Department of Electrical Engineering
Indian Institute of Technology Kharagpur

Digital Signal Processing Laboratory (EE39203)

Autumn, 2022-23

Experiment 6: Discrete Fourier Transform

Slot: Date:

Student Name: Roll No.:

Grading Rubric

	Tick the best applicable per row			Points
	Below Expectation	Lacking in Some	Meets all Expectation	
Completeness of the report				
Organization of the report (5 pts) *With cover sheet, answers are in the same order as questions in the lab, copies of the questions are included in report, prepared in LaTeX*				
Quality of figures (5 pts) *Correctly labelled with title, x-axis, y-axis, and name(s)*				
Understanding the effects of truncating the signal on its DTFT (30 pts) *Magnitude and phase plots, hamming/rect windows, questions*				
Implementation of DFT and inverse DFT (30 pts) *Matlab codes, frequency and time-domain plots, analytical expressions*				
Implementation of DFT and IDFT using matrix multiplication (20 pts) *Matrices A,B,C, Matlab codes, plots, questions*				
Computation time comparison (10 pts) *Runtimes, questions*				
			TOTAL (100 pts)	

Total Points (100): TA Name: TA Initials:

Digital Signal Processing Laboratory (EE39203)
Experiment 6: Discrete Fourier Transform Algorithm

Name: Sujay Vivek
Roll Number: 22EE30029

October 16th 2024

1. Learning Objective

The discrete Fourier Transform (DFT) is a fundamental tool in digital signal processing, enabling the analysis and representation of discrete-time signals in the frequency domain. Unlike the continuous Fourier transform, which operates on signals defined over continuous time, the DFT focuses on finite-length sequences, making it more suitable for practical computational applications. In this experiment, we explore the derivation of the DFT from the Discrete-Time Fourier Transform (DTFT), highlighting the effects of windowing and truncation on signal analysis

2. Deriving DFT from DTFT

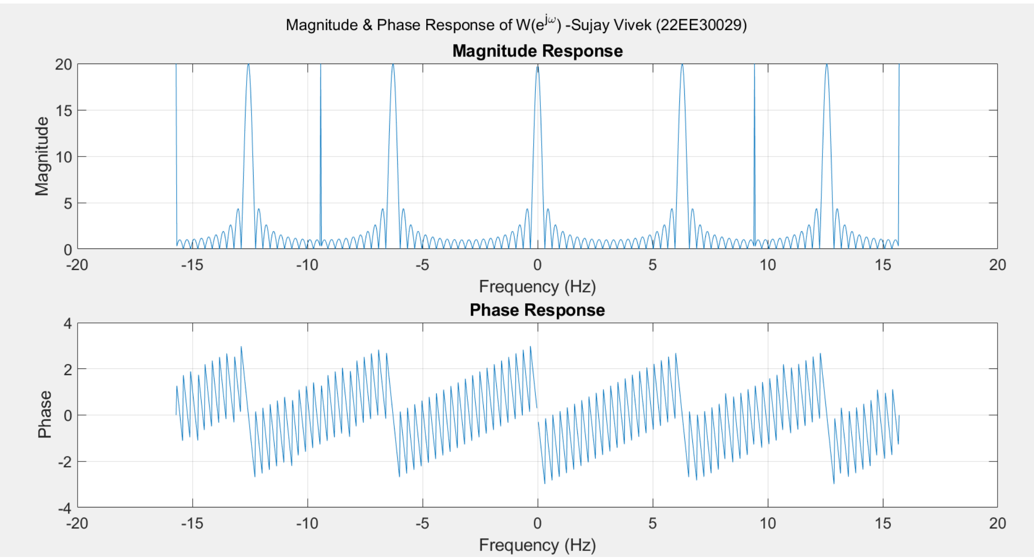


Figure 1: Magnitude Response and Phase Response of W(e^{jw})

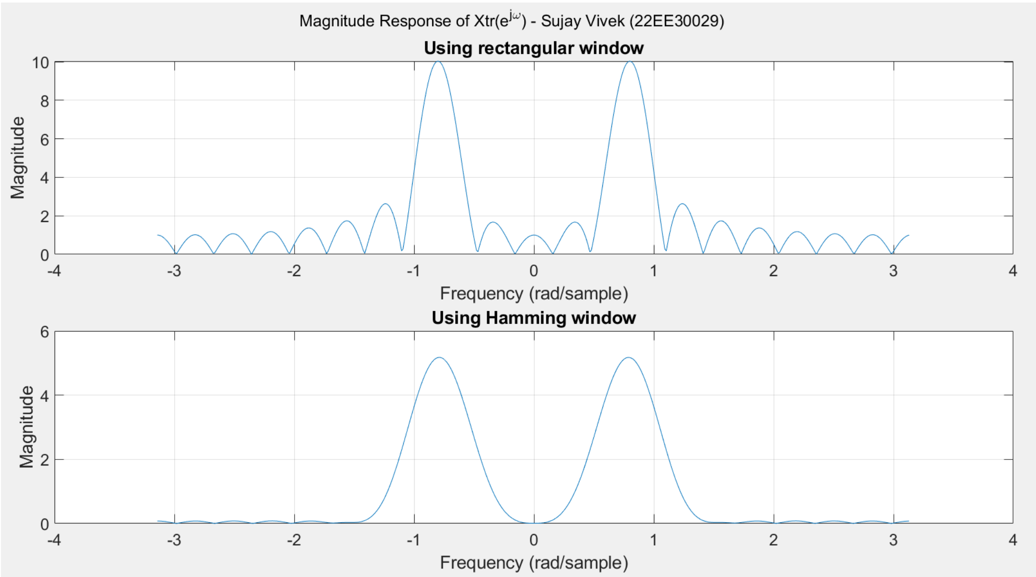


Figure 2: Magnitude Response of $X_{tr}(e^{jw})$ using Different windows

We have used the Discrete-Time Fourier Transform (DTFT) extensively for analyzing signals and linear time-invariant systems.

$$\text{DTFT: } X\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x(n) e^{-jn\omega}$$

$$\text{Inverse DTFT: } x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X\left(e^{j\omega}\right) e^{j\omega n} d\omega$$

The discrete Fourier transform (DFT) is a sampled version of the DTFT, hence it is better suited for numerical evaluation on computers.

$$DFT : X_N(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}$$

Inverse DFT: $x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X_N(k) e^{j2\pi k\pi/N}$ Here $X_{\mathrm{M}}(k)$ is an N-point DFT of $x(n)$. Note that $XN(k)$ is a function of discrete integer k, where k ranges from 0 to $N-1$

Analytical Expression for X(e^{jw})

$$W\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} w(n)e^{-j\omega n} = \sum_{n=0}^{N-1} e^{-j\omega n} = \begin{cases} \frac{1-e^{-j\omega N}}{1-e^{-j\omega}}, & \text{for } \omega \neq 0, \pm 2\pi, \ldots \\ N, \\ \text{for } \omega = 0, \pm 2\pi, \ldots \end{cases}$$

$$X_{tr}\left(e^{j\omega}\right) = \sum_{n=0}^{N-1} x(n)e^{-j\omega n}$$

$$x[n] = \cos\left(\frac{\pi n}{4}\right) = \frac{1}{2}\left(e^{j\frac{m}{4}} + e^{-j\frac{\pi n}{4}}\right)$$

$$X\left(e^{j\omega}\right) = \frac{1}{2}\sum_{n=-\infty}^{\infty}\left(e^{j} \quad + e^{-j\frac{\pi}{\tau}}\right)e^{-j\omega n}$$

$$X\left(e^{j\omega}\right) = \frac{1}{2}\left(\sum_{n=-\infty}^{\infty} e^{jn(\xi-\omega)} + \sum_{n=-\infty}^{\infty} e^{-jn(\xi+\omega)}\right)$$

where, N = 20.
The summation formula for an infinite geometric series is given by:

$$\sum_{n=-\infty}^{\infty} r^n = 2\pi \sum_{k=-\infty}^{\infty} \delta(\theta - 2\pi k), \text{ for } |r| = 1$$

$$\Rightarrow X\left(e^{j\omega}\right) = \frac{\pi}{2}\left(\delta\left(\omega - \frac{\pi}{4}\right) + \delta\left(\omega + \frac{\pi}{4}\right)\right)$$

Difference between $X(e^{jw})$ and $Xtr(e^{jw})$

$X(e^{jw})$ represents the Discrete-Time Fourier Transform (DTFT) of the original, untruncated infinite-length signal, whereas, $Xtr(e^{jw})$ represents the DTFT of the truncated signal, i.e., the signal after it has been multiplied by a finite-length window (often rectangular).

The difference lies in the distortion caused by the truncation.

The difference between the truncated signal's DTFT, $Xtr(e^{jw})$, and the original signal's DTFT, $X(e^{jw})$, arises because truncation distorts the frequency content of the signal. Truncating the time-domain signal is equivalent to convolving its frequency spectrum with the DTFT of the truncation window, which smooths out the original spectrum. This results in a spreading of frequency components, especially if the window is not smooth (like a rectangular window). Consequently, $Xtr(e^{jw})$ is often a smoothed or blurred version of $X(e^{jw})$.

Difference on using different windows for $w(n)$

Using a Hamming window instead of a rectangular window $w(n)$ would significantly reduce the spectral leakage and side lobes in the frequency domain. A rectangular window (which is used in simple truncation) produces a sinc-like frequency response with strong side lobes, which cause energy from a single frequency component to spread across adjacent frequencies. This results in broader peaks and more distortion in the frequency spectrum. In contrast, the Hamming window is designed to minimize side lobes while sacrificing a little resolution in the main lobe width. The main lobe in the frequency spectrum for a Hamming window is slightly wider than that of the rectangular window, but the side lobes are much smaller and decrease faster. This means that while the frequency resolution may be slightly reduced (wider peaks), the overall accuracy in distinguishing frequency components would improve due to the suppression of side lobes.

3. The Discrete Fourier Transform

3.1 Computing the DFT and IDFT

1. $x[n] = \delta[n]$ (Unit Impulse for $N = 10$)

$$X[k] = \sum_{n=0}^{g} \delta[n] e^{-j\frac{2\pi k}{\omega t}} = 1$$
$$X[k] = 1 \quad \text{for all } k$$

2. $x[n] = 1$ for $N = 10$ (Constant Signal)

$$X[k] = \sum_{n=0}^{9} 1 \cdot e^{-j\frac{\text{and}}{10}} = \sum_{n=0}^{9} e^{-j\frac{\text{and}}{10}}$$
$$X[k] = \begin{cases} 10, & k = 0 \\ 0, & k \neq 0 \end{cases}$$

4 3. $x[n] = e^{j\frac{2m}{10}}$ for $N = 10$

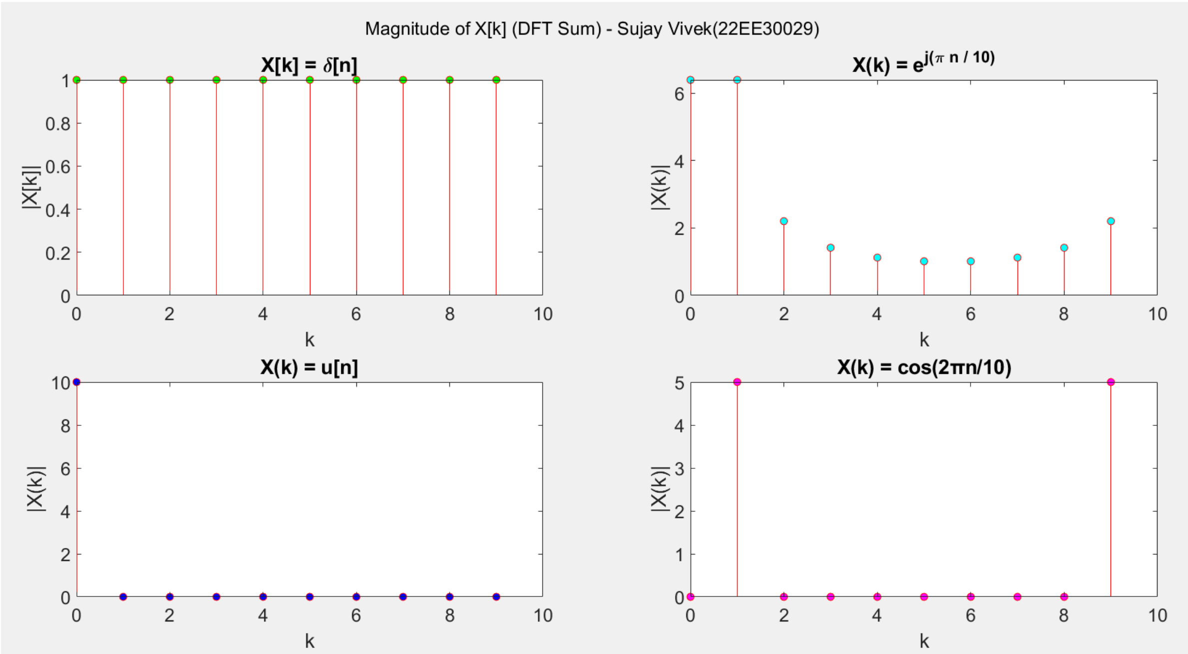
$$X[k] = \begin{cases} 10, & k = 1 \\ 0, & k \neq 1 \end{cases}$$

4. $x[n] = \cos\left(\frac{2\pi n}{10}\right)$ for $N = 10$

Using Euler's formula, $\cos\left(\frac{2\pi \mathrm{rn}}{10}\right) = \frac{1}{2}\left(e^{j\frac{\pi}{16}} + e^{-j\frac{2}{2}}\right)$, we compute the DFT:

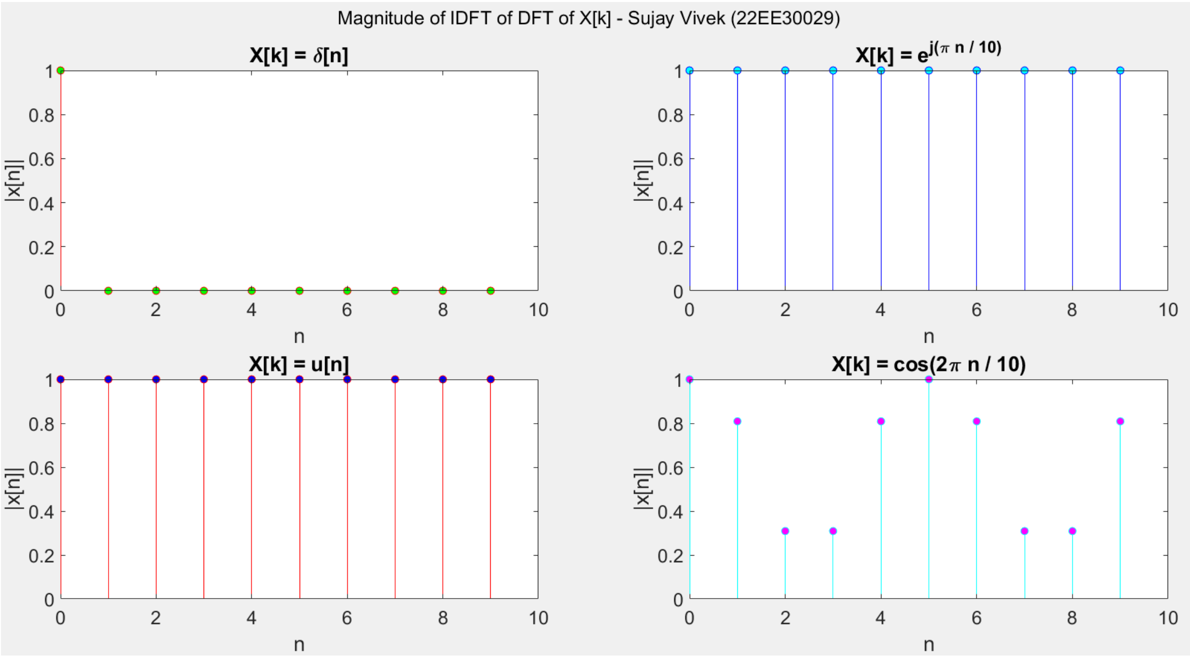
From sequence 3, we know that the DFT of $e^{j\frac{3\pi}{10}}$ is 10 at $k = 1$ and zero elsewhere. Similarly, the DFT of $e^{-j\frac{3m}{3}}$ is 10 at $k = 9$. Thus, the DFT becomes

$$X[k] = \begin{cases} 5, & k = 1 \text{ or } k = 9 \\ 0, & \text{otherwise} \end{cases}$$



(a) Magnitude of DFT of different functions

Now, plotting the magnitude response of x[n] that is the IDFT of X[k] with the help of MATLAB,



(a) Magnitude of IDFT of DFT of different functions

3.2 Matrix Representation of DFT and IDFT

Matrix representation of DFT

The DFT can be implemented as a matrix-vector product. To see this, consider the equation: $\mathrm{X} = \boldsymbol{Ax}$ where $\boldsymbol{A}$ is an $N \times N$ matrix, and both $\boldsymbol{X}$ and $\boldsymbol{x}$ are $N \times 1$ column vectors. This matrix product is equivalent to the summation

$$X_k = \sum_{n=1}^{N} \boldsymbol{A}_{kn} \boldsymbol{x}_n$$

where $\boldsymbol{A}_{\mathrm{kn}}$ is the matrix element in the k^{th} row and n^{th} column of $\boldsymbol{A}$.

$$\boldsymbol{A}_{kn} = e^{-j2\pi(k-1)(n-1)/N}$$

Matrix **A** for N = 5 :

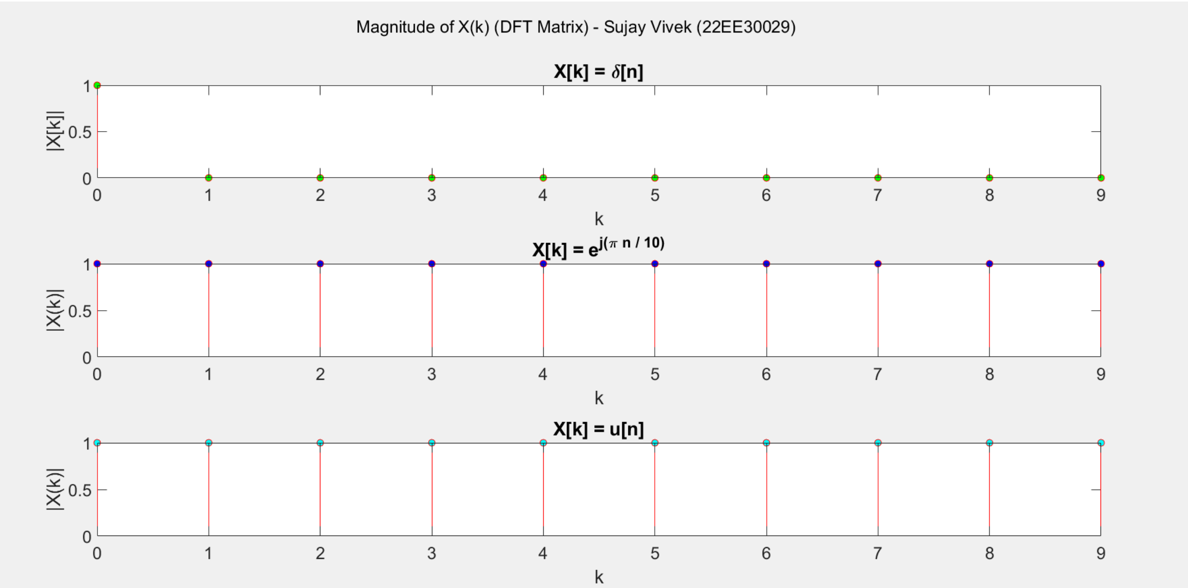
$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & e^{-j\frac{2\pi}{5}} & e^{-j\frac{4\pi}{5}} & e^{-j\frac{6\pi}{5}} & e^{-j\frac{8\pi}{5}} \\ 1 & e^{-j\frac{4\pi}{5}} & e^{-j\frac{8\pi}{5}} & e^{-j\frac{12\pi}{5}} & e^{-j\frac{16\pi}{5}} \\ 1 & e^{-j\frac{6\pi}{5}} & e^{-j\frac{12\pi}{5}} & e^{-j\frac{18\pi}{5}} & e^{-j\frac{24\pi}{5}} \\ 1 & e^{-j\frac{8\pi}{5}} & e^{-j\frac{16\pi}{5}} & e^{-j\frac{24\pi}{5}} & e^{-j\frac{32\pi}{5}} \end{pmatrix}$$

Number of Multiplications :- For each element $X[k]$ in the output vector X, we need to compute:

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

This means that for each k (from O to $\mathrm{N}-1$), we need to compute N complex multiplications (one for each n). Thus, the total number of multiplications required for the matrix method is:

$$\text{Total Multiplications } =\mathrm{N}\times\mathrm{N}=\mathrm{N}^{2}$$



(a) DFT matrix Magnitude of X[k]

Matrix representation of IDFT

As $X=Ax$ and $x=BX$ Hence, $\mathbf{B}=\boldsymbol{A}^{-1}$

$$\mathrm{B}[\mathrm{k},n]=\frac{1}{N}\mathrm{e}^{\mathrm{j}2\pi(\mathrm{k}-1)(n-1)/\mathrm{N}}$$

Matrix B for $\mathrm{N}=5$:

$$B=\begin{pmatrix}\frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5}\\ \frac{1}{5} & e^{j\frac{2\pi}{5}}\cdot\frac{1}{5} & e^{j\frac{4\pi}{5}}\cdot\frac{1}{5} & e^{j\frac{6\pi}{5}}\cdot\frac{1}{5} & e^{j\frac{8\pi}{5}}\cdot\frac{1}{5}\\ \frac{1}{5} & e^{j\frac{4\pi}{5}}\cdot\frac{1}{5} & e^{j\frac{8\pi}{5}}\cdot\frac{1}{5} & e^{j\frac{12\pi}{5}}\cdot\frac{1}{5} & e^{j\frac{16\pi}{5}}\cdot\frac{1}{5}\\ \frac{1}{5} & e^{j\frac{6\pi}{5}}\cdot\frac{1}{5} & e^{j\frac{12\pi}{5}}\cdot\frac{1}{5} & e^{j\frac{18\pi}{5}}\cdot\frac{1}{5} & e^{j\frac{24\pi}{5}}\cdot\frac{1}{5}\\ \frac{1}{5} & e^{j\frac{8\pi}{5}}\cdot\frac{1}{5} & e^{j\frac{16\pi}{5}}\cdot\frac{1}{5} & e^{j\frac{24\pi}{5}}\cdot\frac{1}{5} & e^{j\frac{32\pi}{5}}\cdot\frac{1}{5}\end{pmatrix}$$

Matrix representation of C

Matrix C : Here, $\quad C = BA$

$$\Rightarrow C = A^{-1}\boldsymbol{A}$$

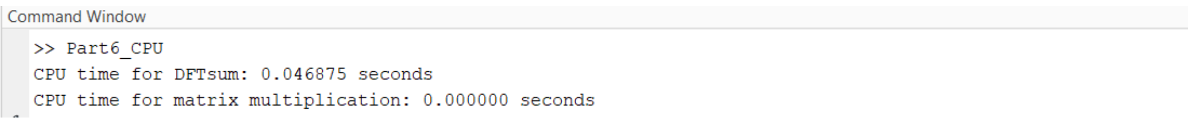
$$\Rightarrow C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

For N = 5

The Matrix C is an identity Matrix of $N \times N$.

The identity matrix indicates that the DFT followed by the IDFT recovers the original signal perfectly. In other words, applying the DFT matrix A to a signal and then multiplying by the IDFT matrix B restores the signal to its original form without any loss of information, validating the mathematical relationship between the DFT and IDFT. Thus, C=B_A=I_N, where I_N is the identity matrix.

Computation Time Analysis



(a) Computation Time

Which Method is Faster :

Typically, the matrix multiplication method (using A) is faster for larger values of N due to optimized linear algebra operations in MATLAB. The complexity of the direct nested loop implementation is O(N2) whereas the matrix method, while also O(N2), benefits from more efficient handling of matrix operations in the backend.

Which Method Requires Less Storage :

The direct computation method (DFTsum) does not require storage for a large matrix and instead only needs to store the input signal and a few temporary variables. Thus, it is more memory-efficient for small to moderate N. The matrix multiplication method requires storing the entire DFT matrix A, which is N×N. For large N, this can consume significant memory.