EXPERIMENT 2

DFT, IDFT, FFT and Parseval's Theorem

AIM

Part I: DFT and IDFT matrices:

(a) Generate and display DFT & IDFT matrices for a given length N

(b) Compute DFTs & IDFTs of sequences using the DFT and IDFT matrices

(c) Verify the results using MATLAB's inbuilt functions - fft and ifft

Part II: Direct computation of DFT and IDFT:

(a) Compute the DFT of a sequence by direct computation method without using the inbuilt function fft

(b) Compute the IDFT of the obtained DFT sequence without using the inbuilt ifft function

(c) Plot the magnitude and phase of the DFT coeffcients

(d) Verify the results using the MATLAB inbuilt functions - fft and ifft

Part III: Verification of Parseval's Theorem for DFT:

(a) Verify Parseval's theorem for DFT using two random sequences as inputs

THEORY

Discrete Fourier Transform(DFT)

The N-point DFT of a finite duration sequence $x[n]$ is denoted as $X[k]$ and is obtained by evaluating the expression,

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}; \text{ for } 0 \le k \le N-1$$

Similarly, the N-point IDFT of the finite duration sequence $X[k]$ is given by,

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}; \text{ for } 0 \le n \le N-1$$

Where W_N is a complex N-th root of unity given by $W_N = e^{-j2\pi/N}$

DFT and IDFT as linear transformations

The DFT and IDFT can be viewed as linear transformations on sequences $\{x[n]\}$ and $\{X[k]\}$ respectively.
The DFT analysis & synthesis equations can be written in matrix form as,

$$\mathbf{X} = \mathbf{W}_N \mathbf{x}$$

$$\mathbf{x} = \mathbf{W}_N^{-1} \mathbf{X}$$

Where,

- $\mathbf{x}$ is the $N \times 1$ matrix(Column Vector) corresponding to the sequence $\{x[n]\}$,
- $\mathbf{X}$ is the $N \times 1$ matrix(Column Vector) of DFT coefficients $\{X[k]\}$, and
- $$\mathbf{W}_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

  is the $N \times N$ matrix of linear transformation for N-point DFT. *i.e.*, the N-point DFT matrix.
- $\mathbf{W}_N^{-1}$ is the corresponding inverse DFT matrix

We may evaluate the IDFT matrix $\mathbf{W}_N^{-1}$ by using the relation,

$$\mathbf{W}_N^{-1} = \frac{\mathbf{W}_N^*}{N}$$

Example: DFT matrix for $N = 4$ is,

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

Fast Fourier Transforms(FFT)

Fast Fourier Transforms refer to an efficient class of algorithms used for digital computation of the DFT coefficients. Direct computation of DFT (or IDFT) coefficients are computationally intensive since they require N^2 complex multiplications and $N(N-1)$ complex additions where N dentotes the DFT-length. This translates to a total of $4N^2$ real multiplications and $N(4N-2)$ real additions.

FFT algorithms use a "divide and conquer" strategy to reduce the number of computations required by decomposing an N-point DFT into successively smaller DFTs. They exploit the following two properties of W_N^{kn}

- Symmetry: $W_N^{k[N-n]} = W_N^{-kn} = (W_N^{kn})^*$
- Periodicity in n and k: $W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n}$

Radix-2 FFT algorithms involves decomposing the input sequence into two smaller subsequences. It can either be Decimation in time or Decimation in frequency. For $N = 2^\gamma$ where γ is an integer, Radix-2 FFT will contain $v = log_2 N$ number of stages. Each stage will contain $N/2$ butterfly computations. Each butterfy requires 1 complex multiplication & 2 complex additions.

Hence the total number of complex multiplications needed for computing the N point DFT using Radix-2 FFT is $(N/2)log_2 N$ and the total number of complex additions required for the same is $Nlog_2 N$. This is much less than the computations needed for direct implementation

The signal flow graph representations for implementing DIT-r2 and DIF-r2 FFT algorithms are shown below.

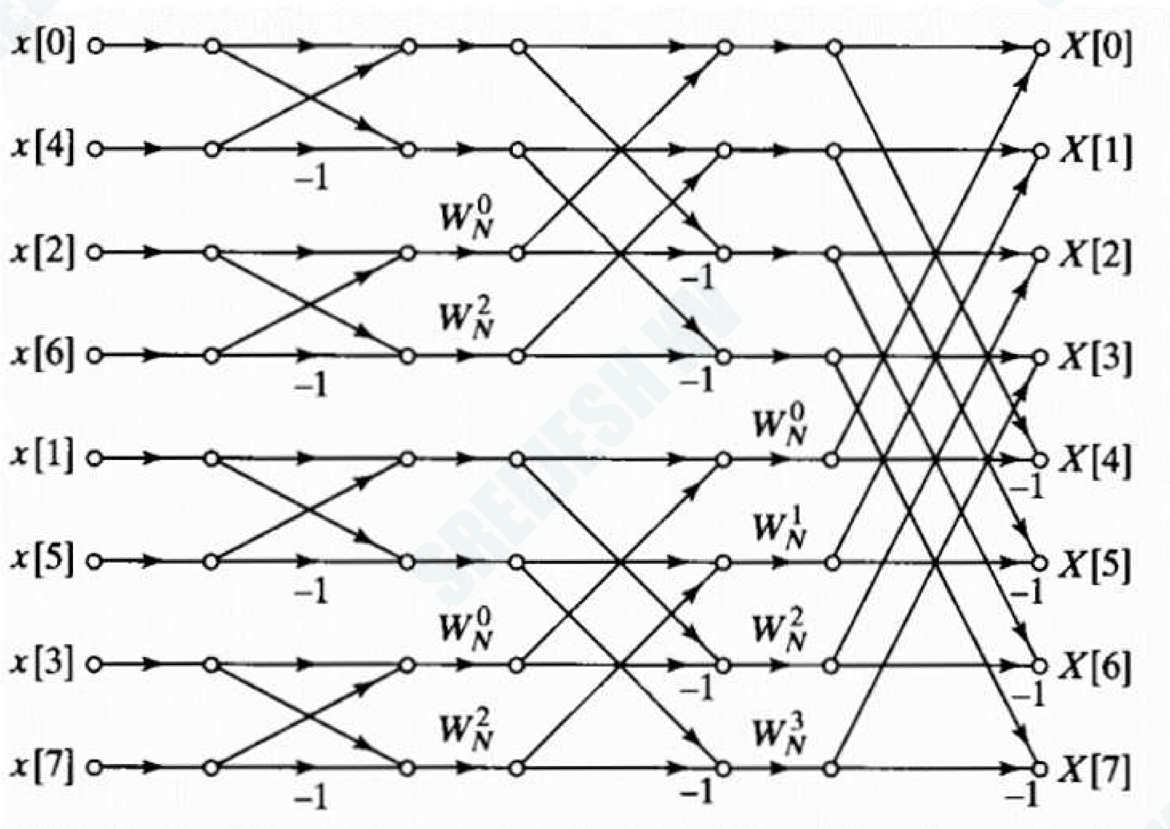


Figure 2.1: Flow graph for Decimation in Time Radix-2 FFT (N=8)

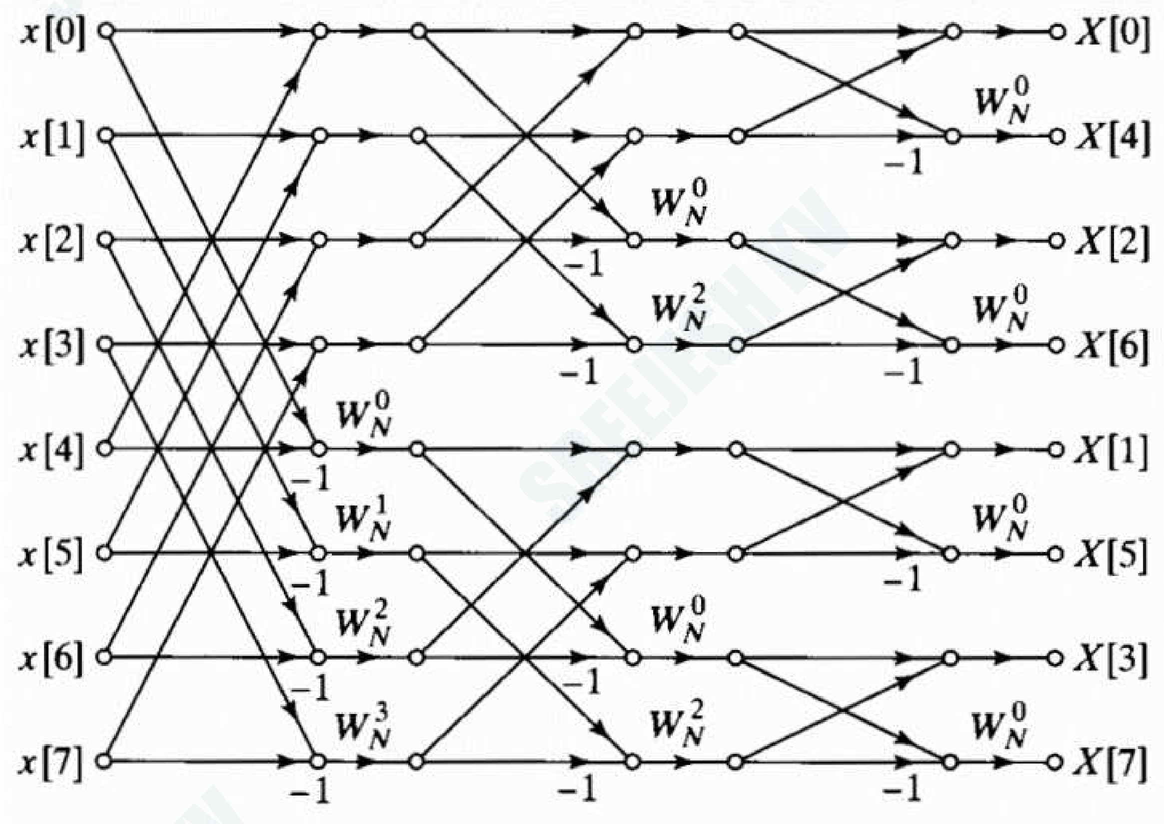


Figure 2.2: Flow graph for Decimation in Frequency Radix-2 FFT (N=8)

Parseval's Theorem for DFT coefficients

Parseval's theorem results from the fact that a unitary transformation preserves inner product (i.e., energy) DFT is (with appropriate selection of scaling) a unitary transform, i.e., one that preserves energy.

Mathematically,

Parseval's Theorem

if,

$$x[n] \xleftrightarrow[N]{DFT} X[k] \text{ and } g[n] \xleftrightarrow[N]{DFT} G[k]$$

Then,

$$\sum_{n=0}^{N-1} x[n]g^*[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]G^*[k]$$

Special Case: When $x[n] = g[n]$, we have,

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X[k]|^2$$

This implies that the energy of the finite duration sequence is the same when computed in either domain.

ALGORITHM

Part 1: DFT and IDFT matrices

Step 1. Start

Step 2. Prompt the user to enter the value of N and read it

Step 3. Generate the $N \times N$ DFT matrix $\mathbf{W}_N$ by evaluating $W_N^{ij} = e^{\frac{-j2\pi ij}{N}}$ as the ij-th element of the matrix

Step 4. Display the DFT matrix in command window

Step 5. Verify it by displaying the DFT matrix obtained using the inbuilt function dftmatx for the same length N

Step 6. Compute & Display the IDFT matrix by using the relation $\mathbf{W}_N^{-1} = \frac{\mathbf{W}_N^*}{N}$

Step 7. Prompt the user to enter an input sequence $\mathbf{x}$ of length N or less

Step 8. Compute the N point DFT $\mathbf{X}$ of the given sequence by using the matrix relation $\mathbf{X} = \mathbf{W}_N\mathbf{x}$

Step 9. Verify the result by using the inbuilt function fft

Step 10. Compute the inverse DFT of the obtained sequence $\mathbf{X}$ using the matrix relation $\mathbf{x} = \mathbf{W}_N^{-1}\mathbf{X}$

Step 11. Verify the result using the inbuilt function ifft

Part 2: Direct computation of DFT and IDFT

Step 1. Start

Step 2. Read the input sequence x[n]. For convenience of plotting use a real sequence

Step 3. Find the length of x[n] and assign it to N

Step 4. Calculate X[k] using the DFT analysis expression

Step 5. Plot the magnitude and the unwrapped phase of X[k] against k

Step 6. Use the IDFT equation to obtain the N-point IDFT of X[k] and plot the sequence obtained

Step 7. Use the MATLAB's inbuilt functions fft and ifft to verify the above result

Step 8. Stop

Part 3: Verification of Parseval's Theorem for DFT

Step 1. Start

Step 2. Read two random input sequences(complex in general)

Step 3. Evaluate the LHS and RHS of Parseval's relation separately

Step 4. Display and verify that the LHS and RHS are equal

Step 5. Stop

PROGRAM

Part 1: DFT and IDFT matrices:

```
%Title: Program to
% 1) Generate and display DFT & IDFT matrices for a given length N
% 2) Compute DFTs \& IDFTs of sequences using the DFT and IDFT ...
    matrices
% 3) Verify the results using MATLAB's inbuilt functions- fft and ...
    ifft


%Author: Sreejesh K V, Dept. of ECE, GCEK
%Date: 18/09/2022

clc;
clear;
close all;

N=input('Enter the value of N\n');%read the value of N from user

%-----generating the DFT matrix without using inbuilt function
w=zeros(N);%initialize the dft matrix with zeros
for k=0:N-1
for l=0:N-1
w(k+1,l+1)=cos((2*pi*k*l)/N)-1i*sin((2*pi*k*l)/N);
end
end

%-----displaying the generated DFT matrix in command window
disp(['The DFT matrix for order ' num2str(N) ' computed without ...
    using inbulit function is:']);
disp(w);

%------verification using inbuilt function
disp(['The DFT matrix for order ' num2str(N) ' using the inbuilt ...
    function dftmtx is:']);
a = dftmtx(N);
disp(a);
```

```
%-----generating the IDFT matrix without using inbuilt function
widft=conj(w)/N; %matrix for computing Nth order IDFT
disp(['The IDFT matrix for order ' num2str(N) ' is:']);
disp(widft);



%------computing N point DFT using the dftmatrix
x=input(['Enter the input sequence of length less than or equal ...
    to ' num2str(N) '\n']);%read the input sequence
x=[x zeros(1,N-length(x))];

X=w*x.';%transpose for converting to column vector
%Note: Using ' only(without the preceding dot) will give the ...
    conjugate transpose


disp('The DFT sequence computed using DFT matrix method is');
disp(X.');%transpose for displaying as row vector

%------verification using inbuilt function
disp('The DFT sequence computed using the inbuilt fft function is');
Xf=fft(x,N);
disp(Xf);


%------computing N point IDFT using the dftmatrix

idx=widft*X;
disp('The IDFT of the DFT of the given sequence computed using ...
    matrix method is');
disp(idx.');%transpose for displaying as row vector

%------verification of IDFT computation using inbuilt function
disp('The IDFT of the DFT of the given sequence computed using ...
    inbuilt function is');
idxf=ifft(X.',N);
disp(idxf);
```

Part 2: Direct computation of DFT and IDFT

```
%Title: Program to
% 1) Compute the DFT of a sequence without using the inbuilt ...
    fft functions
% 2) Compute the IDFT of the obtained DFT sequence without ...
    using the inbuilt ifft function
% 3) Plot the magnitude and phase of the DFT coeffcients
```

```
% 4) Verify the results using the MATLAB's inbuilt functions ...
    - fft and ifft

%Author: Sreejesh K V, Dept. of ECE, GCEK
%Date: 18/09/2022

clc;
clear;
close all;
%----------------------------------------------
x=[0 1 2 -3 2 -1];%random input sequence(may also be input by ...
    the user)
N=length(x); %Length of the sequence
X=zeros(1,N); %initialize DFT coefficients
ix=zeros(1,N); %initialize IDFT sequence

%------To plot the input sequence
n=0:N-1;
subplot(221);
stem(n,x);%assuming x[n] is real
ylabel ('x[n]');
xlabel ('n');
title('Input Sequence x[n]');

%---To Find the DFT of the sequence---%
for k=0:N-1
for n=0:N-1
X(k+1)=X(k+1)+(x(n+1)*exp((-1i)*2*pi*k*n/N));
end
end

%------To plot the DFT magnitude response------------%
n=0:N-1;
magnitude=abs(X); % The magnitudes of DFT coefficients
subplot(222);
stem(n,magnitude);
ylabel ('|X[k]|');
xlabel ('k');
title('Magnitude of DFT coefficients');

%------To plot the DFT phase response------------%
n=0:N-1;
phase=unwrap(angle(X)); % Find the phases of individual DFT ...
    points
subplot(223);
stem(n,radtodeg(phase));
ylabel ('<X[k] in degrees');
xlabel ('k');
title('Phase of DFT coefficients in degrees');

%------To find the IDFT of the sequence X[k]----------%
```

```
for n=0:N-1
for k=0:N-1
ix(n+1)=ix(n+1)+(X(k+1)*exp(1i*2*pi*k*n/N));
end
end
ix=ix./N;

%-----To plot the IDFT sequence----%
n=0:N-1;
subplot(224);
stem(n,ix);%assuming ix[n] is real
ylabel ('ix[n]');
xlabel ('n');
title('The IDFT sequence ix[n]');

%---- To verify the results using inbuilt functions---%
disp('The DFT coefficient values obtained without using ...
    inbuilt function are: ');
disp(X);

disp('The DFT coefficient values computed using inbuilt fft ...
    function are: ');
X1=fft(x);
disp(X1);

disp('The IDFT coefficient values obtained without using ...
    inbuilt function are: ');
disp(ix);
disp('The IDFT coefficient values computed using inbuilt fft ...
    function are: ');
ix1=ifft(X);
disp(ix1);
```

Part 3: Verification of Parseval's Theorem for DFT

```
%Title: Program to verify Parseval's Theorem for DFT coefficients
%Author: Sreejesh K V, Dept. of ECE, GCEK
%Date: 18/09/2022
clc;
clear;
close all;


x=[1+3i, 2, 2i, -1 , -2-2i, 7+5i];%random complex input ...
    sequence 1
y=[2i, 12-1i, 2, -5 , 3i, 8, 6-3i];%random complex input ...
    sequence 2 (different length)

N = max(length(x),length(y));%Length of DFT
```

```
xpad=[x zeros(1,N-length(x))];%Padding with zeros to make ...
    both sequences have the same length
ypad=[y zeros(1,N-length(y))];%Padding with zeros to make ...
    both sequences have the same length

lhs=sum(xpad.*conj(ypad));%The LHS of Parseval's theorem ...
    expression

X = fft(xpad,N);
Y = fft(ypad,N);
rhs=sum(X.*conj(Y))/N;%The RHS of Parseval's theorem expression

%displaying both LHS and RHS to show that Parseval's relation ...
    holds
disp('LHS of Parseval''s expression='); disp(lhs);
disp('RHS of Parseval''s expression='); disp(rhs);
```

OUTPUT & OBSERVATIONS

Part 1: DFT and IDFT matrices:

Command Window Output:

```
Enter the value of N
4
The DFT matrix for order 4 computed without using inbulit function is:
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 + 1.0000i
   1.0000 + 0.0000i  -1.0000 - 0.0000i   1.0000 + 0.0000i  -1.0000 - 0.0000i
   1.0000 + 0.0000i  -0.0000 + 1.0000i  -1.0000 - 0.0000i   0.0000 - 1.0000i

The DFT matrix for order 4 using the inbuilt function dftmtx is:
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 + 0.0000i   0.0000 + 1.0000i
   1.0000 + 0.0000i  -1.0000 + 0.0000i   1.0000 + 0.0000i  -1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 + 1.0000i  -1.0000 + 0.0000i   0.0000 - 1.0000i

The IDFT matrix for order 4 is:
   0.2500 + 0.0000i   0.2500 + 0.0000i   0.2500 + 0.0000i   0.2500 + 0.0000i
   0.2500 + 0.0000i   0.0000 + 0.2500i  -0.2500 + 0.0000i  -0.0000 - 0.2500i
   0.2500 + 0.0000i  -0.2500 + 0.0000i   0.2500 - 0.0000i  -0.2500 + 0.0000i
   0.2500 + 0.0000i  -0.0000 - 0.2500i  -0.2500 + 0.0000i   0.0000 + 0.2500i
```

```
Enter the input sequence of length less than or equal to 4
[1 2 3 4]
The DFT sequence computed using DFT matrix method is
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 - 0.0000i  -2.0000 - 2.0000i

The DFT sequence computed using the inbuilt fft function is
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 + 0.0000i  -2.0000 - 2.0000i

The IDFT of the DFT of the given sequence computed using matrix method is
   1.0000 - 0.0000i   2.0000 - 0.0000i   3.0000 - 0.0000i   4.0000 + 0.0000i

The IDFT of the DFT of the given sequence computed using inbuilt function is
   1.0000 - 0.0000i   2.0000 - 0.0000i   3.0000 + 0.0000i   4.0000 + 0.0000i
```

Part 2: Direct computation of DFT and IDFT

Command Window Output:

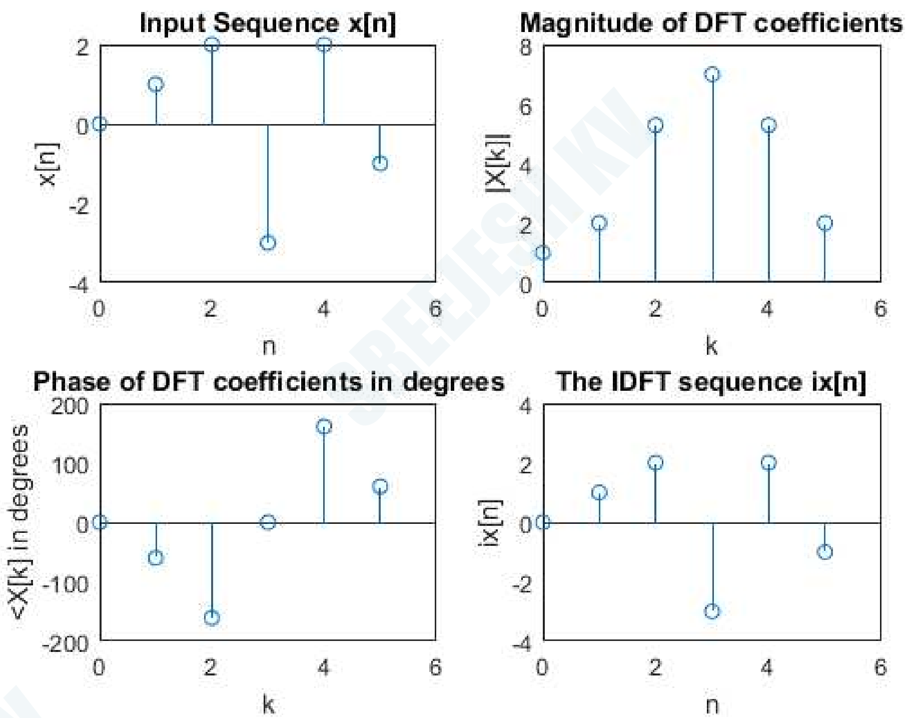
```
The DFT coefficient values obtained without using inbuilt function are:
   1.0000 + 0.0000i   1.0000 - 1.7321i  -5.0000 - 1.7321i   7.0000 + 0.0000i  -5.0000 + 1.7321i   1.0000 + 1.7321i

The DFT coefficient values computed using inbuilt fft function are:
   1.0000 + 0.0000i   1.0000 - 1.7321i  -5.0000 - 1.7321i   7.0000 + 0.0000i  -5.0000 + 1.7321i   1.0000 + 1.7321i

The IDFT coefficient values obtained without using inbuilt function are:
  -0.0000 + 0.0000i   1.0000 + 0.0000i   2.0000 - 0.0000i  -3.0000 - 0.0000i   2.0000 - 0.0000i  -1.0000 + 0.0000i

The IDFT coefficient values computed using inbuilt fft function are:
   0.0000 + 0.0000i   1.0000 + 0.0000i   2.0000 + 0.0000i  -3.0000 - 0.0000i   2.0000 + 0.0000i  -1.0000 + 0.0000i
```

Figure Window Output:



Part 3: Verification of Parseval's Theorem for DFT

Command Window Output:

```
Command Window
LHS of Parseval's expression=
  85.0000 +50.0000i

RHS of Parseval's expression=
  85.0000 +50.0000i

fx >>
```

RESULTS

Part I: DFT and IDFT matrices:

(a) DFT & IDFT matrices for a given length N are generated and displayed

(b) The DFTs & IDFTs of given sequences were evaluated using the DFT and IDFT matrices

(c) Above results were verified using MATLAB's inbuilt functions - fft and ifft

Part II: Direct computation of DFT and IDFT:

(a) The DFT of a given sequence was evaluated by direct computation method without using the inbuilt function fft

(b) The IDFT of the obtained DFT sequence was computed without using the inbuilt ifft function

(c) The magnitude and phase of the DFT coeffcients were plotted and observed

(d) The results were verified using the MATLAB inbuilt functions - fft and ifft

Part III: Verification of Parseval's Theorem for DFT:

(a) Parseval's theorem for DFT coeffecients was verified using two random sequences as inputs