NAME - SANTOSH - [CB.EN.U4CCE20053]
DEPARTMENT - COMPUTER AND COMMUNICATION ENGINEERING (CCE)

LAB TITLE AND CODE: SIGNAL PROCESSING LAB 19CCE251
EXPERIMENT NUMBER: 9
DATE: 14/12/2021

DISCRETE FOURIER TRANSFORM

* **AIM:**
For a given sequence $x[n]$, compute DFT coefficients using the direct and linear transformation methods. Sketch the magnitude and phase spectrum obtained from the same.

* **SOFTWARE REQUIRED:**
Spyder IDE - Anaconda 3 2021.11 (Python 3.9.7 64-bit)
Publisher - Anaconda, Inc.

* **THEORY:** (DIRECT METHOD)
The discrete Fourier transform (DFT) is a method for converting a sequence of N complex numbers $x_0, x_1, \ldots, x_{N-1}$ to a new sequence of N complex numbers,

$$x_k = \sum_{n=0}^{N-1} x_n e^{-2\pi j kn/N}$$

for $0 \le k \le N-1$.

The x_i are thought of as the values of a function, or signal, at equally spaced times $t = 0, 1, \ldots, N-1$. The output x_k is a complex number which encodes the amplitude and phase of a sinusoidal wave with frequency $\frac{k}{N}$ cycles per unit time.

[This comes from Euler's formula -
$e^{2\pi j kn/N} = \cos(2\pi kn/N) + j \sin(2\pi kn/N)$]

PTO

(b)

The effect of computing the X_k is to find the coefficients of an approximation of the signal by a linear combination of such waves. Since each wave has an integer number of cycles per N time units, the approximation will be periodic with period N. This approximation is given by the inverse Fourier transform -

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \, e^{2\pi j kn/N}$$

→ TWIDDLE FACTOR -

It is denoted as W_N and defined as $W_N = e^{-j2\pi/N}$. Its magnitude is always maintained at unity. Phase of $W_N = -2\pi/N$. It is a vector on unit circle and is used for computational convenience. mathematically, it can be shown as -

$$W_N^n = W_N^{n \pm N} = W_N^{n \pm 2N} = \dots$$

It is a function of n and period N.

consider $N = 8$, $n = 0, 1, 2, 3, \dots, 14, 15, 16, \dots$

$$\Rightarrow W_8^0 = W_8^8 = W_8^{16} = \dots = \dots = W_8^{32} = \dots = 1 = 1 \angle 0$$

$$W_8^1 = W_8^9 = W_8^{17} = \dots = \dots = W_8^{33} = \dots = \frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}} = 1 \angle -\frac{\pi}{4}$$

→ LINEAR TRANSFORMATION -

We know that -

$$DFT(k) = DFT[x(n)] = X\left(\frac{2\pi}{N} k\right) = \sum_{n=0}^{N-1} x(n) \cdot W_N^{-nk} ;$$

$$k = 0, 1, \dots, N-1$$

$$x(n) = IDFT[X(k)]$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} X(k) \cdot W_N^{-nk} \; ; n = 0, 1, \dots, N-1$$

Computation of DFT can be performed with N^2 complex multiplication and $N(N-1)$ complex addition

$$x_N = \begin{bmatrix} x(0) \\ x(1) \\ \cdot \\ \cdot \\ x(N-1) \end{bmatrix}$$; N-point vector of signal x_N

$$x_N = \begin{bmatrix} x(0) \\ x(1) \\ \cdot \\ \cdot \\ x(N-1) \end{bmatrix}$$; N-point vector of signal x_N

$$W_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{N-1} \\ \cdot & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$; matrix of linear transformation

N-point DFT in matrix form is given by $x_N = W_N x_N$

$$\Rightarrow x_N = W_N^{-1} X_N$$

IDFT in matrix form is given by -

$$x_N = \frac{1}{N} W_N^* X_N$$

Comparing these expressions of x_N,

$$W_N^{-1} = \frac{1}{N} W_N^* \quad \text{and} \quad W_N \times W_N^* = N[I]_{N \times N}$$

Therefore, W_N is a linear transformation matrix, an orthogonal unitary matrix. From periodic property of W_N and from its symmetric property, it can be concluded that,

$$W_N^{k+N/2} = -W_N^k$$

* **GRAPH PLOTTING ALGORITHM:**

The following steps are followed-

1. Define the x-axis and corresponding y-axis as lists.
2. Plot them on canvas using .plot() function.
3. Give a name to x-axis using .xlabel() and .ylabel() functions.
4. Give a title to your plot using the .title() function.
5. Finally, to view your plot, we use the .show() function.

* **THEORETICAL CALCULATION: (DIRECT METHOD)**

Given sequence, $x[n] = [0\ 1\ 2\ 3]$

By keen observation, we find that fundamental period, $N=4$

Angular frequency, $\Omega_0 = \frac{2\pi}{4}$

$$x[k] = \sum_{n=0}^{N-1} x[n]\, e^{-jk\frac{2\pi}{4}n} = \sum_{n=0}^{3} x[n]\, e^{-jk\frac{2\pi}{4}n}$$

$$x[k] = x[0]\, e^{-jk\frac{2\pi}{4}(0)} + x[1]\, e^{-jk\frac{2\pi}{4}(1)} + x[2]\, e^{-jk\frac{2\pi}{4}(2)} + x[3]\, e^{-jk\frac{2\pi}{4}(3)}$$

$$x[k] = 0(1) + 1\left(e^{-jk\frac{2\pi}{4}}\right) + 2\left(e^{-jk\frac{4\pi}{4}}\right) + 3\left(e^{-jk\frac{6\pi}{4}}\right)$$

Substituting the values of k, we get-

$x[0] = 6+0j$, for $k=0$

$x[1] = -2+2j$, for $k=1$

$x[2] = -2+0j$, for $k=2$

$x[3] = -2-2j$, for $k=3$

Upon computing the magnitude spectrum, we find that-

$|x[0]| = \sqrt{6^2+0^2} = 6.00$

$|x[1]| = \sqrt{(-2)^2+(2)^2} = 2.83$

$|x[2]| = \sqrt{(-2)^2+(0)^2} = 2$

$|x[3]| = \sqrt{(-2)^2+(-2)^2} = 2.83$

Upon computing the phase spectrum, we find that -

$(y[n]) = \tan^{-1}\left(\frac{\text{Im}}{\text{Re}}\right) = \_\_^{\circ} = \_\_ * \frac{\Pi}{180^{\circ}}$ radians

$(y[0]) = 0^{\circ} = 0$ radians
$(y[1]) = 135^{\circ} = -0.79$ radians
$(y[2]) = 180^{\circ} = 3.68$ radians
$(y[3]) = -135^{\circ} = 0.79$ radians

→ LINEAR TRANSFORMATION -

The first step is to determine the matrix W_4. By exploiting the periodicity property of W_N and the symmetric property -

$$W_N^{k+N} = -W_N^{k}$$

the matrix W_4 may be expressed as -

$$W_4 = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^4 & W_4^6 \\ 1 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & 1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

Then,

$$x_4 = W_4 a_4 = \begin{bmatrix} 6 \\ -2+2j \\ -2 \\ -2-2j \end{bmatrix}$$

Upon computing the magnitude spectrum, we find that -

$|x[0]| = \sqrt{6^2+0^2} = 6.00$
$|x[1]| = \sqrt{(-2)^2+(2)^2} = 2.83$
$|x[2]| = \sqrt{(-2)^2+(0)^2} = 2$
$|x[3]| = \sqrt{(-2)^2+(-2)^2} = 2.83$

Upon computing the phase spectrum, we find that -

$$\angle x[n] = \tan^{-1}\left(\frac{H}{R}\right) = -\theta = -\theta \times \frac{\pi}{180} \text{ radians}$$

Therefore,

$\angle x[0] = 0^\circ = 0$ radians
$\angle x[1] = -135^\circ = -0.79$ radians
$\angle x[2] = 180^\circ = 3.68$ radians
$\angle x[3] = 135^\circ = 0.79$ radians

* PROGRAM WITH COMMENTS:

```
# Function to calculate X(k) for Direct method
def summation
    sum = 0
    for n in range (size_x):
        exponential = np.exp(-2j * np.pi * k * n / size_x)
        sum = sum + float(x[n]) * exponential
    return (sum)

# import library source files
import numpy as np
import matplotlib.pyplot as plt

#Given Input Sequence x[n]
size_x = int(input("Enter the size of input x[n]: "))
x = [0] * (size_x)
print("Enter the elements of the input x[n] one-by-one as follows: -")
for sample in range(0, size_x, 1):
    x[sample] = input("element " + str(sample+1) + ": ")

# sketch Input Sequence
plt.xlabel('n')
plt.ylabel('x[n]')
```

```
plt.title("Input Sequence x[n]".format(x))
plt.stem(np.arange(0, size_x), x)
plt.grid(True) # Configure the grid lines
plt.show()
print("\n The entered input x[n] is: -" + str(x) + "\n")

# Compute Discrete Fourier Transform Coefficients Using Direct Method
fourier_transform_direct = []
for k in range(size_x):
    fourier_transform_direct.append(summation(k))
print("\n The Discrete Fourier Transform (DFT) coefficients of the sequence x[n] obtained using direct method are as follows: {} ".format(fourier_transform_direct))

# Compute Magnitude and Phase Spectrum of DFT Coefficients
magnitude_spectrum_direct = []
phase_spectrum_direct = []

for sample in range(len(fourier_transform_direct)):
    magnitude_spectrum_direct.append((fourier_transform_direct[sample].real**2 + fourier_transform_direct[sample].imag**2)**0.5)
    phase_direct = np.arctan(fourier_transform_direct[sample].imag / fourier_transform_direct[sample].real)
    phase_spectrum_direct.append(phase_direct)

# Sketch Magnitude Spectrum Using Direct Method
plt.xlabel('k')
plt.ylabel('|x[k]|')
plt.title("Magnitude Spectrum obtained Using Direct Method")
```

PTO

```
plt.stem(np.arange(0, len(magnitude_spectrum_direct)), magnitude_spectrum_direct)
plt.grid(True) #Configure the grid lines
plt.show()
print("\n The magnitude spectrum of DFT coefficients obtained using direct method are as follows : {}".format(magnitude_spectrum_direct))

# sketch Phase spectrum using Direct method
plt.xlabel('x')
plt.ylabel('Angle (in radians)')
plt.title("Phase Spectrum obtained using Direct Method")
plt.stem(np.arange(0, len(phase_spectrum_direct)), phase_spectrum_direct)
plt.show()
plt.grid(True) # Configure the grid lines
print("\n The phase spectrum of DFT coefficients obtained using direct method are as follows : {}".format(phase_spectrum_direct))

# Compute DFT coefficients using Linear Transformation method

# Compute W(N) 1D Array
rl = cl = size_x
wn = []
for i in range(rl):
    for j in range(cl):
        wn.append(np.exp(-2j * np.pi * i * j / size_x))

# numpy.reshape() is used to give a new shape to an array without changing its data.
```

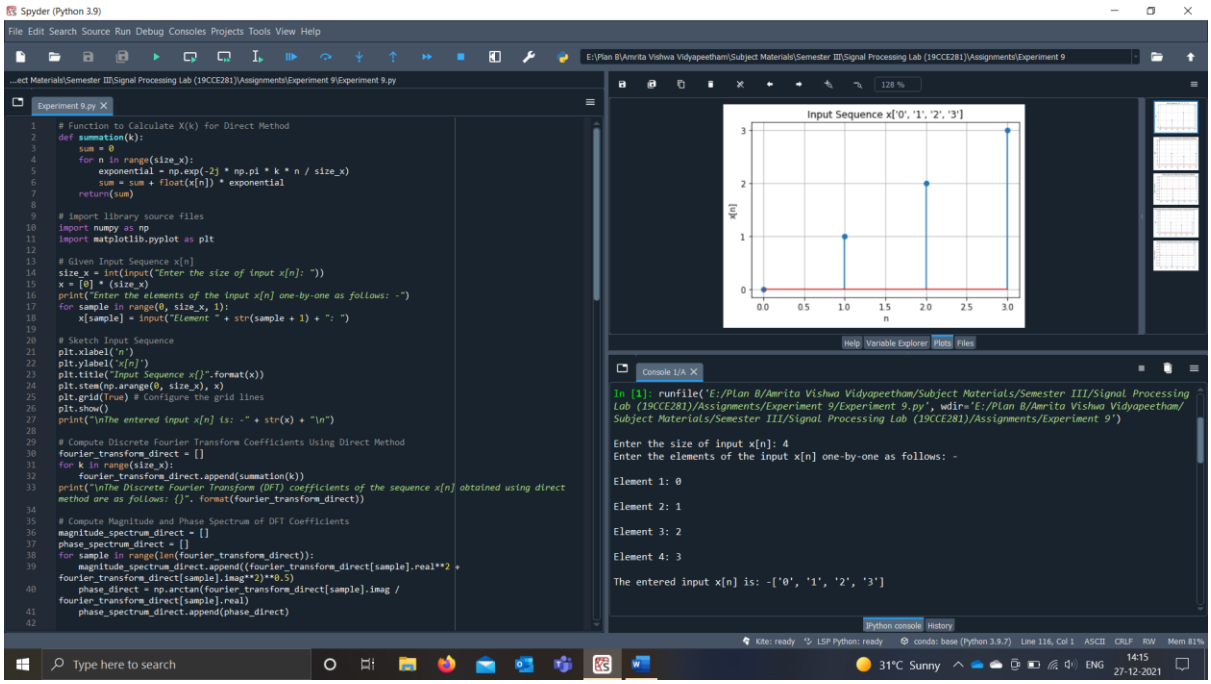
```
wn_multidim = np.reshape(wn, (r1, c1)) # An N*N W(N) matrix
print("\n The matrix W(n) is as follows: \n{}".format
(wn_multidim))
r2 = size_x ; c2 = 1
x_multidim = np.reshape(x,(r2, c2)) # An N*1 x(N) matrix
print("\n The matrix x(N) is as follows: \n{}".format
(x_multidim))

#Compute X(N) = W(N) * x(N), an N*1 matrix
fourier_transform_multidim = [[0] * c2] * r1
#Null Multidimensional Array
fourier_transform_l_t = [] # Convert multidimensional
Array to 1D
for i in range(r1):
    for j in range(c2):
        fourier_transform_multidim[i][j] = 0
        for k in range(c1):
            fourier_transform_multidim[i][j] +=
wn_multidim[i][k] * float(x_multidim[k][j])
        temp = fourier_transform_multidim[i][j]
        fourier_transform_l_t.append(fourier_transform
_multidim[i][j])

print("\n The Discrete Fourier Transform (DFT) coefficients
of the sequence x[n] obtained using linear transformation
are as follows: {}".format(fourier_transform_l_t))

#Compute magnitude and Phase spectrum of DFT Coefficients
magnitude_spectrum_l_t = []
phase_spectrum_l_t = []
for sample in range(len(fourier_transform_l_t)):
    magnitude_spectrum_l_t.append((fourier_transform_l_t
[sample].real**2 + fourier_transform_l_t[sample].imag**2)**0.5)
```

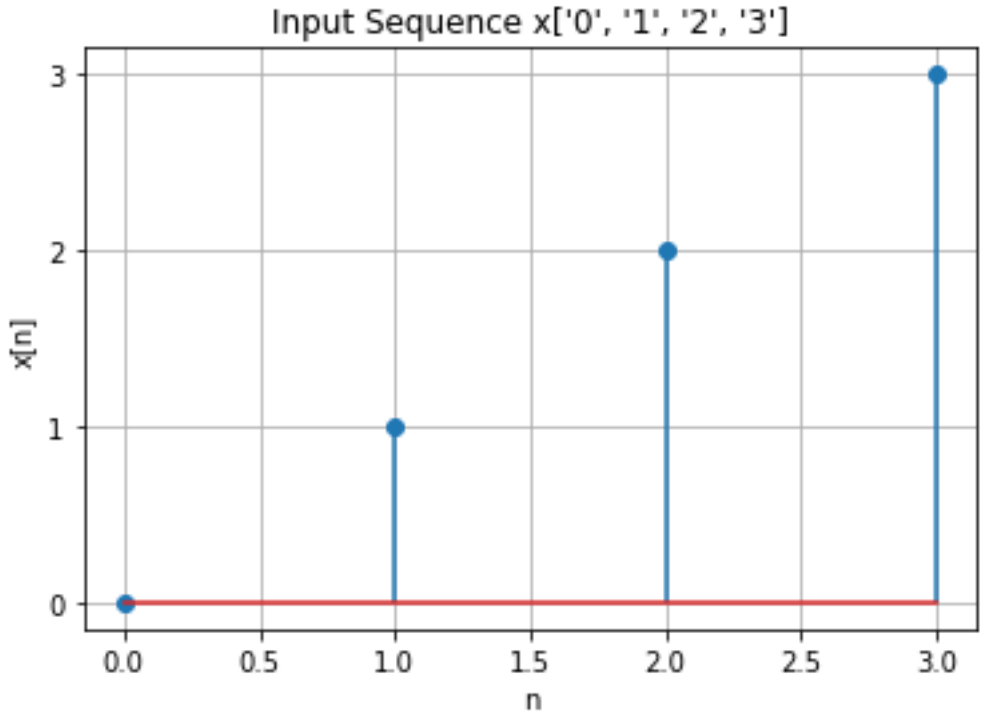
```
96      phase_l_t = np.arctan(fourier_transform_l_t[sample].
imag / fourier_transform_l_t[sample].real)
97      phase_spectrum_l_t.append(phase_l_t)
98
99  # sketch magnitude spectrum using Linear Transformation
    method
100 plt.xlabel('k')
101 plt.ylabel('|x[k]|')
102 plt.title("Magnitude spectrum Obtained Using Linear
    Transformation Method")
103 plt.grid(True)
104 plt.stem(np.arange(0, len(magnitude_spectrum_l_t)),
    magnitude_spectrum_l_t)
105 plt.show()
106 print("\n The magnitude spectrum of DFT coefficients obtained
    using linear transformation are as follows: {}".format
    (magnitude_spectrum_l_t))
107
108 # sketch Phase spectrum using Linear Transformation Method
109 plt.xlabel('k')
110 plt.ylabel('Angle in radians')
111 plt.title("Phase spectrum Obtained Using Linear
    Transformation method")
112 plt.stem(np.arange(0, len(phase_spectrum_l_t)),
    phase_spectrum_l_t)
113 plt.grid(True) # Configure the grid lines
114 plt.show()
115 print("\n The phase spectrum of DFT coefficients obtained
    using linear transformation method are as follows: {}"
    .format(phase_spectrum_l_t))
116
```

RESULTS

VERIFIED

Given Input Sequence x[n]





Step 1: Import library source files, NumPy and matplotlib.pyplot.

Step 2: Enter the size of input x[n] and declare an array with the number of elements equal to the value of size.

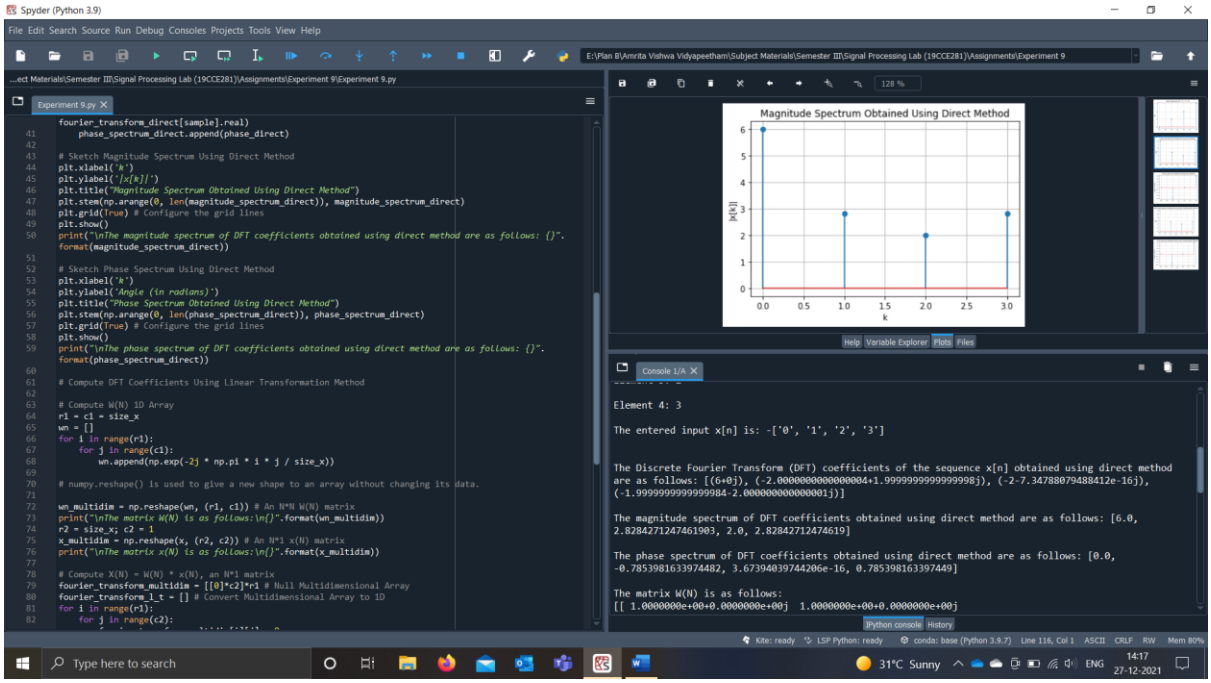
Step 3: Enter the elements of the input x[n] and show the labelled plot.

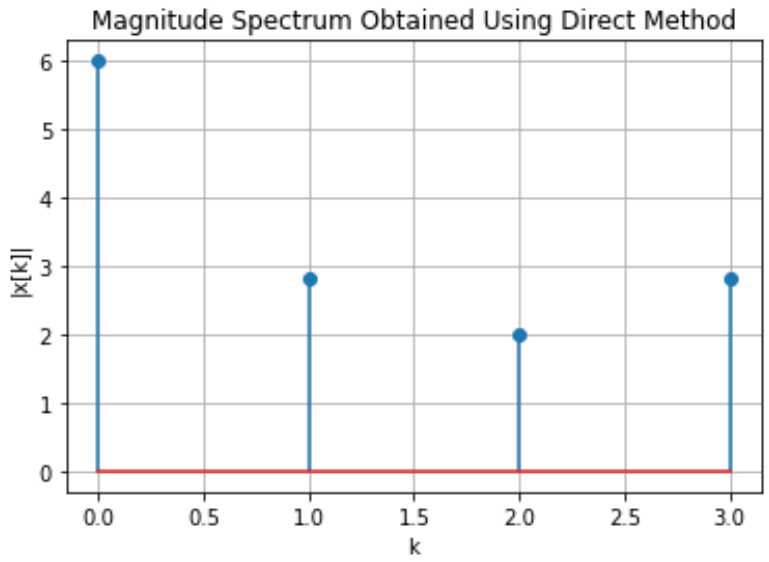
Step 4: Compute Discrete Fourier Transform (DFT) coefficients using the direct method and find its magnitude and phase spectrum.

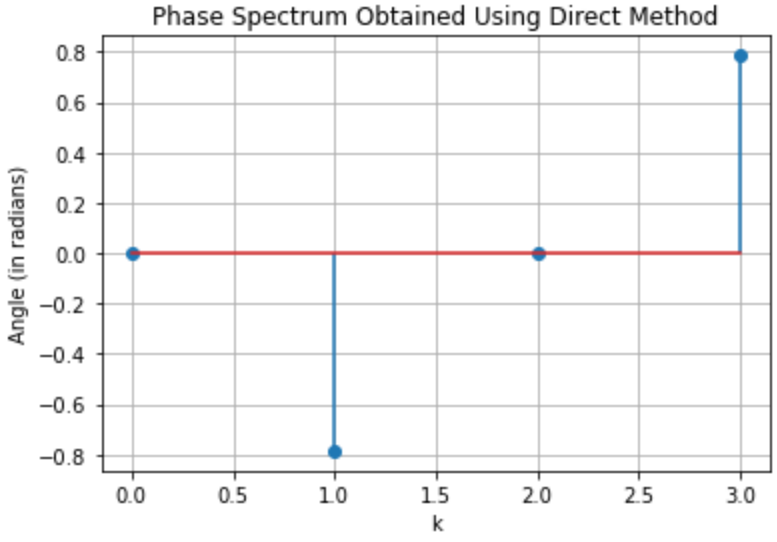
Step 5: Compute W(N) 1D Array followed by X(N) = W(N) * x(N), an N*1 matrix.

Step 6: Repeat step 4 for the linear transformation method.

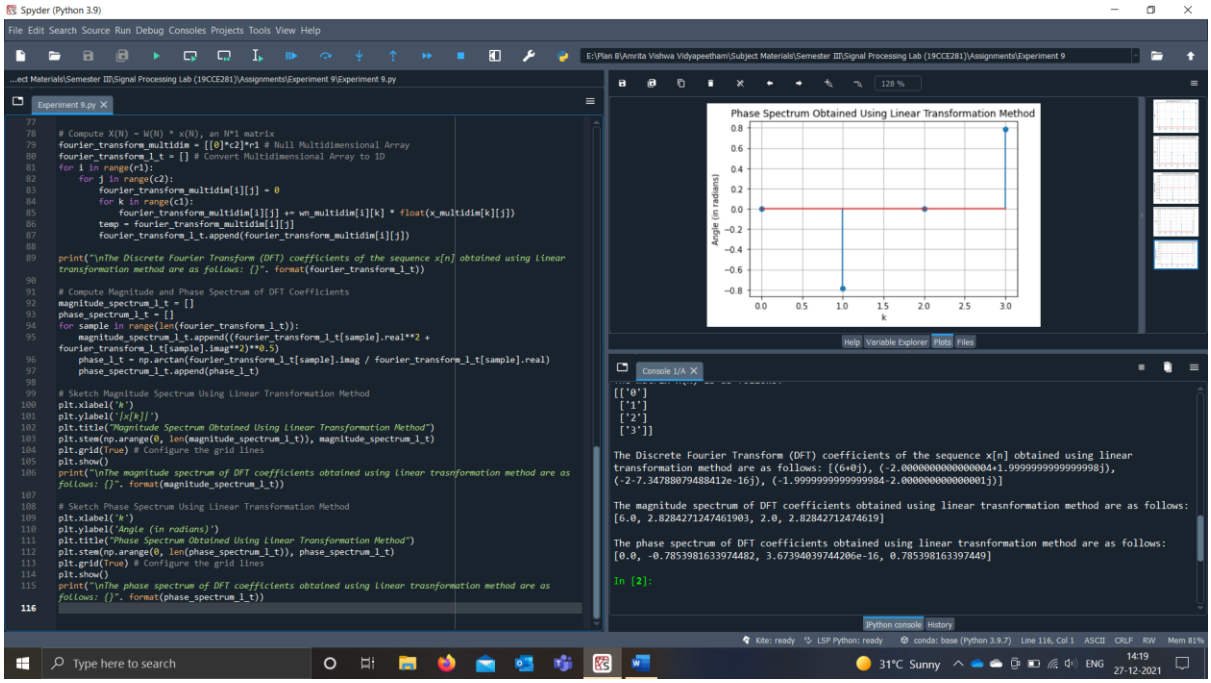
Compute DFT Coefficients Using Direct Method

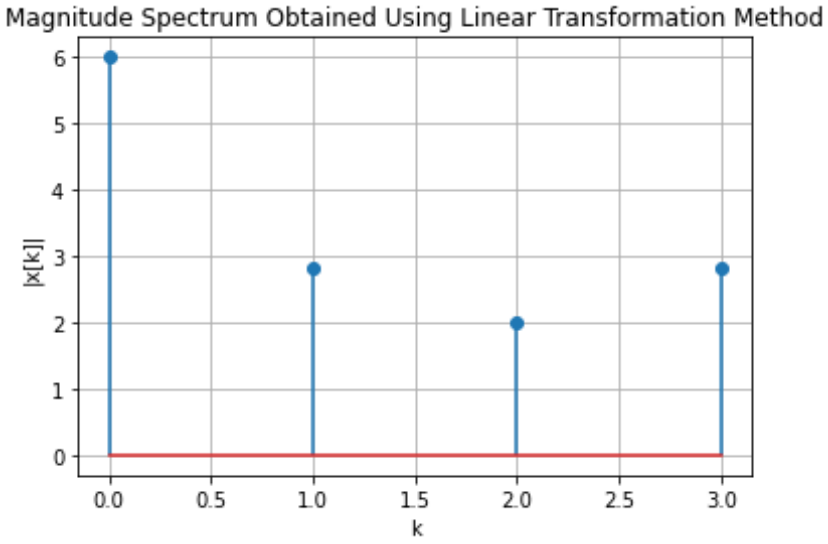


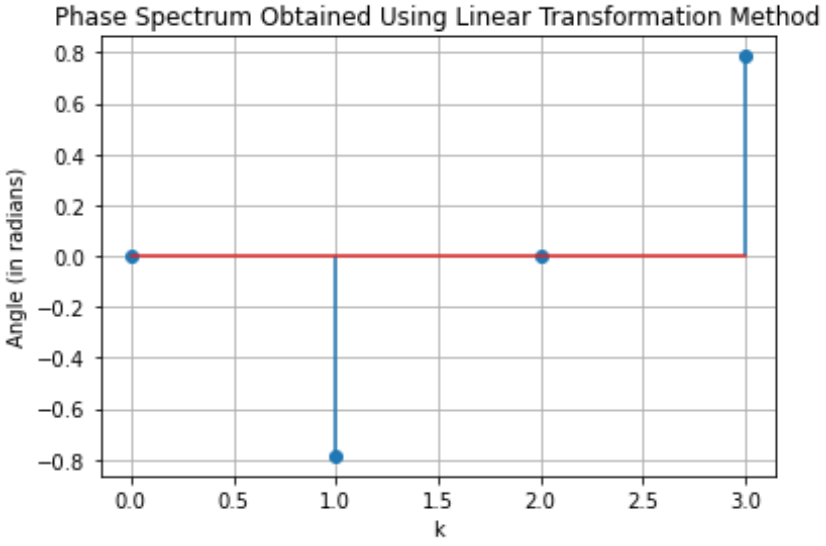




Compute DFT Coefficients Using Linear Transformation Method







```
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.29.0 -- An enhanced Interactive Python.

Restarting kernel...
```

```
In [1]:         'E:/Plan B/Amrita Vishwa Vidyapeetham/Subject Materials/Semester III/
Signal Processing Lab (19CCE281)/Assignments/Experiment 9/Experiment 9.py'      ='E:/Plan
B/Amrita Vishwa Vidyapeetham/Subject Materials/Semester III/Signal Processing Lab
(19CCE281)/Assignments/Experiment 9'

Enter the size of input x[n]: 4
Enter the elements of the input x[n] one-by-one as follows: -

Element 1: 0

Element 2: 1

Element 3: 2

Element 4: 3

The entered input x[n] is: -['0', '1', '2', '3']


The Discrete Fourier Transform (DFT) coefficients of the sequence x[n] obtained using
direct method are as follows: [(6+0j), (-2.0000000000000004+1.9999999999999998j),
(-2-7.34788079488412e-16j), (-1.9999999999999984-2.000000000000001j)]

The magnitude spectrum of DFT coefficients obtained using direct method are as follows:
[6.0, 2.8284271247461903, 2.0, 2.82842712474619]

The phase spectrum of DFT coefficients obtained using direct method are as follows: [0.0,
-0.7853981633974482, 3.67394039744206e-16, 0.785398163397449]

The matrix W(N) is as follows:
[[ 1.0000000e+00+0.0000000e+00j  1.0000000e+00+0.0000000e+00j
   1.0000000e+00+0.0000000e+00j  1.0000000e+00+0.0000000e+00j]
 [ 1.0000000e+00+0.0000000e+00j  6.1232340e-17-1.0000000e+00j
  -1.0000000e+00-1.2246468e-16j -1.8369702e-16+1.0000000e+00j]
 [ 1.0000000e+00+0.0000000e+00j -1.0000000e+00-1.2246468e-16j
   1.0000000e+00+2.4492936e-16j -1.0000000e+00-3.6739404e-16j]
 [ 1.0000000e+00+0.0000000e+00j -1.8369702e-16+1.0000000e+00j
  -1.0000000e+00-3.6739404e-16j  5.5109106e-16-1.0000000e+00j]]

The matrix x(N) is as follows:
[['0']
 ['1']
 ['2']
 ['3']]

The Discrete Fourier Transform (DFT) coefficients of the sequence x[n] obtained using
```

linear transformation method are as follows: [(6+0j),
(-2.0000000000000004+1.9999999999999998j), (-2-7.34788079488412e-16j),
(-1.9999999999999984-2.000000000000001j)]

The magnitude spectrum of DFT coefficients obtained using linear trasnformation method are as follows: [6.0, 2.8284271247461903, 2.0, 2.82842712474619]

The phase spectrum of DFT coefficients obtained using linear trasnformation method are as follows: [0.0, -0.7853981633974482, 3.67394039744206e-16, 0.785398163397449]

In [2]: