Convolution of two functions
and
Python code to evaluate convolution of two functions

Shyamal Bhar
Assistant Professor
Vidyasagar College for Women
Kolkata 700 006

<u>Convolution:</u> The convolution between two functions $f(x)$ and $g(x)$ (in one-dimension) is defined as

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(x') g(x - x') \, dx'$$

$$= \int_{0}^{x} f(x') g(x - x') \, dx'$$

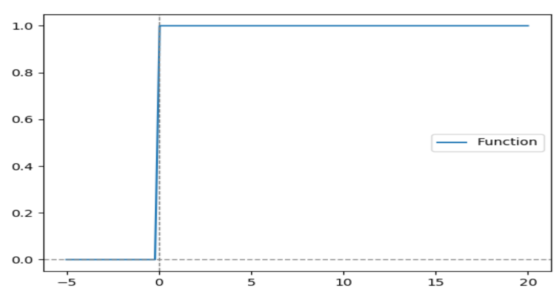
where x' is a dummy variable of integration. <u>The convolution of these two functions evaluates the area of overlap between the function $f(x)$ and shifted spatially reversed function of the function $g(x)$.</u>

Graphical Example:

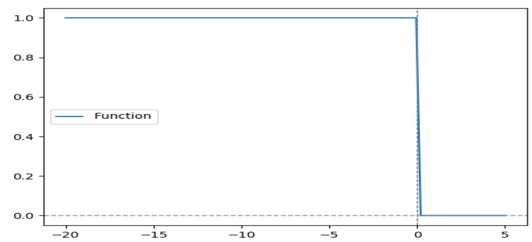
Let us consider the convolution of two **unit step function**. Unit function is defined as

$$u(x) = \begin{cases} 1 & \text{for } x>0 \\ 0 & \text{otherwise} \end{cases}$$

Sketch of $u(x')$

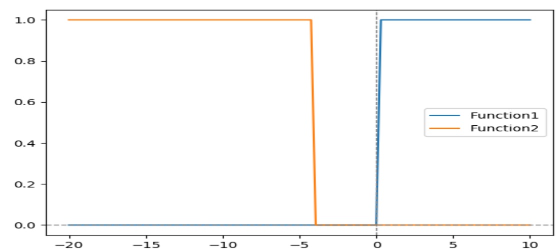


Sketch of $u(-x')$



Sketch of $u(x-x')$:

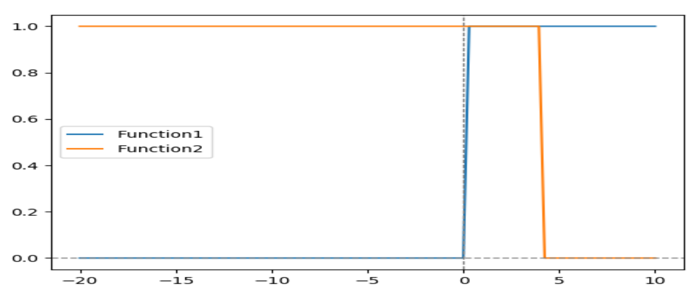
Case-1 (when $x<0$)



$$\text{Now, } u(x')u(x-x') = 0$$

$$\text{so, } f*g = \int_{-\infty}^{\infty} u(x')u(x-x')dx' = 0$$

Case-2 (when $x>=0$)



$$\text{Now, } u(x')u(x-x') = \begin{cases} 1 & \text{for } 0 <= x' <= x \\ 0 & \text{otherwise} \end{cases}$$

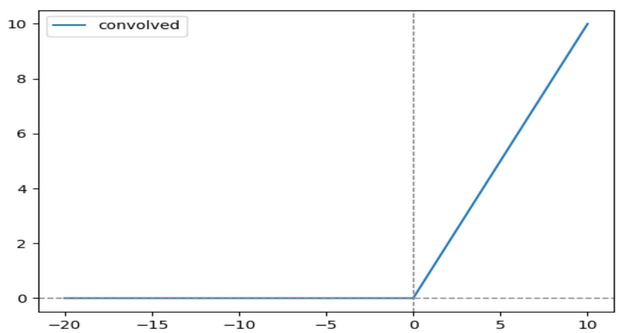
$$\text{so, } f*g = \int_{-\infty}^{\infty} u(x')u(x-x')\,dx'$$

$$= \int_0^x dx'$$

$$= x$$

Combining two cases:

$$\text{so, } f*g = \begin{cases} x & \text{for } x >= 0 \\ 0 & \text{for } x < 0 \end{cases}$$



Python code:

Example-1

Convolution of two unit functions:

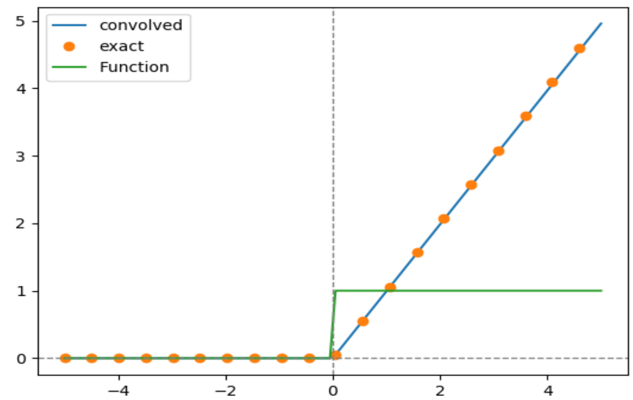
Unit function is defined as

$$u(x) = \begin{cases} 1 & \text{for } x>0 \\ 0 & \text{otherwise} \end{cases}$$

Code:

```
# Convolution of two unit function
# u(x)=1 for x>0 else 0
# fg=Integration(f(xp)*g(x-xp)dxp) with limit -inf to inf
OR 0 to x
```

```
import numpy as np
from scipy.integrate import quad,simps
import matplotlib.pyplot as plt

# Range of X
a,b=-5,5

# Definition of functions

x=np.linspace(a,b,100)
u=lambda x: 1.0 if x>0.0 else 0.0
u=np.vectorize(u)


# Calculation of convolution integration

conv=[]
for xx in x:
         xp=np.linspace(0,xx,100)
         h=u(xp)*u(xx-xp)
         conv.append(simps(h,xp))

# Exact result
exact=lambda x: x if x>0.0 else 0.0
exact=np.vectorize(exact)

# Plotting

plt.plot(x,conv,label='convolved')
plt.plot(x[::5],exact(x)[::5],'o',label='exact')
plt.plot(x,u(x),label='Function')
plt.axvline(0, c='gray',ls='--', lw=1)
plt.axhline(0, c='gray',ls='--', lw=1)
plt.legend(loc='best',prop={"size":10})
plt.savefig("con_unit.png")
plt.show()
```



Example 2

Convolution of two Gaussian functions

Gaussian function is defined as

$$f(x)=\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

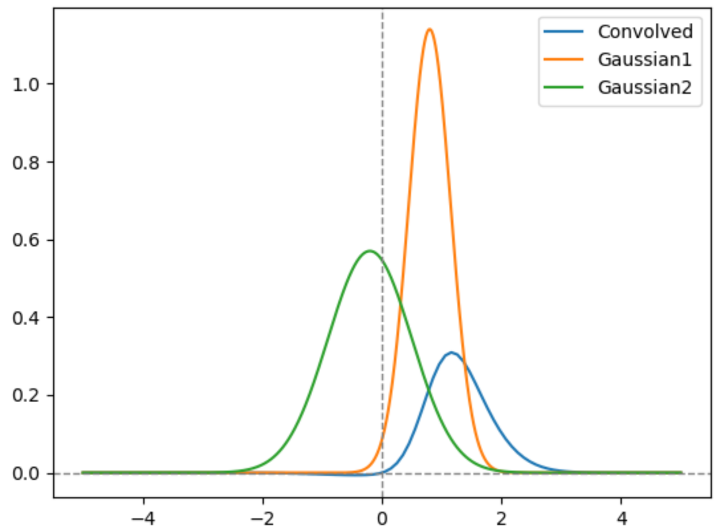
where σ and μ are the standard deviation and mean of the Gaussian function.

We shall show that the convolution of two Gaussian functions is a Gaussian function.

Code:

```
# Convolution of two Gaussian function
#f(x)=1.0/(sig*np.sqrt(2*np.pi))*np.exp(-1*((x-
mu)**2)/(2*sig**2))
# fg=Integration(f(xp)*g(x-xp)dxp) with limit -inf to inf

import numpy as np
from scipy.integrate import quad,simps
import matplotlib.pyplot as plt

# Parameter for two Gaussian Function
p1=[0.35,0.8] # For 1st Gaussian function
p2=[0.7,-0.2] # For 2nd Gaussian function
```

```
# Range of X
a,b=-5,5

# Definition of Gaussian function

def g(x,p):
        sig=p[0] # p1[0]=sigma(std deviation) p1[1]=mean
        mu=p[1]
        return 1.0/(sig*np.sqrt(2*np.pi))*np.exp(-1*((x-
mu)**2)/(2*sig**2))

# Calculation of convolution integration
X=np.linspace(a,b,100)
conv=[]
for x in X:
        xp=np.linspace(0,x,100)
        h=g(xp,p1)*g(x-xp,p2)
        conv.append(simps(h,xp))

# plotting

plt.plot(X,conv,label='Convolved')

xr=np.linspace(a,b,1000)
plt.plot(xr,g(xr,p1),label='Gaussian1')
plt.plot(xr,g(xr,p2),label='Gaussian2')
plt.axvline(0, c='gray',ls='--', lw=1)
plt.axhline(0, c='gray',ls='--', lw=1)
plt.legend(loc='best',prop={"size":10})
plt.savefig("gaussian.png")
plt.show()
```



Example -3

Convolution of two functions

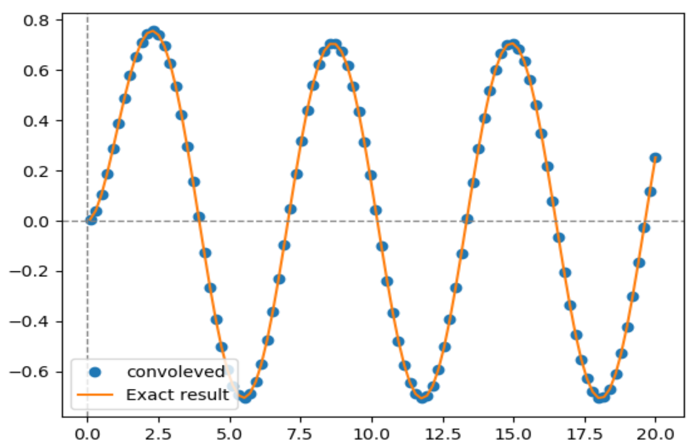
$$f(x) = e^{-x}$$
$$g(x) = \sin(x)$$

$$f * g(x) = \frac{\left(e^{-x} + \sin(x) - \cos(x)\right)}{2}$$

Code:

```
# Convolution of two  function
# f(x)=exp(-x), g(x)=sin(x)
#Exact result: f*g(x)=0.5*(exp(-x)+sin(x)-cos(x))
# fg=Integration(f(xp)*g(x-xp)dxp) with limit -inf to inf

import numpy as np
from scipy.integrate import quad,simps
import matplotlib.pyplot as plt

# Range of X
a,b=0.1,20

# Definition of functions
```

```
f=lambda x: np.exp(-x)
g=lambda x: np.sin(x)


# Calculation of convolution integration
X=np.linspace(a,b,100)
conv=[]
for x in X:
         xp=np.linspace(0,x,100)
         h=f(xp)*g(x-xp)
         conv.append(simps(h,xp))

exact=0.5*(np.exp(-X)+np.sin(X)-np.cos(X))
# Plotting

plt.plot(X,conv,'o',label='convoleved')
plt.plot(X,exact,'-',label='Exact result')
plt.axvline(0, c='gray',ls='--', lw=1)
plt.axhline(0, c='gray',ls='--', lw=1)
plt.legend(loc='best',prop={"size":10})
plt.savefig("con3.png")
plt.show()
```



Annexure:

Calculation of convolution of two functions $f(x) = e^{-x}$, $g(x) = \sin(x)$:

By definition $f * g(x) = \int_0^t e^{-x'} \sin(x - x')dx'$

Integrating by parts twice we get,

$$\int_0^t e^{-x'} \sin(x')dx' =$$

$$\left[e^{-x'}\cos(x-x')\right]_0^x - \left[e^{-x'}\sin(x-x')\right]_0^x - \int_0^x\int_0^t e^{-x'} \sin(x')dx'$$

$$\therefore 2\int_0^t e^{-x'} \sin(x')dx' = e^{-x} - \cos(x) + \sin(x)$$

$$\therefore f * g(x) = \frac{1}{2}\left[e^{-x} + \sin(x) - \cos(x)\right]$$