

| **Title:** Write a program to Compute linear and circular convolution of two discrete time signal sequences using Matlab. |
| --- |

**Objective:** To familiarize the beginnerto MATLAB by introducing the basic features and commands of the program.

**Expected Outcome of Experiment:**

| **CO** | **Outcome** |
| --- | --- |
| **CO3** | To understand the concept of convolution and perform different convolution operations on the given input signals. |

**Books/ Journals/ Websites referred:**

1. http://www.mathworks.com/support/
2. www.math.mtu.edu/~msgocken/intro/intro.html
3. www.mccormick.northwestern.edu/docs/efirst/matlab.pdf
4. A.Nagoor Kani “Digital Signal Processing”, 2nd Edition, TMH Education.

**Pre Lab/ Prior Concepts:**

**Convolution**

Discrete time convolution is a method of finding response of linear time invariant system. It is based on the concepts of linearity and time invariance and assumes that the system information

is known in terms of its impulse response h[n].

Convolution is defined as

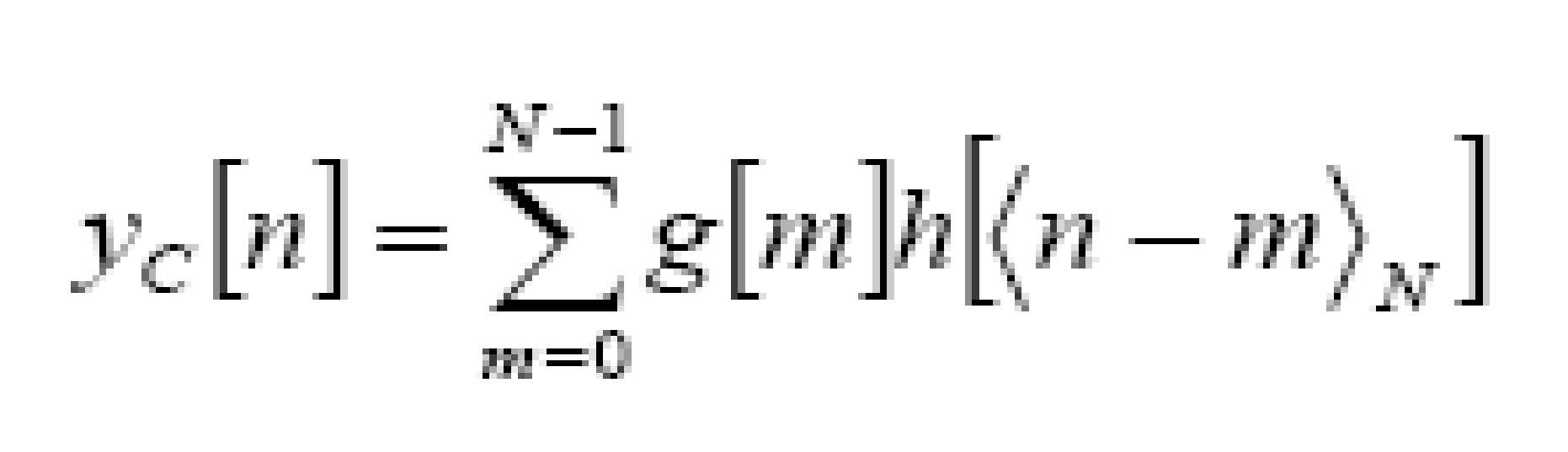
∞

Y[n] = Σ h[k]x [n-k] =h[n]\*x[n] k=-∞

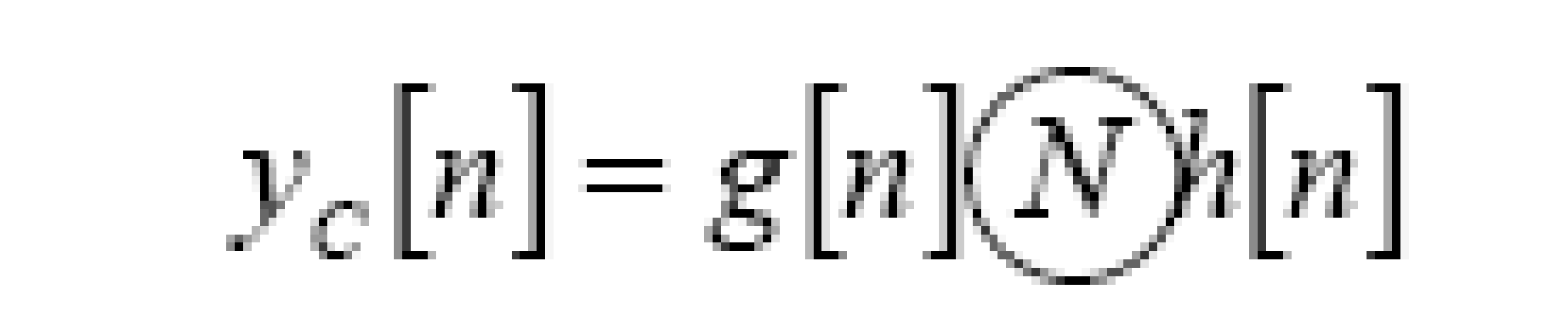
Convolution consists of folding, shifting, Multiplication and summation operations.

**Circular Convolution**

Circular convolution between two length N sequences can be carried out as shown by the expression below:

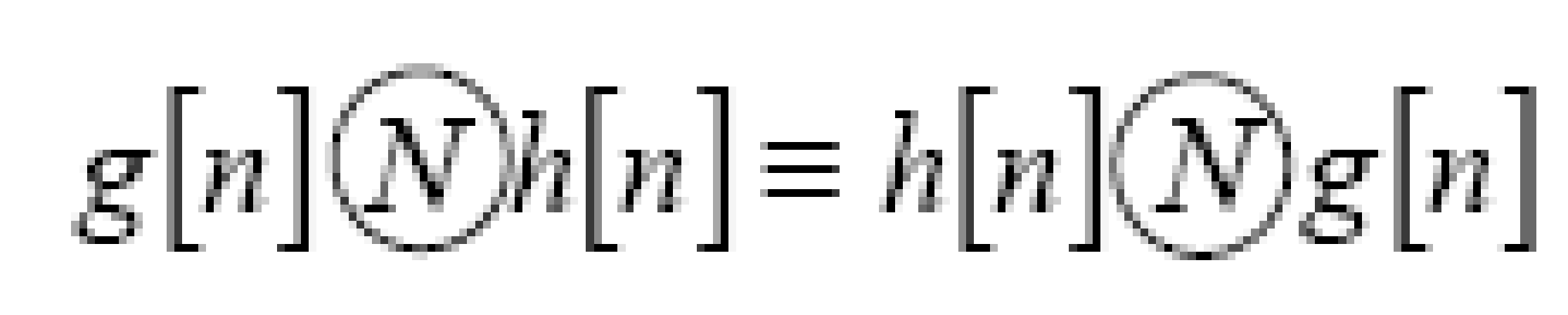


Since the above operation involves two length-N sequences it is referred to as the N-point circular convolution and denoted by:

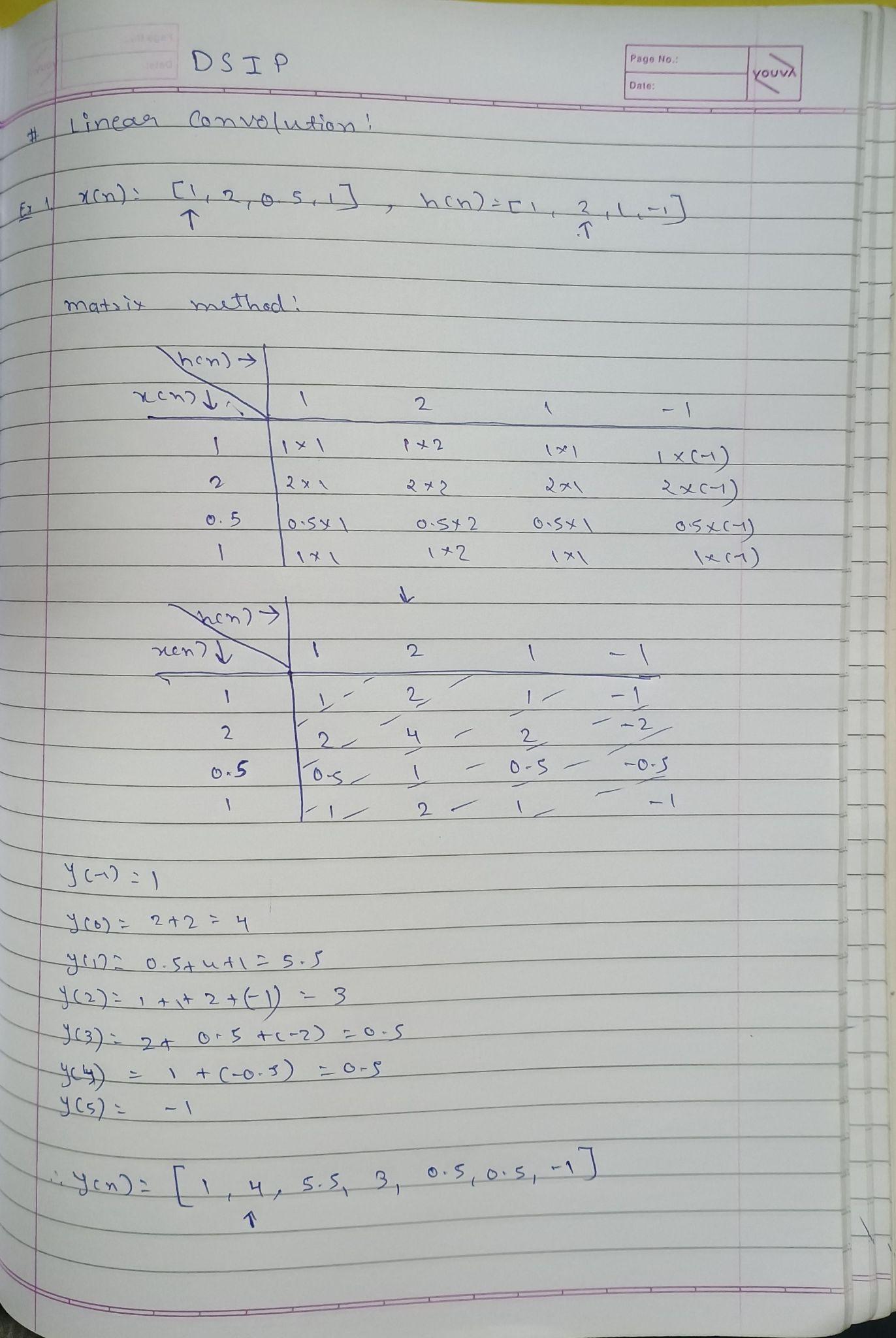


As in linear convolution circular convolution is commutative.

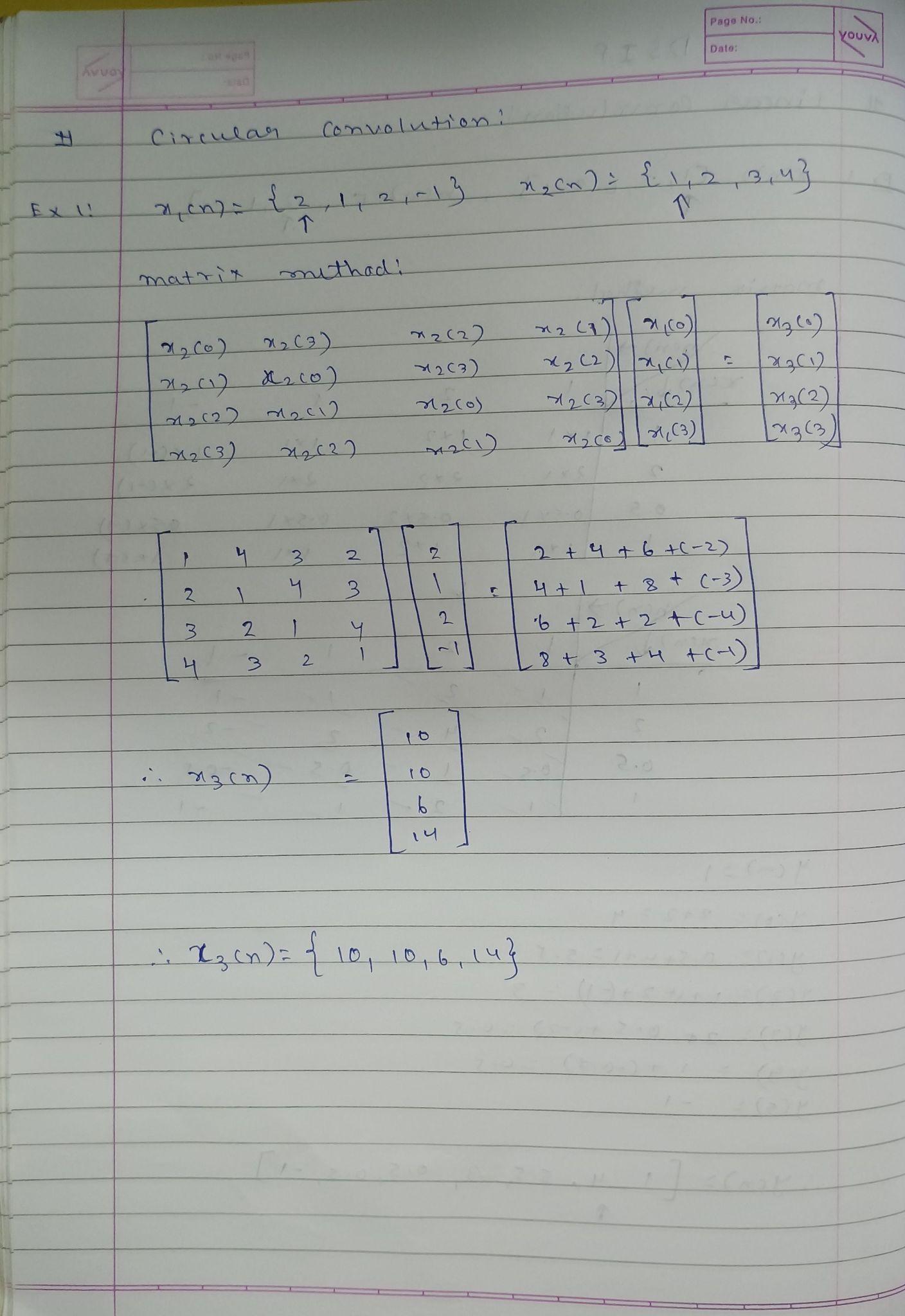
i.e.



**Example Of Linear Convolution:**

****

**Example Of Circular Convolution:**

****

**Implementation details along with screenshots:**

**1] Linear Convolution:**

**x = [1, 2, 3];**

**h = [0, 1, 0.5];**

**Lx = length(x);**

**Lh = length(h);**

**Ly = Lx + Lh - 1;**

**y\_linear = zeros(1, Ly);**

**for n = 1:Ly**

**for m = 1:Lh**

**if (n - m + 1) > 0 && (n - m + 1) <= Lx**

**y\_linear(n) = y\_linear(n) + x(n - m + 1) \* h(m);**

**end**

**end**

**end**

**figure;**

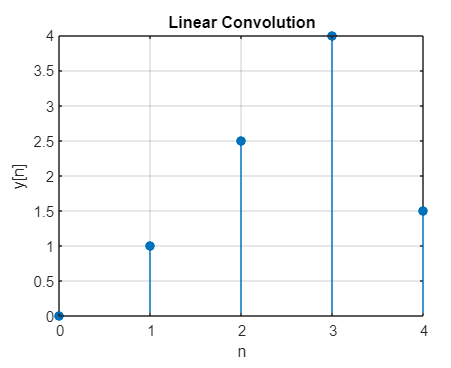
**stem(0:Ly-1, y\_linear, 'filled');**

**title('Linear Convolution');**

**xlabel('n');**

**ylabel('y[n]');**

**grid on;**

**OUTPUT:  
**

**2] Circular Convolution:**

**x = [1, 2, 3];**

**h = [0, 1, 0.5];**

**N = length(x);**

**y\_circular = zeros(1, N);**

**for n = 1:N**

**for m = 1:N**

**y\_circular(n) = y\_circular(n) + x(m) \* h(mod(n - m, N) + 1);**

**end**

**end**

**figure;**

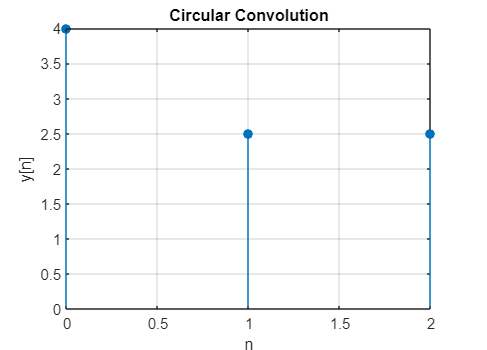
**stem(0:N-1, y\_circular, 'filled');**

**title('Circular Convolution');**

**xlabel('n');**

**ylabel('y[n]');**

**grid on;**

**OUTPUT:**

**Conclusion:-**

Convolution is essential in signal processing, and MATLAB provides an effective platform for implementing and visualizing both types.

**Date: 31 \ 01 \ 2025 Signature of faculty in-charge**

**Post Lab Descriptive Questions**

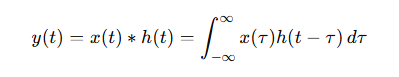
* 1. Explain the role of convolution in signal processing.

Convolution is a mathematical operation used in signal processing to analyze the way a system responds to a given input signal. It describes how the output signal of a system (such as a filter) is related to an input signal.

In signal processing, convolution is typically used to:

* **Apply filters to signals**: For instance, in image processing, convolution can be used to apply filters like blurring or sharpening.
* **Model linear time-invariant (LTI) systems**: Convolution helps to determine the output of an LTI system when the input and system's impulse response are known.
* **Characterize system behavior**: It provides insight into how the system "shapes" or modifies the input signal over time.

Mathematically, convolution is represented as:



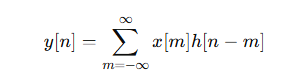
Where:

* y(t) is the output signal,
* x(t) is the input signal,
* h(t) is the system's impulse response,
* \* denotes the convolution operation.
  1. Explain the difference between linear and circular convolution?

**Linear Convolution**:

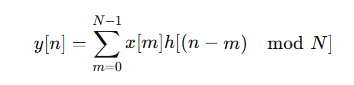
* Linear convolution is the standard form of convolution used in most applications.
* It is performed between two finite-duration signals and the result is generally a signal whose length is the sum of the lengths of the two signals minus one.
* It is used when the signals do not overlap, and it assumes that the signals start at zero and the output extends beyond the duration of the input signals.

Mathematically, for two discrete signals x[n] and h[n], the linear convolution is defined as:



**Circular Convolution**:

* Circular convolution, on the other hand, assumes that the signals are periodic. This means that the signals "wrap around" when the indices exceed their lengths.
* It is often used in the context of processing signals in the frequency domain, particularly when dealing with discrete Fourier transforms (DFT) or fast Fourier transforms (FFT).
* Circular convolution of two sequences x[n] and h[n] of length NNN is defined as:



Key differences:

* Linear convolution does not assume periodicity, whereas circular convolution assumes the signals are periodic and "wrap around".
* Circular convolution is often used in applications involving DFT/FFT, where signals are periodic in nature due to the finite length of the input.
  1. Explain with the help of an example the steps required to transform linear convolution with circular convolution and vice-versa.

**Transforming Linear Convolution to Circular Convolution**: To transform linear convolution into circular convolution, we typically use the following steps:

1. **Zero-padding**: Extend both signals to the same length by padding them with zeros. The new length is usually the sum of the lengths of the two sequences minus one, i.e., N=Lx+Lh−1, where Lx​ is the length of x[n] and Lh​ is the length of h[n].
2. **Apply Circular Convolution**: Now perform circular convolution on the zero-padded signals. Circular convolution is performed efficiently using FFTs, where the signals are transformed into the frequency domain, multiplied point-by-point, and then inverse-transformed.

**Example**: Consider two sequences x[n]=[1,2,3] and h[n]=[0,1,0.5].

1. Zero-padding: We extend both sequences to a length of 5 (since 3+3−1=5).
   * Zero-padded x[n] becomes [1,2,3,0,0].
   * Zero-padded h[n]h[n]h[n] becomes [0,1,0.5,0,0].
2. Perform circular convolution of these two zero-padded sequences using FFT.

**Transforming Circular Convolution to Linear Convolution**: To convert circular convolution to linear convolution:

1. **Extend the length**: If you perform circular convolution with length NNN, the linear convolution result will be obtained by zero-padding both signals to a length NNN (for signals of lengths Lx and Lh, use N=Lx+Lh−1).
2. **Perform Linear Convolution**: After zero-padding the sequences, you can perform linear convolution (using FFTs or direct summation). The result will be a sequence whose length is Lx+Lh−1.

**Example**: Using the same sequences:

* We would zero-pad the sequences to the required length, N=5, and perform circular convolution as described above. After this, the result will represent the linear convolution, though the circular nature of the FFT would have wrapped the signal (leading to a similar but distinct result without proper zero-padding).