
1
>>> squares[-1]
25
>>> squares[-3:]  # slicing returns a new list
[9, 16, 25]
```

```
>>> squares[:]
[1, 4, 9, 16, 25]
>>> squares + [36, 49, 64, 81, 100]
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

Lists

```
>>> cubes = [1, 8, 27, 65, 125]
>>> cubes[3] = 64  # replace the wrong value
>>> cubes
[1, 8, 27, 64, 125]
>>> cubes.append(216)  # add the cube of 6
>>> cubes.append(7 ** 3)  # and the cube of 7
>>> cubes
[1, 8, 27, 64, 125, 216, 343]
```

The del statement

```
>>> a = [-1, 1, 66.25, 333, 333, 1234.5]
>>> del a[0]
>>> a
[1, 66.25, 333, 333, 1234.5]
>>> del a[2:4]
>>> a
[1, 66.25, 1234.5]
>>> del a[:]
>>> a
[]
```

Tuples

```
>>> t = 12345, 54321, 'hello!'
>>> t[0]
12345
>>> t
(12345, 54321, 'hello!')
>>> # Tuples are immutable:
... t[0] = 88888
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
```

Sets

```
>>> basket = ['apple', 'orange', 'apple', 'pear', 'orange', 'banana']
>>> fruit = set(basket)               # create a set without duplicates
>>> fruit
set(['orange', 'pear', 'apple', 'banana'])
>>> 'orange' in fruit                 # fast membership testing
True
>>> 'crabgrass' in fruit
False
```

Dictionaries

```
>>> tel = {'jack': 4098, 'sape': 4139}
>>> tel['guido'] = 4127
>>> tel
{'sape': 4139, 'guido': 4127, 'jack': 4098}
>>> tel['jack']
4098
```

Control Flow Tools – *if* statements

```
>>> x = int(raw_input("Please enter an integer: "))
Please enter an integer: 5
>>> if x<0:
...     print('Negative')
... elif x>0:
...     print('Positive')
... else:
...     print('Zero')
...
Positive
>>>
```

Control Flow Tools – *while* statements

```
>>> b=1
>>> while b<5:
...     print b
...     b=b+1
...
1
2
3
4
>>>
```

Control Flow Tools – *for* statements

```
>>> # Measure some strings:
... words = ['cat', 'window', 'defenestrate']
>>> for w in words:
...     print w, len(w)
...
cat 3
window 6
defenestrate 12
```

Defining Functions

```
>>> def fib(n):    # write Fibonacci series up to n
...         a, b = 0, 1
...         while a < n:
...         print a,
...         a, b = b, a+b
...
>>> # Now call the function we just defined:
... fib(2000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597
```

numpy

```
>>> import numpy as np
>>> a = np.array([2,3,4])
>>> a
array([2, 3, 4])
>>> a.dtype
dtype('int64')
>>> b = np.array([1.2, 3.5, 5.1])
>>> b.dtype
dtype('float64')
```

numpy

```
>>> np.zeros( (3,4) )
array([[ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.]])
```

Factorial Code

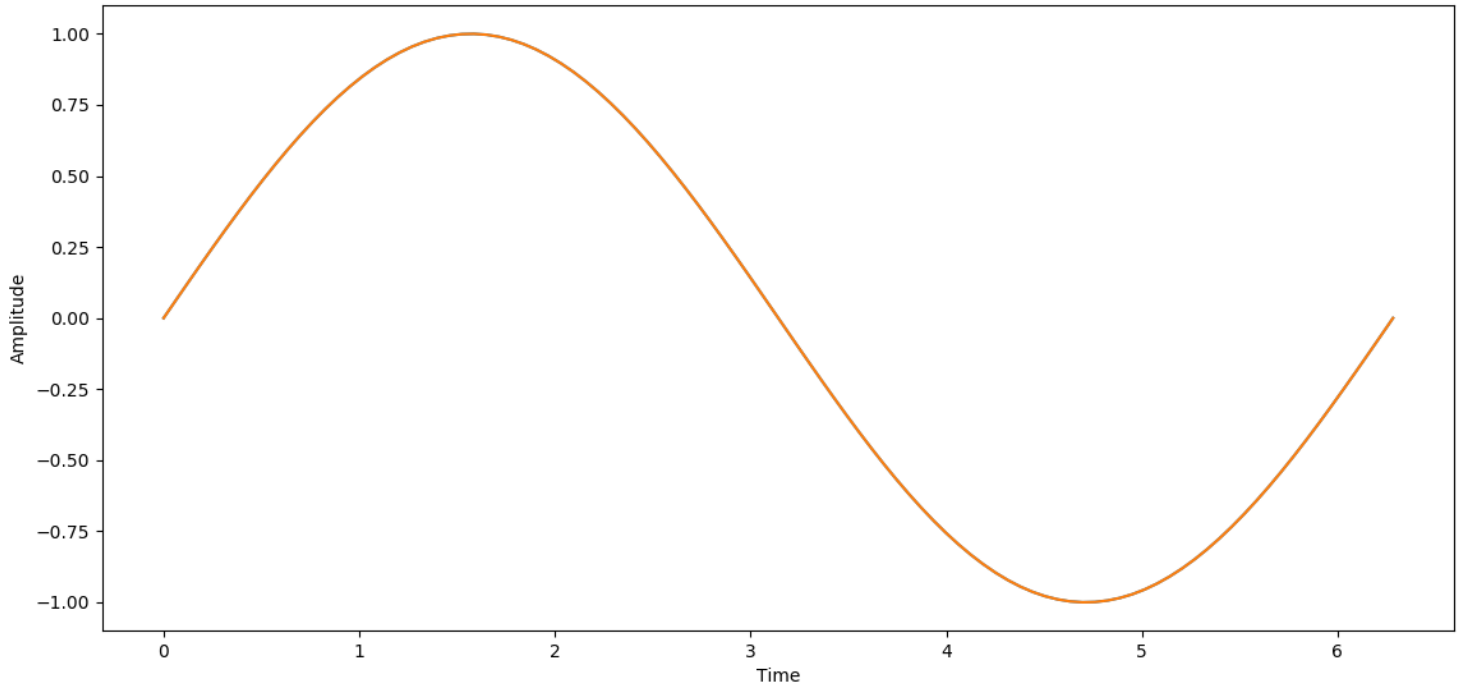
```
import numpy as np


def factorial(inpNumber):
    fact = np.array([1])
    for x in range(1,inpNumber+1):
        fact = fact*x
    return fact
```

```
inpNumber = input("Enter the number")


fact = factorial(inpNumber)

print(fact)
```

```
>>> runfile('C:/Users/SMK/D03P04.py', wdir='C:/Users/SMK')
Enter the number 5
[120]
>>>
```

Plotting a Sine wave

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


x = np.linspace( 0, 2*pi, 100 );
y = np.sin(x);
plt.plot(x,y)
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
plt.savefig('sineWave.png')
```



Loading data files

```
import scipy.io as sio


# Loading neural network's output and target
neuralOut = sio.loadmat('neuralOut.mat')
neuralOut = neuralOut['neuralOut']
```

Convolution

```
import numpy as np

from scipy import signal


x = np.array([1.0, 2.0, 3.0])

h = np.array([4.0, 5.0, 6.0])

y = signal.convolve(x, h)

print(y)

z = signal.convolve(x, h, 'same')

print(z)
```

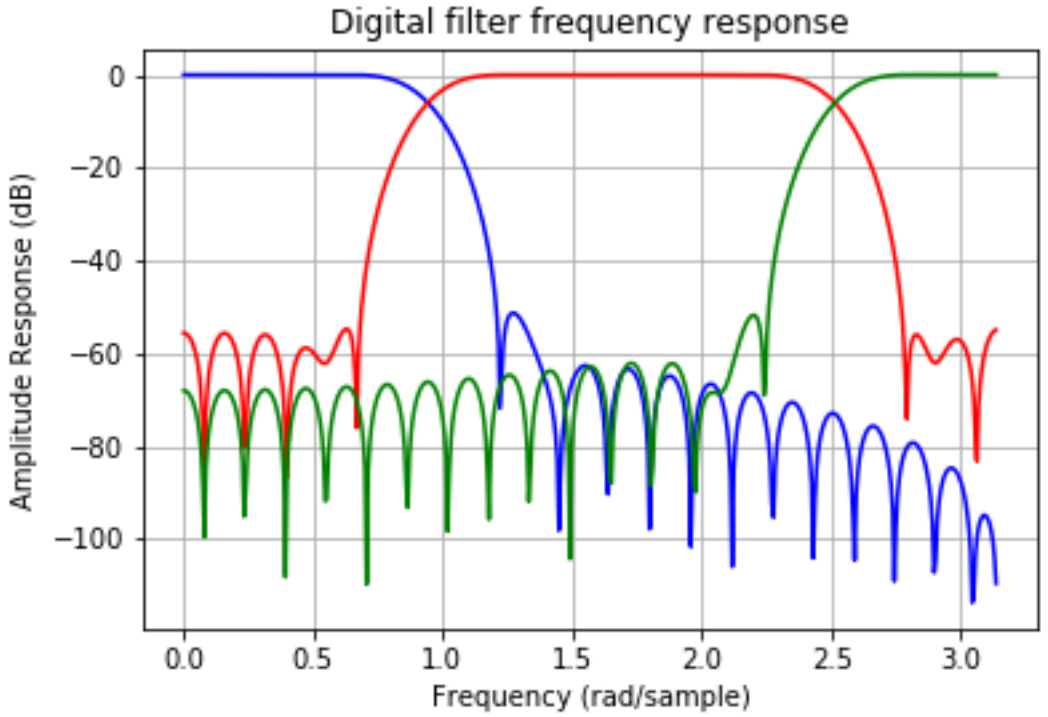
```
[ 4.  13.  28.  27.  18.]

[ 13.  28.  27.]
```

Filter Design

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


b1 = signal.firwin(40, 0.3)
b2 = signal.firwin(41, [0.3, 0.8],pass_zero=False)
b3 = signal.firwin(41, 0.8, pass_zero=False)
w1, h1 = signal.freqz(b1)
w2, h2 = signal.freqz(b2)
w3, h3 = signal.freqz(b3)
```

```
plt.title('Digital filter frequency response')
plt.plot(w1, 20*np.log10(np.abs(h1)), 'b')
plt.plot(w2, 20*np.log10(np.abs(h2)), 'r')
plt.plot(w3, 20*np.log10(np.abs(h3)), 'g')
plt.ylabel('Amplitude Response (dB)')
plt.xlabel('Frequency (rad/sample)')
plt.grid()
#plt.show()
plt.savefig('FilterResponse.png')
```

Filter Response

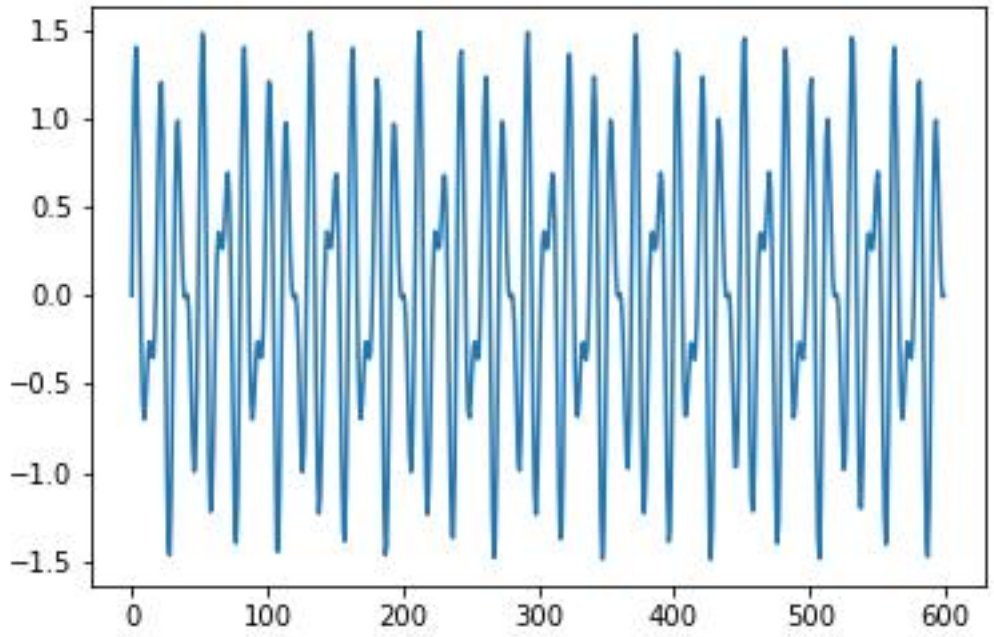


FFT

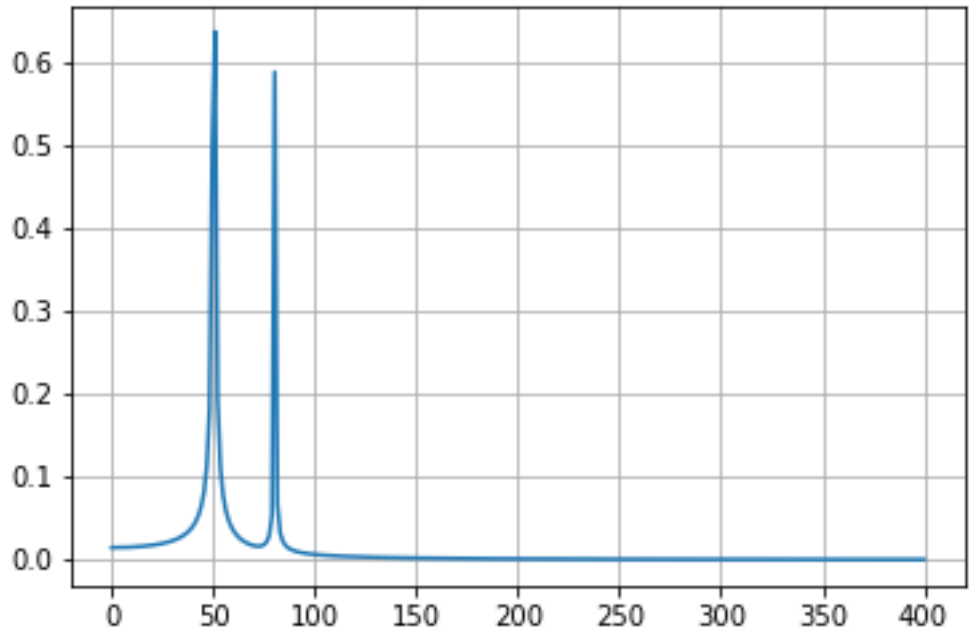
```
import numpy as np
from scipy.fftpack import fft
import matplotlib.pyplot as plt

N = 600
T = 1.0 / 800.0
x = np.linspace(0.0, N*T, N)
y = 0.9*np.sin(50.0 * 2.0*np.pi*x) + 0.6*np.sin(80.0 * 2.0*np.pi*x)
plt.figure(1)
plt.plot(y)
plt.savefig('Input Wave.png')
yf = fft(y)
xf = np.linspace(0.0, 1.0/(2.0*T), N//2)
plt.figure(2)
plt.plot(xf, 2.0/N * np.abs(yf[0:N//2]))
plt.grid()
plt.show()
plt.savefig('Frequency Response.png')
```

FFT

Input Signal



Frequency Response



FFT, IFFT

```
import numpy as np
from scipy.fftpack import fft, ifft

x = np.array([1.0, 1.0, 1.0, 1.0, 1.0])
y = fft(x)
print(y)
yinv = ifft(y)
print(yinv)
```

```
[ 5.+0.j  0.+0.j  0.+0.j  0.-0.j  0.-0.j]

[ 1.+0.j  1.+0.j  1.+0.j  1.+0.j  1.+0.j]
```