Sreenivasa Institute of Technology and Management Studies [SITAMS]
[Autonomous]
Chittoor, Andhra Pradesh

Department of Electronics and Communication Engineering

23ECE361T – Digital Signal Processing

Unit II - Discrete Fourier Transform & Fast Fourier Transform

Introduction, Discrete Fourier Series, properties of DFS, Discrete Fourier Transform, Inverse DFT, properties of DFT, Linear and Circular convolution, convolution using DFT, sampling, Quantization effects. Introduction, Fast Fourier Transform, Radix-2 Decimation in time and Decimation in frequency FFT, Inverse FFT (Radix-2).

1.Introduction to Discrete Fourier Series (DFS):

The Discrete Fourier Series(DFS) represents a periodic discrete-time signal as a sum of complex exponential signals at discrete frequencies.

If x[n] is periodic with fundamental period N, then x[n+N]=x[n]

The DFS representation is

$$x[n] = \sum_{k=0}^{N-1} X[k] e^{j2\pi kn/N}$$

where X[k] represents the DFS coefficients.

The analysis equation is

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

Here:

- $x[n]$: periodic discrete-time signal.
- $X[k]$: DFS coefficient for frequency index k.
- N: fundamental period.
- $j = \sqrt{-1}$.

The DFS coefficients are periodic in k, with period N.

Example:

Consider a periodic signal with one period: x[n]=[1,1,1,1], N=4

Using the analysis equation,

$$X[k] = \frac{1}{4} \sum_{n=0}^{3} x[n] e^{-j2\pi kn/4}$$

For k=0,

X[0]=1/4(1+1+1+1)=1

For k=1,2,3, the complex exponential terms sum to zero. Therefore,

X[k]=[1,0,0,0]

The signal is constant, so only the zero-frequency (DC) coefficient is nonzero.

2.Properties of DFS

Let x[n] < --- > X[k] and y[n] < -- > Y[k], using the DFS convention above.

Property	Time-domain operation	DFS coefficients
Linearity	$ax[n] + by[n]$	$aX[k] + bY[k]$
Time shift	$x[n - n_0]$	$e^{-j2\pi k n_0/N} X[k]$
Frequency shift	$e^{j2\pi mn/N} x[n]$	$X[(k - m) \bmod N]$
Time reversal	$x[-n]$	$X[(-k) \bmod N]$
Conjugation	$x^*[n]$	$X^*[(-k) \bmod N]$
Periodic convolution	$x[n] \circledast y[n]$	$NX[k]Y[k]$
Multiplication	$x[n]y[n]$	$\sum_{r=0}^{N-1} X[r]Y[(k - r) \bmod N]$

Important: The factor N in the convolution property follows from the chosen DFS convention, where the synthesis equation has no $1/N$ factor and the analysis equation has $1/N$.

3. Discrete Fourier Transform (DFT)

The Discrete Fourier Transform converts a finite-length discrete-time sequence into its frequency-domain representation.

For a sequence x[n] of length N, the DFT is

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

where k=0,1,…..,N-1.

The DFT provides N frequency-domain samples of the signal.

DFT in matrix form

$$X = W_N x$$

where x is the input vector, X is the DFT vector, and the elements of the DFT matrix are

$$W_N^{kn} = e^{-j2\pi kn/N}$$

Example: 4-point DFT

Given: x[n]=[1,2,3,4]

The 4-point DFT is

$$X[k] = \sum_{n=0}^{3} x[n] e^{-j2\pi kn/4}$$

For k=0,

X[0]=1+2+3+4=10

For k=1,

X[1]=1+2(-j)+3(-1)+4(j)
X[1]=-2+2j

For k=2,

X[2]=1-2+3-4=-2

For k=3,

X[3]=-2-2j

Therefore, X[k]=[10,-2+2j,-2,-2-2j]

Because the input sequence is real-valued, the DFT has conjugate symmetry: X[3]=X*[1].

4.Inverse Discrete Fourier Transform (IDFT)

The Inverse Discrete Fourier Transform converts frequency-domain DFT coefficients back into the original time-domain sequence.

The IDFT equation is

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi kn/N}$$

where n=0,1,…..,N-1.

The factor 1/N is essential in this DFT/IDFT convention.

Example

Suppose the 2-point DFT coefficients are

X[0]=3,X[1]=-1

Using the IDFT equation with N=2:

For n=0,

x[0]=1/2[3+(-1)]=1

For n=1,

x[1]=1/2[3+(-1)$e^{j\pi}$]=2

Hence,

x[n]=[1,2]

The IDFT recovers the original time-domain sequence.

5. Properties of DFT

For an N-point DFT, let x[n] < - - - > X[k].

Property	Time-domain operation	DFT result
Linearity	$ax[n] + by[n]$	$aX[k] + bY[k]$
Circular time shift	$x[(n - n_0) \bmod N]$	$e^{-j2\pi k n_0/N} X[k]$
Frequency shift	$e^{j2\pi mn/N} x[n]$	$X[(k - m) \bmod N]$
Time reversal	$x[(-n) \bmod N]$	$X[(-k) \bmod N]$
Conjugation	$x^*[n]$	$X^*[(-k) \bmod N]$
Circular convolution	$x[n] \circledast y[n]$	$X[k]Y[k]$
Pointwise multiplication	$x[n]y[n]$	$\frac{1}{N} \sum_{r=0}^{N-1} X[r]Y[(k - r) \bmod N]$
Parseval's relation	Signal energy	$\sum \lvert x[n]\rvert^2 = \frac{1}{N} \sum \lvert X[k]\rvert^2$

DFT symmetry for real signals

If x[n] is real-valued, then

$$\boxed{X[N - k] = X^*[k]}$$

for the applicable indices interpreted modulo N.

Consequences:

- The magnitude spectrum is symmetric.
- The phase spectrum is antisymmetric, subject to phase wrapping.
- The DC coefficient is real.
- For even N, the Nyquist coefficient X[N/2] is also real.

Parseval's theorem

Parseval's theorem states that the signal energy in the time domain equals the appropriately scaled energy in the frequency domain:

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

This property is useful for verifying DFT calculations and comparing signal energy in both domains.

1. Linear Convolution

1.1 Definition

Linear convolution is a mathematical operation used to determine the output of a discrete-time Linear Time-Invariant (LTI) system from its input signal and impulse response.

If $x[n]$ is the input signal and $h[n]$ is the impulse response, the output $y[n]$ is

$$\boxed{y[n] = x[n] * h[n]}$$

The linear convolution equation is

$$\boxed{y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]}$$

For finite-length sequences, the sum includes only the nonzero terms.

1.2 Length of linear convolution

If the input sequence has length L and the impulse response has length M, then the output length is

$$\boxed{N = L + M - 1}$$

1.3 Worked example

Given:

$$x[n] = [1, 2, 3], \qquad h[n] = [1, 1]$$

Calculate $y[n] = x[n] * h[n]$.

Step 1: Calculate the output length.

$$N = 3 + 2 - 1 = 4$$

Step 2: Calculate each output sample.

$$y[0] = 1(1) = 1$$

$$y[1] = 1(1) + 2(1) = 3$$

$$y[2] = 2(1) + 3(1) = 5$$

$$y[3] = 3(1) = 3$$

Therefore,

$$\boxed{y[n] = [1, 3, 5, 3]}$$

Applications: Digital filtering, system response calculation, echo processing, and signal analysis.

2. Circular Convolution

2.1 Definition

Circular convolution is used to calculate the convolution of two finite-length sequences under the assumption that they are periodic with the same period N.

For two N-point sequences $x[n]$ and $h[n]$, the circular convolution is

$$\boxed{y[n] = \sum_{k=0}^{N-1} x[k]h[(n-k) \bmod N]}$$

The modulo-N operation wraps sequence indices around the beginning when they exceed the range 0 to $N-1$.

2.2 Worked example

Given two 3-point sequences:

$$x[n] = [1, 2, 3]$$

$$h[n] = [1, 1, 0]$$

Calculate the 3-point circular convolution.

For $n = 0$,

$$y[0] = x[0]h[0] + x[1]h[2] + x[2]h[1]$$

$$y[0] = 1(1) + 2(0) + 3(1) = 4$$

For $n = 1$,

$$y[1] = x[0]h[1] + x[1]h[0] + x[2]h[2]$$

$$y[1] = 1(1) + 2(1) + 3(0) = 3$$

For $n = 2$,

$$y[2] = x[0]h[2] + x[1]h[1] + x[2]h[0]$$

$$y[2] = 1(0) + 2(1) + 3(1) = 5$$

Hence,

$$\boxed{y[n] = [4, 3, 5]}$$

3. Linear Convolution vs. Circular Convolution

Feature	Linear convolution	Circular convolution
Signal assumption	Finite-length sequences	Periodic sequences of length N
Output length	$L+M-1$	N, for N-point sequences
Index handling	No wraparound	Indices wrap around modulo N
Main application	LTI system response	DFT-based signal processing
Aliasing in convolution	No circular wraparound	May produce time-domain wraparound
Relationship	Standard convolution	Equals linear convolution when sequences are appropriately zero-padded

4. Convolution Using DFT

4.1 Principle

The DFT converts time-domain sequences into frequency-domain sequences. The circular convolution property states that circular convolution in the time domain corresponds to multiplication in the frequency domain.

$$\boxed{x[n] \circledast h[n] \;\longleftrightarrow\; X[k]H[k]}$$

Thus, circular convolution can be calculated using the following steps:

Input sequences $x[n], h[n]$

↓

Calculate DFTs $X[k], H[k]$

↓

Multiply: $Y[k] = X[k]H[k]$

↓

Calculate IDFT to obtain $y[n]$

4.2 Procedure for linear convolution using DFT

To obtain **linear convolution** using the DFT, zero-pad both sequences to a common length N satisfying

$$\boxed{N \geq L + M - 1}$$

Then calculate

$$X[k] = \mathrm{DFT}\{x[n]\}$$

$$H[k] = \mathrm{DFT}\{h[n]\}$$

Multiply the spectra:

$$Y[k] = X[k]H[k]$$

Finally, calculate the IDFT:

$$\boxed{y[n] = \mathrm{IDFT}\{Y[k]\}}$$

Zero-padding prevents circular wraparound from corrupting the desired linear-convolution result.

4.3 Worked example using DFT

Let

$$x[n] = [1, 2], \qquad h[n] = [1, 1]$$

The required output length is

$$N = 2 + 2 - 1 = 3$$

Zero-pad both sequences to length 3:

$$x[n] = [1, 2, 0], \qquad h[n] = [1, 1, 0]$$

Using the 3-point DFT, define $W = e^{-j2\pi/3}$.

$$X[0] = 3, \quad X[1] = 1 + 2W, \quad X[2] = 1 + 2W^2$$

$$H[0] = 2, \quad H[1] = 1 + W, \quad H[2] = 1 + W^2$$

Multiply corresponding DFT values:

$$Y[k] = X[k]H[k]$$

Taking the 3-point IDFT gives

$$\boxed{y[n] = [1, 3, 2]}$$

This is the same result obtained by direct linear convolution.

5. Sampling

5.1 Definition

Sampling is the process of converting a continuous-time signal into a discrete-time sequence by measuring its value at uniform time intervals.

If $x_a(t)$ is a continuous-time signal and T_s is the sampling period, the sampled signal is

$$\boxed{x[n] = x_a(nT_s)}$$

The sampling frequency is

$$\boxed{f_s = \frac{1}{T_s}}$$

where:

- f_s: sampling frequency in hertz (Hz).
- T_s: sampling period in seconds.

5.2 Sampling theorem

The Nyquist–Shannon sampling theorem states that a band-limited signal with maximum frequency $f_{\max}$ can theoretically be reconstructed from its samples if

$$\boxed{f_s \geq 2f_{\max}}$$

The minimum theoretical sampling frequency $2f_{\max}$ is called the **Nyquist rate**.

In practical systems, sampling is generally performed above the Nyquist rate to allow for real filter transition bands and implementation tolerances.

5.3 Worked example

A signal contains frequency components up to $4\,\mathrm{kHz}$.

Calculate its Nyquist rate.

Given:

$$f_{\max} = 4\,\mathrm{kHz}$$

Using the sampling theorem:

$$f_s \geq 2f_{\max}$$

$$f_s \geq 2(4) = 8\,\mathrm{kHz}$$

Therefore, the theoretical minimum sampling frequency is

$$\boxed{f_s = 8\,\mathrm{kHz}}$$

In practice, a higher sampling frequency may be selected.

5.4 Aliasing

Aliasing occurs when a signal is sampled below the required rate, causing different frequency components to become indistinguishable in the sampled signal.

Effects of aliasing:

- Distortion in the reconstructed signal.
- Incorrect frequency measurement.
- Loss of information about the original signal.

Prevention: Use an anti-aliasing low-pass filter before the analog-to-digital converter and choose an appropriate sampling frequency.

5.5 Types of sampling

1. **Ideal sampling:** Samples are represented by impulses at uniform time intervals.
2. **Natural sampling:** The signal is sampled through periodic pulses whose tops follow the input waveform.
3. **Flat-top sampling:** Each sample is held at a constant amplitude for a short duration.

6. Quantization Effects

6.1 Definition

Quantization is the process of mapping each sampled signal amplitude to one of a finite number of allowed amplitude levels.

Sampling discretizes time, while quantization discretizes amplitude.

In an N_b-bit uniform quantizer, the number of possible quantization levels is

$$\boxed{L = 2^{N_b}}$$

where N_b is the number of bits per sample.

For a uniform quantizer covering a total amplitude range $V_{\max} - V_{\min}$, the approximate step size is

$$\boxed{\Delta = \frac{V_{\max} - V_{\min}}{L}}$$

The exact step-size convention can vary by quantizer design.

6.2 Quantization error

Quantization error is the difference between the actual sampled value and its quantized value.

$$\boxed{e_q[n] = x[n] - x_q[n]}$$

where $x[n]$ is the original sample and $x_q[n]$ is the quantized sample.

For an ideal uniform rounding quantizer that is not overloaded,

$$\boxed{-\frac{\Delta}{2} \le e_q[n] \le \frac{\Delta}{2}}$$

The error is bounded by half a quantization step in either direction.

6.3 Worked example

An ADC uses 3 bits and covers the range from 0 V to 8 V.

Step 1: Calculate the number of levels.

$$L = 2^3 = 8$$

Step 2: Calculate the quantization step size.

$$\Delta = \frac{8 - 0}{8} = 1\,\mathrm{V}$$

Step 3: Consider a sample of 3.4 V. Under a nearest-level convention with representative levels $0.5, 1.5, \ldots, 7.5$ V, the quantized value is 3.5 V.

$$e_q = 3.4 - 3.5 = -0.1\,\mathrm{V}$$

Therefore, the quantization error is

$$\boxed{e_q = -0.1\,\mathrm{V}}$$

This example uses a mid-rise quantizer. Other ADC coding and endpoint conventions may produce different representative levels.

6.4 Quantization noise

Quantization error behaves like noise in many signal-processing applications. Under the common model of a uniform quantizer with sufficiently varied input and no overload, the error is approximated as uniformly distributed between $-\Delta/2$ and $+\Delta/2$.

The mean-square quantization error is then

$$\boxed{\sigma_q^2 = \frac{\Delta^2}{12}}$$

The corresponding RMS quantization noise is

$$\boxed{\sigma_q = \frac{\Delta}{\sqrt{12}}}$$

These are model-based approximations, not universal guarantees for every input signal.

6.5 Signal-to-quantization-noise ratio (SQNR)

For an ideal N_b-bit uniform quantizer and a full-scale sinusoidal input, the approximate SQNR is

$$\boxed{\mathrm{SQNR} \approx 6.02 N_b + 1.76 \text{ dB}}$$

This formula assumes an ideal ADC, a full-scale sine wave, and quantization noise approximations.

For an 8-bit ADC:

$$\mathrm{SQNR} \approx 6.02(8) + 1.76$$

$$\boxed{\mathrm{SQNR} \approx 49.92 \text{ dB}}$$

Increasing the number of bits generally improves amplitude resolution and reduces quantization noise, provided the signal stays within the ADC input range.

6.6 Effects of quantization

- **Quantization noise:** Small differences between the original and quantized values.
- **Rounding or truncation error:** Results from the quantization rule.
- **Overload distortion:** Occurs when the input exceeds the quantizer's supported range.
- **Reduced dynamic range:** Low-level signals may be represented with limited precision.
- **Improved accuracy with more bits:** More bits generally provide finer amplitude resolution.

1. Fast Fourier Transform (FFT)

Definition

The **Fast Fourier Transform (FFT)** is an efficient algorithm used to calculate the Discrete Fourier Transform (DFT) and its inverse. It reduces the number of computations required to obtain the frequency-domain representation of a discrete-time signal.

The DFT is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

where

$$W_N = e^{-j\frac{2\pi}{N}}$$

Here:

- $x(n)$: input sequence
- $X(k)$: DFT output
- N: number of samples
- W_N: twiddle factor
- $j = \sqrt{-1}$

Need for FFT

Method	Approximate computational complexity
Direct DFT	$O(N^2)$
Radix-2 FFT	$O(N \log_2 N)$

For $N = 1024$, direct DFT requires approximately $1,048,576$ complex multiplications in a straightforward implementation, whereas a radix-2 FFT uses 5120 butterfly operations, each involving a twiddle-factor multiplication and additions.

Twiddle factor properties

$$W_N^k = e^{-j2\pi k/N}$$

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

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

The radix-2 FFT requires $N = 2^m$, where m is a positive integer.

2. Radix-2 Decimation-in-Time (DIT) FFT

Definition

In the **DIT-FFT algorithm**, the input sequence is divided into even-indexed and odd-indexed samples. The DFT is computed recursively using smaller DFTs.

The input sequence is decomposed as

$$x(2r), \qquad x(2r+1)$$

The DFT can be written as

$$X(k) = E(k) + W_N^k O(k)$$

where:

- $E(k)$: DFT of the even-indexed samples
- $O(k)$: DFT of the odd-indexed samples

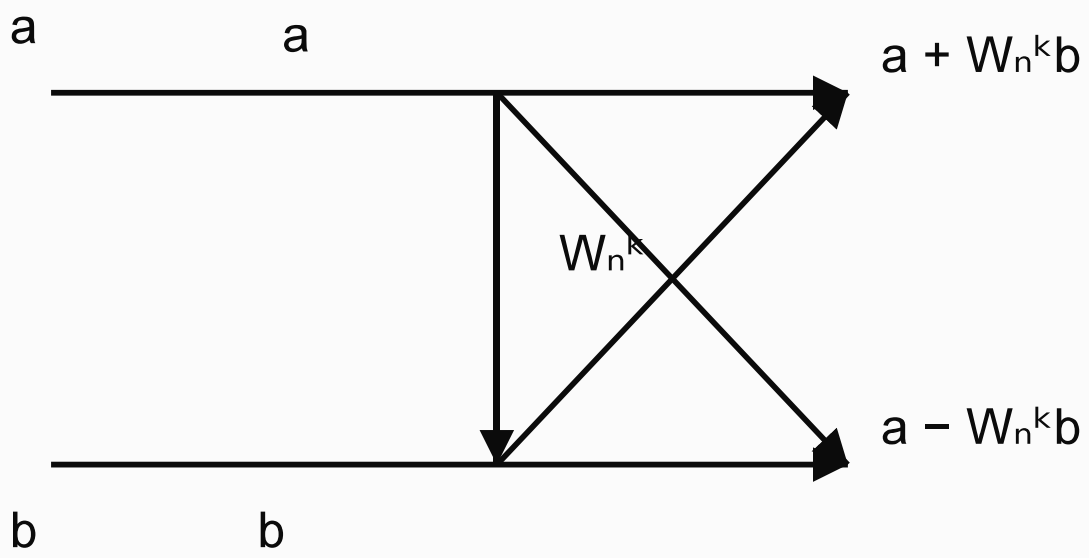
The second half of the output is

$$X(k + N/2) = E(k) - W_N^k O(k)$$

for $k = 0, 1, \ldots, N/2 - 1$.

DIT butterfly operation

Radix-2 DIT butterfly



The butterfly combines two input values to produce two output values.

For input values a and b,

$$y_1 = a + W_N^k b$$

$$y_2 = a - W_N^k b$$

Numerical example: 4-point DIT-FFT

Given

$$x(n) = \{1, 2, 3, 4\}$$

The 4-point DFT is

$$X(k) = \sum_{n=0}^{3} x(n) W_4^{nk}$$

Since $W_4 = e^{-j\pi/2} = -j$, split the sequence into even and odd samples:

- Even samples: $\{1, 3\}$
- Odd samples: $\{2, 4\}$

Their 2-point DFTs are

$$E(0) = 1 + 3 = 4, \quad E(1) = 1 - 3 = -2$$

$$O(0) = 2 + 4 = 6, \quad O(1) = 2 - 4 = -2$$

Now combine the results:

$$X(0) = E(0) + O(0) = 10$$

$$X(2) = E(0) - O(0) = -2$$

$$X(1) = E(1) + W_4 O(1) = -2 + (-j)(-2) = -2 + 2j$$

$$X(3) = E(1) - W_4 O(1) = -2 - 2j$$

Final output:

$$\boxed{X(k) = \{10, -2 + 2j, -2, -2 - 2j\}}$$

Important characteristics of DIT-FFT

- The sequence is divided by time-index parity.
- The input commonly uses bit-reversed ordering for an in-place iterative implementation.
- The output is in natural order in that implementation.
- Each stage contains $N/2$ butterflies.
- Number of stages: $\log_2 N$.

3. Radix-2 Decimation-in-Frequency (DIF) FFT

Definition

In the **DIF-FFT algorithm**, the input sequence is divided into sums and differences of samples separated by $N/2$. The decomposition is applied to the output-frequency calculation.

For $n = 0, 1, \ldots, N/2 - 1$, form

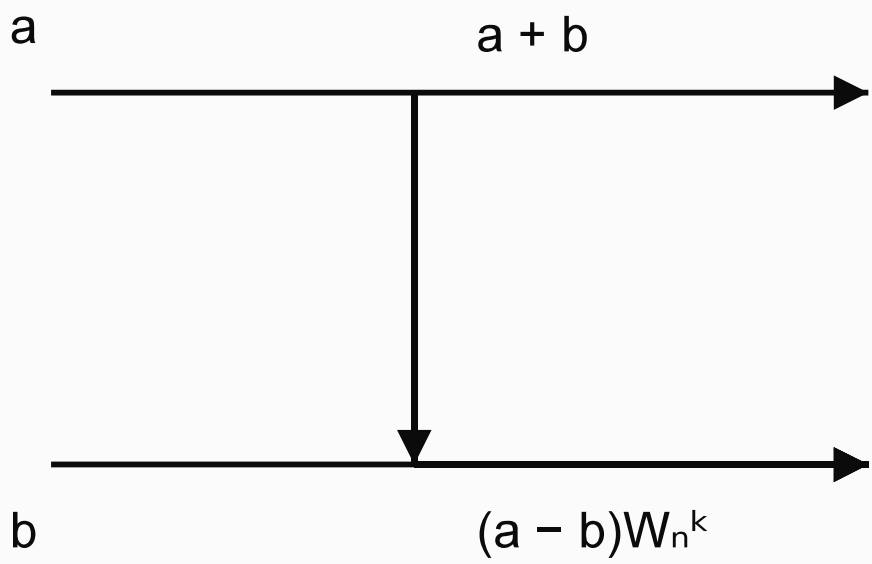
$$g(n) = x(n) + x(n + N/2)$$

$$h(n) = [x(n) - x(n + N/2)]\, W_N^n$$

The DFT is then calculated recursively from $g(n)$ and $h(n)$, each of length $N/2$.

DIF butterfly operation

Radix-2 DIF butterfly



The upper output is the sum; the lower output is the difference multiplied by a twiddle factor.

The butterfly equations are

$$y_1 = a + b$$

$$y_2 = (a - b)W_N^k$$

Numerical example: 4-point DIF-FFT

Given

$$x(n) = \{1, 2, 3, 4\}$$

Step 1: First-stage butterflies

$$g(0) = x(0) + x(2) = 1 + 3 = 4$$

$$g(1) = x(1) + x(3) = 2 + 4 = 6$$

$$h(0) = [1 - 3]W_4^0 = -2$$

$$h(1) = [2 - 4]W_4^1 = (-2)(-j) = 2j$$

Thus,

$$g = \{4, 6\}, \qquad h = \{-2, 2j\}$$

Step 2: Calculate the 2-point DFTs

For g:

$$G(0) = 4 + 6 = 10, \quad G(1) = 4 - 6 = -2$$

For h:

$$H(0) = -2 + 2j, \quad H(1) = -2 - 2j$$

Place the results in DIF output order:

$$\boxed{X(k) = \{10, -2, -2 + 2j, -2 - 2j\}}$$

The same DFT values are obtained as with DIT, but the ordering differs in the intermediate and final arrays.

Important characteristics of DIF-FFT

- The input is in natural order.
- The output is commonly in bit-reversed order for an in-place iterative implementation.
- The first stage combines samples separated by $N/2$.
- Each stage contains $N/2$ butterflies.
- Number of stages: $\log_2 N$.

4. DIT-FFT vs DIF-FFT

Feature	DIT-FFT	DIF-FFT
Full form	Decimation-in-Time	Decimation-in-Frequency
Decomposition	Even and odd time samples	Sums and differences of separated samples
Typical input order	Bit-reversed	Natural
Typical output order	Natural	Bit-reversed
Butterfly form	Twiddle multiplication before sum/difference	Sum/difference before twiddle multiplication
Complexity	$O(N \log_2 N)$	$O(N \log_2 N)$
DFT result	Same	Same

Note: Input and output ordering depends on the implementation. The table describes the common in-place radix-2 algorithms.

5. Inverse FFT (Radix-2)

Definition

The **Inverse Fast Fourier Transform (IFFT)** converts frequency-domain samples back into the time-domain sequence.

The inverse DFT is

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk}$$

Since $W_N = e^{-j2\pi/N}$, the inverse uses the opposite exponential sign.

Radix-2 IFFT procedure

1. Take the N frequency-domain input samples $X(k)$.

2. Apply a radix-2 FFT structure with the sign of the twiddle-factor exponent reversed.
3. Use $W_N^{-k} = e^{j2\pi k/N}$ in place of the forward transform's W_N^k.
4. Complete the butterfly stages.
5. Divide every final sample by N.

For an inverse butterfly,

$$y_1 = a + W_N^{-k} b$$

$$y_2 = a - W_N^{-k} b$$

The output is then scaled by $1/N$.

Numerical example: 4-point IFFT

Suppose the frequency-domain sequence is

$$X(k) = \{10, -2+2j, -2, -2-2j\}$$

Using the inverse DFT,

$$x(n) = \frac{1}{4} \sum_{k=0}^{3} X(k) W_4^{-nk}$$

For $n = 0$,

$$x(0) = \frac{1}{4}[10 + (-2+2j) - 2 + (-2-2j)] = \frac{4}{4} = 1$$

For $n = 1$,

$$x(1) = \frac{1}{4}[10 + (-2+2j)j + (-2)(-1) + (-2-2j)(-j)] = 2$$

Similarly,

$$x(2) = 3, \qquad x(3) = 4$$

Therefore,

$$\boxed{x(n) = \{1, 2, 3, 4\}}$$

The original time-domain sequence is recovered.

Important Questions:

1. Define the Discrete Fourier Series (DFS). Derive the DFS analysis and synthesis equations.
2. Explain the properties of DFS with suitable mathematical expressions.
3. Define the Discrete Fourier Transform (DFT) and derive the DFT and Inverse DFT (IDFT) equations.
4. Explain the properties of DFT: periodicity, linearity, circular time shift, frequency shift, symmetry, and circular convolution.
5. Explain the difference between linear convolution and circular convolution with a suitable example.
6. Explain how to obtain linear convolution using DFT. Describe the zero-padding procedure.
7. Explain the sampling process and state the sampling theorem. Discuss aliasing and its prevention.
8. Explain quantization and quantization error. Derive the quantization signal-to-noise ratio (SQNR), where applicable.