**Experiment No. 05**

### **Name: Rebecca Dias Roll No.: 19 Class: BE CMPN A PID: 182027**

**Aim:** To implement linear, circular, and linear via circular convolution in order to be useful in a real-world application.

# **Problem Definition:**

1. Find Linear Convolution and Circular Convolution of L point sequence x[n] and M point sequence h[n].
2. Find Linear Convolution of L point sequence x[n] and M point sequence h[n] using Circular convolution.
3. Give your conclusion about No of values in linearly convolved signal, and the Aliasing effect in Circular Convolution.

# **Theory:**

In [mathematics](https://en.wikipedia.org/wiki/Mathematics) (and, in particular, [functional analysis](https://en.wikipedia.org/wiki/Functional_analysis)) convolution is a [mathematical operation](https://en.wikipedia.org/wiki/Operation_(mathematics)) on two [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) (*f* and *g*); it produces a third function, that is typically viewed as a modified version of one of the original functions, giving the integral of the [pointwise](https://en.wikipedia.org/wiki/Pointwise) multiplication of the two functions as a function of the amount that one of the original functions is [translated](https://en.wikipedia.org/wiki/Translation_(geometry)). Convolution is similar to [cross-correlation](https://en.wikipedia.org/wiki/Cross-correlation). It has applications that include [probability](https://en.wikipedia.org/wiki/Probability), [statistics](https://en.wikipedia.org/wiki/Statistics), [computer vision](https://en.wikipedia.org/wiki/Computer_vision), [natural language](https://en.wikipedia.org/wiki/Natural_language_processing) [processing](https://en.wikipedia.org/wiki/Natural_language_processing), [image](https://en.wikipedia.org/wiki/Image_processing) and [signal processing](https://en.wikipedia.org/wiki/Signal_processing), [engineering](https://en.wikipedia.org/wiki/Engineering), and [differential equations](https://en.wikipedia.org/wiki/Differential_equations).

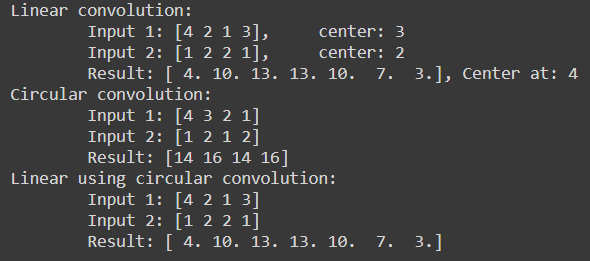
* + Linear convolution is the basic operation to calculate the output for any linear time-invariant system given its input and its impulse response.
  + Circular convolution is the same thing but considering that the support of the signal is periodic (as in a circle, hence the name).

Convolution could be used to calculate the response of an LTI system, and (normalized) cross-correlation could be used for pattern matching: the maxima of the cross-correlation function are *at the offset* where pattern g is most likely to be situated in the signal f. If you know this offset you could use a similarity measure (such as the Euclidean distance) to quantify similarity.

**Code:**

|  |
| --- |
| import numpy as np from scipy.linalg import circulant  def lin\_conv(x,h,c1,c2):  m=x\*h  dim=m.shape[0]+h.shape[0]-1  res=np.empty((dim,dim))  res.fill(0)  j=m.shape[0]  for i in range(m.shape[1]):  res[i:i+j,i]=m[:,i]  lc1=0  lc2=0  while (c1!=1):  lc1-=1  c1-=1  while (c2!=1):  lc2-=1  c2-=1  c=lc1+lc2  r=1;  while (c!=0):  r+=1  c+=1  return np.sum(res,axis=1),r  x=np.array([[4,2,1,3]]) h=np.array([[1],[2],[2],[1]]) c1=3 c2=2 res=lin\_conv(x,h,c1,c2) x=x.reshape(-1) h=h.reshape(-1) print(f"Linear convolution:\n\tInput 1: {x},\tcenter: {c1}\n\tInput 2: {h},\tcenter: {c2}\n\tResult: {res[0]}, Center at: {res[1]}")  def cir\_conv(x,h):  if (x.shape[0]!=h.shape[0]):  m=max(x.shape[0],h.shape[0])  else:  m=0  if (m!=0 and x.shape[0]!=m):  x=np.append(x,np.zeros((m-x.shape[0],)))  elif (m!=0):  h=np.append(h,np.zeros((m-h.shape[0],)))  c=circulant(x)  return c@h  x=np.array([4,3,2,1]) h=np.array([1,2,1,2]) res=cir\_conv(x,h) print(f"Circular convolution:\n\tInput 1: {x}\n\tInput 2: {h}\n\tResult: {res}")  def lin\_cir\_conv(x,h):  m=x.shape[0]  n=h.shape[0]  x=np.append(x,np.zeros((n-1,)))  h=np.append(h,np.zeros((m-1,)))  i1=circulant(x)  return i1@h  x=np.array([4,2,1,3]) h=np.array([1,2,2,1]) res=lin\_cir\_conv(x,h) print(f"Linear using circular convolution:\n\tInput 1: {x}\n\tInput 2: {h}\n\tResult: {res}") |

**Output:**



# **Conclusion:**

The convolution operation is given by the integral over the product of two functions, where one function is flipped and shifted in time. The convolution operation smoothes the input signals, i.e., the output of the convolution is a smoother function than its input functions. Convolution has applications that include probability, statistics, acoustics, spectroscopy, signal processing, and image processing, engineering, physics, computer vision, and differential equations. The program for linear convolution, circular convolution, and linear using circular convolution was implemented in python and the output was obtained successfully.