EE3025 IDP Assignment-1

C.SriRam Saran-EE18BTECH11007

Download all python codes from

https://github.com/csrs2000/EE3025_IDP/tree/main/Assignment−1/code

and latex-tikz codes from

https://github.com/csrs2000/EE3025_IDP/tree/main/Assignment−1

1 Question

1.1. Let

$$x(n) = \left\{\underset{\uparrow}{1}, 2, 3, 4, 2, 1\right\} \tag{1.1.1}$$

$$y(n) + \frac{1}{2}y(n-1) = x(n) + x(n-2) \tag{1.1.2}$$

Compute

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{1.1.3}$$

and $H(k)$ using $h(n)$.

2 Solution

2.1. When Unit impulse signal is given as input to the LTI system then its impulse response is the ouput of the system. So,from equation (1.1.2) we can say that the Impulse response of the system is,

$$h(n) + \frac{1}{2}h(n-1) = \delta(n) + \delta(n-2) \tag{2.1.1}$$

2.2. DFT of a Input Signal $x(n)$ is :

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{2.2.1}$$

Similarly, DFT of Impulse Response $h(n)$ is,

$$H(k) \triangleq \sum_{n=0}^{N-1} h(n) e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{2.2.2}$$

2.3. Dft matrix multiplication method- DFT of x(n) and h(n) using Matrix multiplication method

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \tag{2.3.1}$$

Lets take $W_N^{nk} = e^{-j2\pi kn/N}$, expressing $X(k)$ in terms of product of DFT matrix and $x(k)$:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & W_N^1 & W_N^2 & W_N^3 & W_N^4 & W_N^5 \\ 1 & W_N^2 & W_N^4 & W_N^6 & W_N^8 & W_N^{10} \\ 1 & W_N^3 & W_N^6 & W_N^9 & W_N^{12} & W_N^{15} \\ 1 & W_N^4 & W_N^8 & W_N^{12} & W_N^{16} & W_N^{20} \\ 1 & W_N^5 & W_N^{10} & W_N^{15} & W_N^{20} & W_N^{25} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \end{bmatrix} \tag{2.3.2}$$

given x(n) = $\left\{\underset{\uparrow}{1}, 2, 3, 4, 2, 1\right\}$
,here N=6 so above equation becomes

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} = \begin{bmatrix} 1+2+3+4+2+1 \\ 1+(2)e^{-j\pi/3} + ... + (1)e^{-j5\pi/3} \\ 1+(2)e^{-2j\pi/3} + ... + (1)(e^{-2j5\pi/3} \\ 1+(2)e^{-3j\pi/3} + ... + (1)e^{-3j5\pi/3} \\ 1+(2)e^{-4j\pi/3} + ... + (1)e^{-4j5\pi/3} \\ 1+(2)e^{-5j\pi/3} + ... + (1)e^{-5j5\pi/3} \end{bmatrix} \tag{2.3.3}$$

On solving we get,

$$\implies X(0) = 13 + 0j, \tag{2.3.4}$$

$$X(1) = -4 - 1.732j, \tag{2.3.5}$$

$$X(2) = 1 + 0j, \tag{2.3.6}$$

$$X(3) = -1 + 0j, \tag{2.3.7}$$

$$X(4) = 1 + 0j, \tag{2.3.8}$$

$$X(5) = -4 + 1.732j \tag{2.3.9}$$

in order to find $H(k)$ we need to find h(n) first. So we will first calculate h(n). we will find that using Y(Z),X(Z)For that we need to first find the Y(z) by applying Z-transform on equation

(1.1.2) i.e.,

$$Y(z) + \frac{1}{2}z^{-1}Y(z) = X(z) + z^{-2}X(z) \tag{2.3.10}$$

$$\implies Y(z) = \frac{2(z^2+1)}{z(2z+1)}X(z) \tag{2.3.11}$$

Now we can find H(z) using Y(z) i.e.,

$$H(z) = \frac{Y(z)}{X(z)} \tag{2.3.12}$$

$$H(z) = \frac{2(z^2+1)}{z(2z+1)} \tag{2.3.13}$$

$$H(z) = \frac{1+z^{-2}}{1+\frac{1}{2}z^{-1}} \tag{2.3.14}$$

h(n) is,

$$H(Z) = Z^{-1}\left[\frac{1}{1+\frac{1}{2}z^{-1}} + \frac{z^{-2}}{1+\frac{1}{2}z^{-1}}\right] \tag{2.3.15}$$

$$h(n) = \left[\frac{-1}{2}\right]^{n} u(n) + \left[\frac{-1}{2}\right]^{n-2} u(n-2) \tag{2.3.16}$$

lets assume that length of h(n) is 6,i.e, N = 6. using matrix multiplication method we obtain

$$\begin{bmatrix} H(0) \\ H(1) \\ H(2) \\ H(3) \\ H(4) \\ H(5) \end{bmatrix} = \begin{bmatrix} h(0) + .. + h(5) \\ h(0) + h(1)e^{-j\pi/3} + .. + h(5)e^{-j5\pi/3} \\ h(0) + h(1)e^{-2j\pi/3} + .. + h(5)e^{-2j5\pi/3} \\ h(0) + h(1)e^{-3j\pi/3} + .. + h(5)e^{-3j5\pi/3} \\ h(0) + h(1)e^{-4j\pi/3} + ... + h(5)e^{-4j5\pi/3} \\ h(0) + h(1)e^{-5j\pi/3} + .. + h(5)e^{-5j5\pi/3} \end{bmatrix} \tag{2.3.17}$$

On calculating we get,

$$\implies H(0) = 1.28125 + 0j, \tag{2.3.18}$$

$$H(1) = 0.51625 - 0.5141875j, \tag{2.3.19}$$

$$H(2) = -0.078125 + 1.1095625j, \tag{2.3.20}$$

$$H(3) = 3.84375 + 0j, \tag{2.3.21}$$

$$H(4) = -0.071825 - 1.1095625j. \tag{2.3.22}$$

$$H(5) = 0.515625 + 0.5141875j \tag{2.3.23}$$

lets verify these values by using python code

2.4. Python Code to compute the DFT of $x(n)$ and $h(n)$ is given below

```
https://github.com/csrs2000/EE3025_IDP/
    blob/main/Assignment−1/code/
    ee18btech11007_1.py
```

2.5. using the above code we get the following plots.

```
https://github.com/csrs2000/EE3025_IDP/tree
    /main/Assignment−1/figs
```

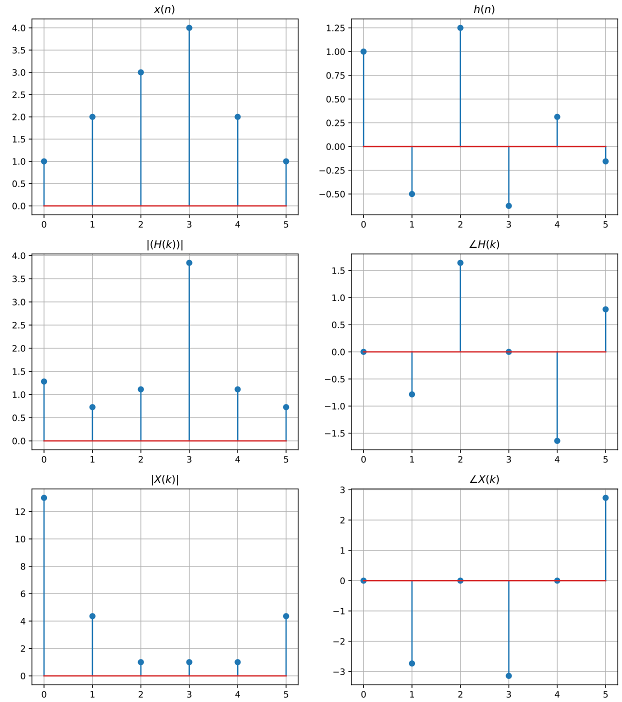


Fig. 2.5: Plots of x(n) and h(n),their DFTs

manually calculated values match with the values obtained from codes

3 Approach-2

3.1. *Properties of W_N*

a) Property-1 :

$$W_N^2 = W_{N/2}$$

b) Property-2 :

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

3.2. a N-point DFT is computed as,

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \quad k = 0, 1, \ldots, N-1 \tag{3.2.1}$$

now for easier calculations we can Divide the inputs into even and odd indices and using Property-1,

$$X(k) = \sum_{n=even} x(n) W_N^{kn} + \sum_{n=odd} x(n) W_N^{kn} \tag{3.2.2}$$

$$= \sum_{m=0}^{N/2-1} x(2m) W_N^{2mk} + \sum_{m=0}^{N/2-1} x(2m+1) W_N^{(2m+1)k} \tag{3.2.3}$$

$$= \sum_{m=0}^{N/2-1} x(2m) W_{N/2}^{mk} + W_N^k \sum_{m=0}^{N/2-1} x(2m+1) W_{N/2}^{mk} \tag{3.2.4}$$

by observing,

$$X(k) = X_e(k) + W_N^k X_o(k) \tag{3.2.5}$$

for the given question N = 6, we can express the even odd DFT's $X_e(k)$, $X_o(k)$ interms of block matrices, we obtain,

$$\begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_o(0) \\ X_o(1) \\ X_o(2) \end{bmatrix} = \begin{bmatrix} W_3^0 & W_3^0 & W_3^0 & 0 & 0 & 0 \\ W_3^0 & W_3^1 & W_3^2 & 0 & 0 & 0 \\ W_3^0 & W_3^2 & W_3^4 & 0 & 0 & 0 \\ 0 & 0 & 0 & W_3^0 & W_3^0 & W_3^0 \\ 0 & 0 & 0 & W_3^0 & W_3^1 & W_3^2 \\ 0 & 0 & 0 & W_3^0 & W_3^2 & W_3^4 \end{bmatrix} \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(1) \\ x(3) \\ x(5) \end{bmatrix} \tag{3.2.6}$$

Lets take , F_N as a N-Point DFT Matrix and P_N as an odd-even Permutation Matrix, we can write above equation as

$$\begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_o(0) \\ X_o(1) \\ X_o(2) \end{bmatrix} = \begin{bmatrix} F_3 & 0 \\ 0 & F_3 \end{bmatrix} P_6 x \tag{3.2.7}$$

where

$$P_6 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2.8}$$

and

$$P_6 x = P_6 \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(1) \\ x(3) \\ x(5) \end{bmatrix} \tag{3.2.9}$$

Now,using (3.2.5) we can formulate $X(k)$ interms of $X_e(k)$ and $X_o(k)$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & W_6^0 & 0 & 0 \\ 0 & 1 & 0 & 0 & W_6^1 & 0 \\ 0 & 0 & 1 & 0 & 0 & W_6^2 \\ 1 & 0 & 0 & W_6^3 & 0 & 0 \\ 0 & 1 & 0 & 0 & W_6^4 & 0 \\ 0 & 0 & 1 & 0 & 0 & W_6^5 \end{bmatrix} \begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_o(0) \\ X_o(1) \\ X_o(2) \end{bmatrix} \tag{3.2.10}$$

lets assume I_3 to be 3x3 identity matrix and $D_N = diag(1, W_N, W_N^2, ..., W_N^{N-1})$ .accordingly we get $D_{\frac{N}{2}} = diag(1, W_N, W_N^2, ..., W_N^{\frac{N}{2}-1})$.so D_3 will be,

$$D_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & W_6^1 & 0 \\ 0 & 0 & W_6^2 \end{bmatrix} \tag{3.2.11}$$

Using Property-2, Equation (3.2.10) gets expressed in terms of D_3 and I_3 as,

$$X = \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} = \begin{bmatrix} I_3 & D_3 \\ I_3 & -D_3 \end{bmatrix} \begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_o(0) \\ X_o(1) \\ X_o(2) \end{bmatrix} \tag{3.2.12}$$

Finally using Eq (3.2.12) and Eq (3.2.7) we obtain

$$X = \begin{bmatrix} I_3 & D_3 \\ I_3 & -D_3 \end{bmatrix} \begin{bmatrix} F_3 & 0 \\ 0 & F_3 \end{bmatrix} P_6 x \tag{3.2.13}$$

we know $X = F_6 x$ for N = 6 hence we obtain F_6 as ;

$$\implies F_6 = \begin{bmatrix} I_3 & D_3 \\ I_3 & -D_3 \end{bmatrix} \begin{bmatrix} F_3 & 0 \\ 0 & F_3 \end{bmatrix} P_6 \tag{3.2.14}$$

above approach can be used for any arbitary N , lets take a N-point DFT Matrix and express it in terms of N/2-point DFT Matrix as

$$F_N = \begin{bmatrix} I_{N/2} & D_{N/2} \\ I_{N/2} & -D_{N/2} \end{bmatrix} \begin{bmatrix} F_{N/2} & 0 \\ 0 & F_{N/2} \end{bmatrix} P_N \tag{3.2.15}$$

3.3. Now,for any $N = 2^m$ where $m \in \mathbb{Z}^+$ we can recursively breakdown N/2 point DFT Matrix to N/4 point DFT Matrix ..so on untill we reach 2-point DFT Matrix.for N = 8, we can break it down using Eq (3.2.15) as follows,

$$F_8 = \begin{bmatrix} I_4 & D_4 \\ I_4 & -D_4 \end{bmatrix} \begin{bmatrix} F_4 & 0 \\ 0 & F_4 \end{bmatrix} P_8 \tag{3.3.1}$$

$$F_4 = \begin{bmatrix} I_2 & D_2 \\ I_2 & -D_2 \end{bmatrix} \begin{bmatrix} F_2 & 0 \\ 0 & F_2 \end{bmatrix} P_4 \tag{3.3.2}$$

Finally,we reach the 2-point DFT Matrix base case

$$F_2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 + x_2 \\ x_1 - x_2 \end{bmatrix} \tag{3.3.3}$$

3.4. lets compute the 8-point dft using Step by Step visualization of recursion.

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_e(3) \end{bmatrix} + \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_o(0) \\ X_o(1) \\ X_o(2) \\ X_o(3) \end{bmatrix} \tag{3.4.1}$$

$$\begin{bmatrix} X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_e(3) \end{bmatrix} - \begin{bmatrix} W_8^0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 \\ 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & W_8^3 \end{bmatrix} \begin{bmatrix} X_o(0) \\ X_o(1) \\ X_o(2) \\ X_o(3) \end{bmatrix} \tag{3.4.2}$$

Now, 4-point DFT's to 2-point DFT's

$$\begin{bmatrix} X_e(0) \\ X_e(1) \end{bmatrix} = \begin{bmatrix} X_{e_1}(0) \\ X_{e_1}(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_1}(0) \\ X_{o_1}(1) \end{bmatrix} \tag{3.4.3}$$

$$\begin{bmatrix} X_e(2) \\ X_e(3) \end{bmatrix} = \begin{bmatrix} X_{e_1}(0) \\ X_{e_1}(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_1}(0) \\ X_{o_1}(1) \end{bmatrix} \tag{3.4.4}$$

$$\begin{bmatrix} X_o(0) \\ X_o(1) \end{bmatrix} = \begin{bmatrix} X_{e_2}(0) \\ X_{e_2}(1) \end{bmatrix} + \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_2}(0) \\ X_{o_2}(1) \end{bmatrix} \tag{3.4.5}$$

$$\begin{bmatrix} X_o(2) \\ X_o(3) \end{bmatrix} = \begin{bmatrix} X_{e_2}(0) \\ X_{e_2}(1) \end{bmatrix} - \begin{bmatrix} W_4^0 & 0 \\ 0 & W_4^1 \end{bmatrix} \begin{bmatrix} X_{o_2}(0) \\ X_{o_2}(1) \end{bmatrix} \tag{3.4.6}$$

$$P_8 \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \\ x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} \tag{3.4.7}$$

$$P_4 \begin{bmatrix} x(0) \\ x(2) \\ x(4) \\ x(6) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(4) \\ x(2) \\ x(6) \end{bmatrix} \tag{3.4.8}$$

$$P_4 \begin{bmatrix} x(1) \\ x(3) \\ x(5) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(1) \\ x(5) \\ x(3) \\ x(7) \end{bmatrix} \tag{3.4.9}$$

at last we get,

$$\begin{bmatrix} X_{e_1}(0) \\ X_{e_1}(1) \end{bmatrix} = F_2 \begin{bmatrix} x(0) \\ x(4) \end{bmatrix} = \begin{bmatrix} x(0) + x(4) \\ x(0) - x(4) \end{bmatrix} \tag{3.4.10}$$

$$\begin{bmatrix} X_{o_1}(0) \\ X_{o_1}(1) \end{bmatrix} = F_2 \begin{bmatrix} x(2) \\ x(6) \end{bmatrix} = \begin{bmatrix} x(2) + x(6) \\ x(2) - x(6) \end{bmatrix} \tag{3.4.11}$$

$$\begin{bmatrix} X_{e_2}(0) \\ X_{e_2}(1) \end{bmatrix} = F_2 \begin{bmatrix} x(1) \\ x(5) \end{bmatrix} = \begin{bmatrix} x(1) + x(5) \\ x(1) - x(5) \end{bmatrix} \tag{3.4.12}$$

$$\begin{bmatrix} X_{o_2}(0) \\ X_{o_2}(1) \end{bmatrix} = F_2 \begin{bmatrix} x(3) \\ x(7) \end{bmatrix} = \begin{bmatrix} x(3) + x(7) \\ x(3) - x(7) \end{bmatrix} \tag{3.4.13}$$

So, $X_{e_2} \in \text{DFT}\{x(1), x(5)\}$ and $X_{o_2} \in \text{DFT}\{x(3), x(7)\}$ would combine to give X_o .And $X_{e_1} \in \text{DFT}\{x(0), x(4)\}$ and $X_{o_1} \in \text{DFT}\{x(2), x(6)\}$ would combine to give X_e.

3.5. *Time Complexity:*
In DFT we perform Matrix multiplication of NxN matrix with Nx1 vector. Hence it has $O(N^2)$ time complexity which is very tedious and slow for high values of N.
the recursion relation for for the above example can be formulated as below recursion tree

$$T(n) = 2T(n/2) + O(n) \tag{3.5.1}$$

by Solving above recurrence relation the time complexity is obtained as $O(NlogN)$ which is much faster than $O(N^2)$ for high values of N we can analyse this by plotting in python.

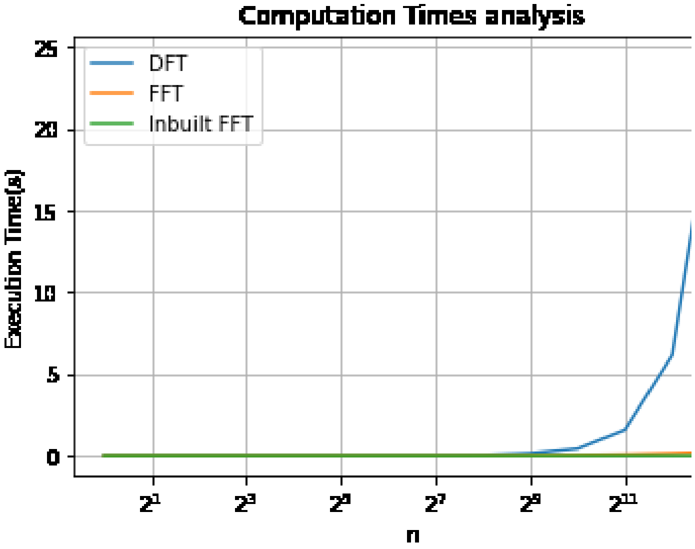
3.6. The following code plots time complexity of fft,dft

```
https://github.com/csrs2000/EE3025_IDP/
    blob/main/Assignment−1/code/
    ee18btech11007_2.py
```

3.7. Computing $X(k)$, $H(k)$ and $Y(k)$ for

$$x(n) = \left\{ \underset{\uparrow}{1}, 2, 3, 4, 2, 1, 0, 0 \right\} \tag{3.7.1}$$

with N = 8, using above FFT approach. We can observe that both the approaches yield same



plots

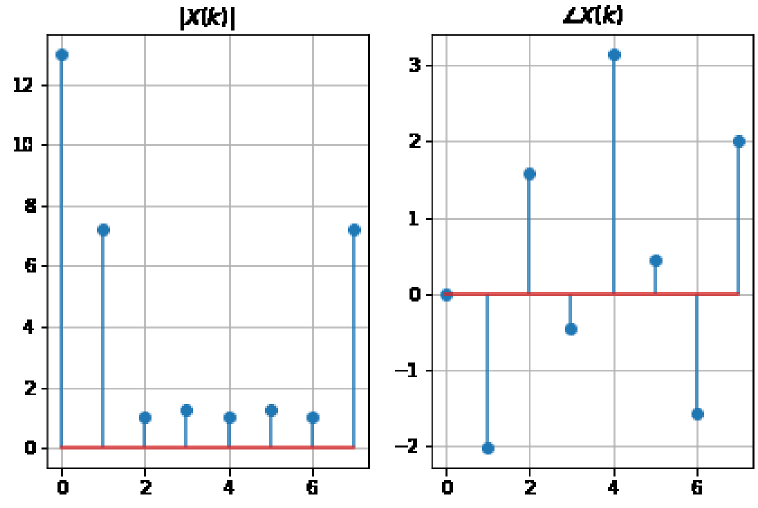
3.8. The following code plots magnitude and phase plots of $X(k)$, $H(k)$.

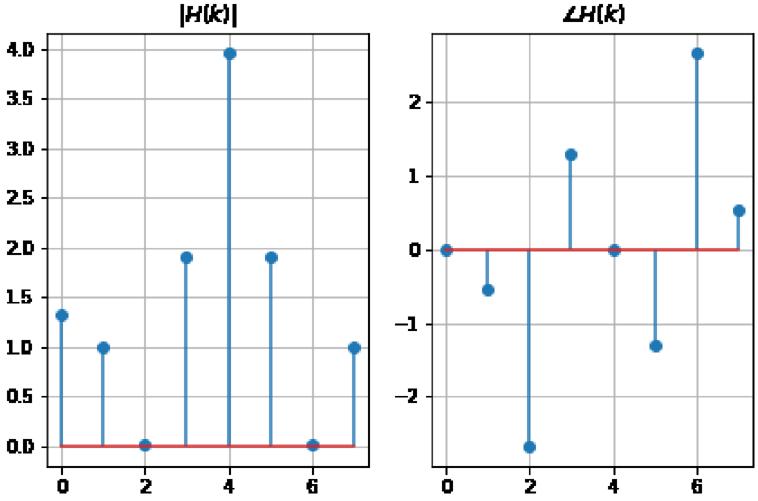
```
https://github.com/csrs2000/EE3025_IDP/
    blob/main/Assignment−1/code/
    ee18btech11007_3.py
```

3.9. Obtaining 8-Point FFT using DFT Matrix

$$\begin{bmatrix} X(0)\\ X(1)\\ X(2)\\ X(3)\\ X(4)\\ X(5)\\ X(6)\\ X(7)\end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0\\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7\\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^8 & W_8^{10} & W_8^{12} & W_8^{14}\\ W_8^0 & W_8^3 & W_8^6 & W_8^9 & W_8^{12} & W_8^{15} & W_8^{18} & W_8^{21}\\ W_8^0 & W_8^4 & W_8^8 & W_8^{12} & W_8^{16} & W_8^{20} & W_8^{24} & W_8^{28}\\ W_8^0 & W_8^5 & W_8^{10} & W_8^{15} & W_8^{20} & W_8^{25} & W_8^{30} & W_8^{35}\\ W_8^0 & W_8^6 & W_8^{12} & W_8^{18} & W_8^{24} & W_8^{30} & W_8^{36} & W_8^{42}\\ W_8^0 & W_8^7 & W_8^{14} & W_8^{21} & W_8^{28} & W_8^{35} & W_8^{42} & W_8^{49}\end{bmatrix} \begin{bmatrix} 1\\ 2\\ 3\\ 4\\ 2\\ 1\\ 0\\ 0\end{bmatrix} \tag{3.9.1}$$

$$\implies \begin{bmatrix} X(0)\\ X(1)\\ X(2)\\ X(3)\\ X(4)\\ X(5)\\ X(6)\\ X(7)\end{bmatrix} = \begin{bmatrix} 13\\ -3.121-6.535j\\ 1j\\ 1.121-0.535j\\ -1\\ 1.121+0.535j\\ -1j\\ -3.121+6.535j\end{bmatrix} \tag{3.9.2}$$





Similarly,

$$\begin{bmatrix} H(0)\\ H(1)\\ H(2)\\ H(3)\\ H(4)\\ H(5)\\ H(6)\\ H(7)\end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0\\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7\\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^8 & W_8^{10} & W_8^{12} & W_8^{14}\\ W_8^0 & W_8^3 & W_8^6 & W_8^9 & W_8^{12} & W_8^{15} & W_8^{18} & W_8^{21}\\ W_8^0 & W_8^4 & W_8^8 & W_8^{12} & W_8^{16} & W_8^{20} & W_8^{24} & W_8^{28}\\ W_8^0 & W_8^5 & W_8^{10} & W_8^{15} & W_8^{20} & W_8^{25} & W_8^{30} & W_8^{35}\\ W_8^0 & W_8^6 & W_8^{12} & W_8^{18} & W_8^{24} & W_8^{30} & W_8^{36} & W_8^{42}\\ W_8^0 & W_8^7 & W_8^{14} & W_8^{21} & W_8^{28} & W_8^{35} & W_8^{42} & W_8^{49}\end{bmatrix} \tag{3.9.3}$$

$$\implies \begin{bmatrix} H(0)\\ H(1)\\ H(2)\\ H(3)\\ H(4)\\ H(5)\\ H(6)\\ H(7)\end{bmatrix} = \begin{bmatrix} 1.32\\ 0.858-0.514j\\ -0.015-0.007j\\ 0.516+1.829j\\ 3.96\\ 0.516-1.829j\\ -0.015+0.007j\\ 0.858+0.514j\end{bmatrix} \tag{3.9.4}$$

3.10. The following is the C program to compute and print the FFT (N-point where N is of the form 2^n)

```
https://github.com/csrs2000/EE3025_IDP/
    blob/main/Assignment−1/code/
    ee18btech11007_1.c
```