EE3025 FFT and IFFT Implementation

Ritwik Sahani - EE18BTECH11038

Codes are available at

```
https://github.com/ritvix23/EE3025−IDP−DSP/tree/
    main/Assignment1_FFT/codes
```

To compile and run(on linux), navigate to the directory and execute-

```
gcc <filename>.c −o <filename> −lm && ./<
    filename>
```

1 Problem

Implement the Fast Fourier and Inverse Fast Fourier Transform algorithms in C to calculate DFT and Inverse DFT of a given sequence respectively. Verify using an example.

2 Method

The N-point Discrete Fourier Transform (DFT) for a given sequence $x[n]$ is defined by the formula -

$$X(k) = \sum_{n=0}^{N-1} x[n] W_N^{kn} \tag{2.0.1}$$

$$W_N = e^{\frac{-2\pi i}{N}} \tag{2.0.2}$$

Upon collecting the odd indices and the even indices together, we get -

$$X(k) = \sum_{r=0}^{\frac{N}{2}-1} x[2r] W_N^{(2r)k} + \sum_{r=0}^{\frac{N}{2}-1} x[2r+1] W_N^{(2r+1)k} \tag{2.0.3}$$

Define $e[r] = x[2r]$ and $o[r] = x[2r+1]$,
Upon substituting this in equation 2.0.3, we get

$$X(k) = \sum_{r=0}^{\frac{N}{2}-1} e[r] W_{N/2}^{kr} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} o[r] W_{N/2}^{kr} \tag{2.0.4}$$

At this point, we can exploit the following property of complex exponentials, along with peridocity, to get a recursive definition of $X(k)$ -

$$W_{\frac{N}{2}} = W_N^2 \tag{2.0.5}$$

which gives -

$$X(k) = E(k) + W_N^k O(k) \ \forall \text{ k in } [0, ..., N/2] \tag{2.0.6}$$

$$X(k) = E(k) - W_N^k O(k) \ \forall \text{ k in } [N/2+1, ..., N] \tag{2.0.7}$$

If we assume the input length to be a power of two, a recursive algorithm can be designed from equations 2.0.6 and 2.0.7 as follows -

Algorithm 1 FFT(x)

```
if $N > 1$ then
    $E = FFT$ (x[0],x[2]...,x[N-2])
    $O = FFT$ (x[1],x[3]...,x[N-1])
    for $k \longleftarrow 0$ to $\frac{N}{2} - 1$ do
        $x[k]= E[k] + e^{-2\pi jk/N} O[k]$
        $x[k + \frac{N}{2}]= E[k] - e^{-2\pi jk/N} O[k]$
    end for
end if
return x
```

The following C code first generates the DFT of the input and then performs IDFT on the resulting sequence. The given input is - $(0, 9, 1, 1, 2, 0, 0, 1)$. The output of both the operations are printed and stored in a DAT file.

```
https://github.com/ritvix23/EE3025−IDP−DSP/blob
    /main/Assignment1_FFT/codes/transform.c
```

The following python code reads the data from the DAT file and plots its frequency spectrum.

```
https://github.com/ritvix23/EE3025−IDP−DSP/blob
    /main/Assignment1_FFT/codes/plot.py
```

The spectrum of the DFT from the inbuilt routine in Python Numpy is plotted for verification using -

```
https://github.com/ritvix23/EE3025−IDP−DSP/blob
    /main/Assignment1_FFT/codes/verify.py
```

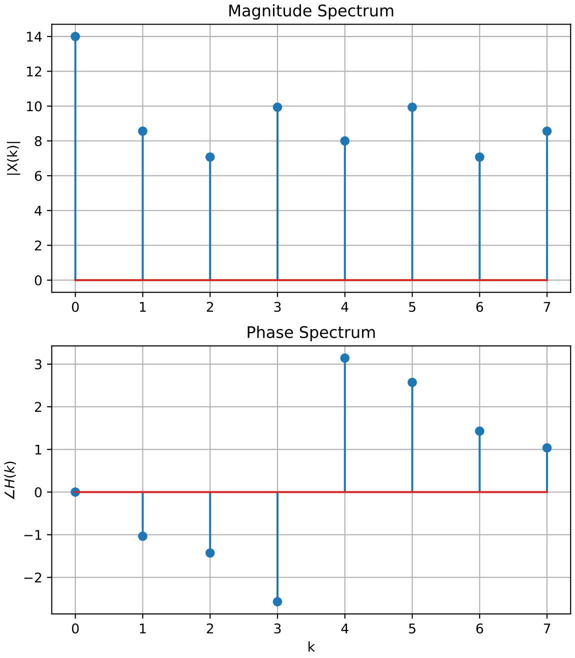

Fig. 0: From own routine

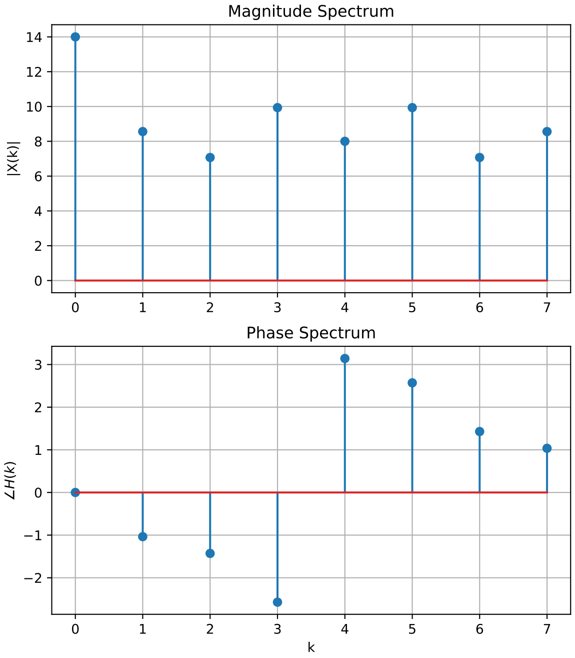

Fig. 0: From builtin routine