EEE3030 Digital Signal Processing Notes

Introduction

This set of notes will cover 1. The prerequisites for learning about digital signal processing methods, 2. The aforementioned methods and how we use them and where they are used.

Vectors

2.1 What is a Vector?

A **vector** is a mathematical object that has both **magnitude** (length) and **direction**. Vectors are often written in boldface, such as $\mathbf{v}$, or with an arrow, $\vec{v}$. In coordinate form, a vector in $\mathbb{R}^n$ is written as

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}.$$

2.2 Vector Equality

Two vectors $\mathbf{u}$ and $\mathbf{v}$ are equal if and only if all their components are equal:

$$\mathbf{u} = \mathbf{v} \quad \Leftrightarrow \quad u_i = v_i \ \ \forall i.$$

2.3 Vector Addition

The sum of two vectors is obtained by adding their corresponding components:

$$\mathbf{u} + \mathbf{v} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} u_1 + v_1 \\ u_2 + v_2 \\ \vdots \\ u_n + v_n \end{bmatrix}.$$

2.4 Scalar Multiplication

Multiplying a vector by a scalar $c \in \mathbb{R}$ stretches or shrinks it:

$$c\mathbf{v} = c \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} cv_1 \\ cv_2 \\ \vdots \\ cv_n \end{bmatrix}.$$

2.5 Vector Magnitude (Norm)

The length (or magnitude) of a vector $\mathbf{v}$ is

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}.$$

2.6 Unit Vectors

A **unit vector** has magnitude 1. To normalize a vector $\mathbf{v}$:

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}.$$

2.7 Dot Product (Scalar Product)

For vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$:

$$\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{n} u_i v_i = \|\mathbf{u}\| \|\mathbf{v}\| \cos\theta,$$

where θ is the angle between $\mathbf{u}$ and $\mathbf{v}$.

- If $\mathbf{u} \cdot \mathbf{v} = 0$, the vectors are orthogonal.

2.8 Cross Product (in $\mathbb{R}^3$)

For vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$:

$$\mathbf{u} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix} = (u_2 v_3 - u_3 v_2)\mathbf{i} - (u_1 v_3 - u_3 v_1)\mathbf{j} + (u_1 v_2 - u_2 v_1)\mathbf{k}.$$

The cross product is a vector perpendicular to both $\mathbf{u}$ and $\mathbf{v}$.

2.9 Projection of One Vector onto Another

The projection of $\mathbf{u}$ onto $\mathbf{v}$ is

$$\text{proj}_{\mathbf{v}} \mathbf{u} = \left(\frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{v}\|^2} \right) \mathbf{v}.$$

2.10 Orthogonality

Vectors are **orthogonal** (perpendicular) if

$$\mathbf{u} \cdot \mathbf{v} = 0.$$

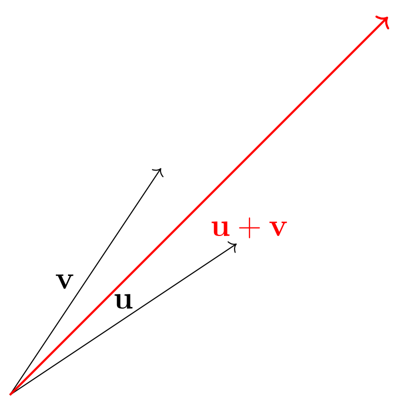
2.11 Linear Combination

A vector $\mathbf{w}$ is a linear combination of $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ if

$$\mathbf{w} = c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \cdots + c_k \mathbf{v}_k$$

for some scalars $c_1, c_2, \ldots, c_k$.

2.12 Geometric Visualization



Matrices

A **matrix** is a rectangular array of numbers arranged in rows and columns. An $m \times n$ matrix has m rows and n columns:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

3.1 Matrix Equality

Two matrices A and B are equal if they have the same size and

$$a_{ij} = b_{ij} \quad \forall i, j.$$

3.2 Matrix Addition

If $A, B \in \mathbb{R}^{m \times n}$, then

$$A + B = \begin{bmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{bmatrix}.$$

3.3 Scalar Multiplication

For $c \in \mathbb{R}$,

$$cA = \begin{bmatrix} ca_{11} & ca_{12} & \cdots & ca_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ ca_{m1} & ca_{m2} & \cdots & ca_{mn} \end{bmatrix}.$$

3.4 Matrix Multiplication

If $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$, then

$$C = AB \in \mathbb{R}^{m \times p},$$

where each entry is computed as

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}.$$

3.4.1 Example

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 \cdot 5 + 2 \cdot 7 & 1 \cdot 6 + 2 \cdot 8 \\ 3 \cdot 5 + 4 \cdot 7 & 3 \cdot 6 + 4 \cdot 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}.$$

3.5 Transpose

The **transpose** of $A \in \mathbb{R}^{m \times n}$ is $A^T \in \mathbb{R}^{n \times m}$, defined by

$$(A^T)_{ij} = a_{ji}.$$

3.6 Identity Matrix

The identity matrix I_n is an $n \times n$ matrix with ones on the diagonal and zeros elsewhere:

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

It satisfies $AI_n = I_m A = A$ for compatible A.

3.7 Determinant

For a square matrix $A \in \mathbb{R}^{n \times n}$, the **determinant** $\det(A)$ is a scalar with geometric meaning (volume scaling).

For 2×2 matrices:

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc.$$

For 3×3 matrices:

$$\det \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = a(ei - fh) - b(di - fg) + c(dh - eg).$$

3.8 Inverse of a Matrix

A square matrix A is invertible if there exists A^{-1} such that

$$AA^{-1} = A^{-1}A = I.$$

For 2×2 matrices:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}, \quad \det(A) \neq 0.$$

3.9 Rank of a Matrix

The **rank** of a matrix A is the dimension of its column space (or row space). It equals the maximum number of linearly independent rows or columns.

3.10 Eigenvalues and Eigenvectors

For a square matrix A, a nonzero vector $\mathbf{v}$ is an eigenvector if

$$A\mathbf{v} = \lambda\mathbf{v},$$

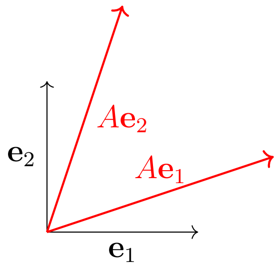
where λ is the corresponding eigenvalue.

To find λ, solve the characteristic equation:

$$\det(A - \lambda I) = 0.$$

3.11 Geometric Interpretation

- Matrix multiplication can be seen as a linear transformation of space.
- Determinant measures area/volume scaling and orientation.
- Eigenvectors are directions that remain unchanged under the transformation.



Complex Numbers

A complex number is defined as

$$z = x + iy, \quad x, y \in \mathbb{R}, \quad i^2 = -1.$$

$$\Re(z) = x, \quad \Im(z) = y, \quad \overline{z} = x - iy.$$

$$|z| = \sqrt{x^2 + y^2}, \quad \arg(z) = \theta.$$

4.1 Forms

$$z = x + iy \quad \text{(rectangular)}, \quad z = r(\cos\theta + i\sin\theta) \quad \text{(polar)}, \quad z = re^{i\theta} \quad \text{(exponential)}.$$

4.2 Euler's Formula

$$e^{i\theta} = \cos\theta + i\sin\theta.$$

Special cases:

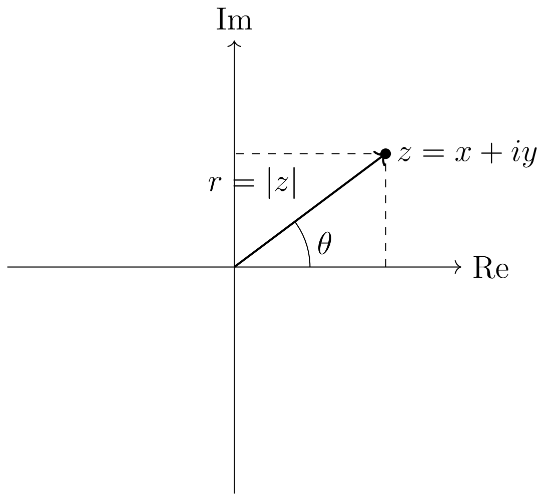
$$e^{i\pi} + 1 = 0, \quad e^{i\pi/2} = i.$$

4.3 De Moivre's Theorem

$$(re^{i\theta})^n = r^n e^{in\theta} = r^n(\cos n\theta + i\sin n\theta).$$

4.4 Argand Diagram

A geometric representation of $z = x + iy$ as a point (x, y).

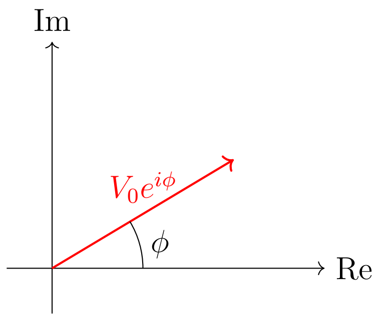


4.5 Phasor Diagram

A sinusoid

$$v(t) = V_0 \cos(\omega t + \phi) = \Re\{V_0 e^{i(\omega t + \phi)}\}$$

is represented by a rotating phasor vector.



4.6 ExponentialTrigonometric Relations

$$\cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2}, \quad \sin\theta = \frac{e^{i\theta} - e^{-i\theta}}{2i}.$$

4.7 Hyperbolic Functions

$$\cosh x = \frac{e^{x} + e^{-x}}{2}, \quad \sinh x = \frac{e^{x} - e^{-x}}{2}.$$

Relations:

$$\cos(ix) = \cosh x, \quad \sin(ix) = i \sinh x.$$

4.8 Trigonometric Identities

4.8.1 Pythagorean

$$\sin^2\theta + \cos^2\theta = 1, \quad 1 + \tan^2\theta = \sec^2\theta, \quad 1 + \cot^2\theta = \csc^2\theta.$$

4.8.2 Sum and Difference

$$\sin(a \pm b) = \sin a \cos b \pm \cos a \sin b,$$
$$\cos(a \pm b) = \cos a \cos b \mp \sin a \sin b,$$
$$\tan(a \pm b) = \frac{\tan a \pm \tan b}{1 \mp \tan a \tan b}.$$

4.8.3 Double Angle

$$\sin(2\theta) = 2\sin\theta\cos\theta, \quad \cos(2\theta) = \cos^2\theta - \sin^2\theta,$$
$$\tan(2\theta) = \frac{2\tan\theta}{1-\tan^2\theta}.$$

4.8.4 Half Angle

$$\sin^2\frac{\theta}{2} = \frac{1-\cos\theta}{2}, \quad \cos^2\frac{\theta}{2} = \frac{1+\cos\theta}{2}.$$

4.8.5 Product-to-Sum

$$\begin{aligned}
\sin a \sin b &= \tfrac{1}{2}[\cos(a-b) - \cos(a+b)],\\
\cos a \cos b &= \tfrac{1}{2}[\cos(a-b) + \cos(a+b)],\\
\sin a \cos b &= \tfrac{1}{2}[\sin(a+b) + \sin(a-b)].
\end{aligned}$$

4.8.6 Sum-to-Product

$$\begin{aligned}
\sin a \pm \sin b &= 2\sin\left(\tfrac{a\pm b}{2}\right)\cos\left(\tfrac{a\mp b}{2}\right),\\
\cos a + \cos b &= 2\cos\left(\tfrac{a+b}{2}\right)\cos\left(\tfrac{a-b}{2}\right),\\
\cos a - \cos b &= -2\sin\left(\tfrac{a+b}{2}\right)\sin\left(\tfrac{a-b}{2}\right).
\end{aligned}$$

4.9 Hyperbolic Identities

4.9.1 Fundamental

$$\cosh^2 x - \sinh^2 x = 1.$$

4.9.2 Sum and Difference

$$\begin{aligned}
\sinh(a\pm b) &= \sinh a\cosh b \pm \cosh a \sinh b,\\
\cosh(a\pm b) &= \cosh a\cosh b \pm \sinh a \sinh b.
\end{aligned}$$

4.9.3 Double Angle

$$\sinh(2x) = 2\sinh x\cosh x, \quad \cosh(2x) = \cosh^2 x + \sinh^2 x.$$

4.9.4 Half Angle

$$\cosh^2\frac{x}{2} = \frac{\cosh x + 1}{2}, \quad \sinh^2\frac{x}{2} = \frac{\cosh x - 1}{2}.$$

4.10 Roots of Unity

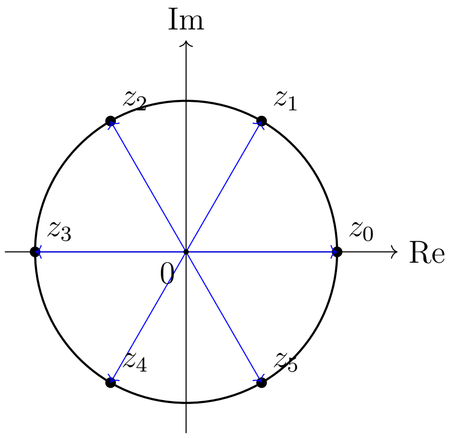
The n-th roots of unity are the solutions of

$$z^n = 1.$$

They are given by

$$z_k = e^{i\frac{2\pi k}{n}} = \cos\left(\frac{2\pi k}{n}\right) + i\sin\left(\frac{2\pi k}{n}\right), \quad k = 0, 1, 2, \ldots, n-1.$$

These correspond to n equally spaced points on the unit circle in the Argand plane.



For example, for $n = 6$ the roots are

$$z_k = e^{i\frac{2\pi k}{6}}, \quad k = 0, 1, 2, 3, 4, 5,$$

which are vertices of a regular hexagon inscribed in the unit circle.

4.11 Poles and Zeroes

In complex analysis and systems theory, the behaviour of a function

$$F(s) = \frac{N(s)}{D(s)}$$

is characterized by its **zeroes** and **poles**.

- A **zero** is a value s_0 such that $F(s_0) = 0$ (i.e. $N(s_0) = 0$).
- A **pole** is a value s_p where $F(s)$ tends to infinity (i.e. $D(s_p) = 0$).

4.11.1 Example:

Consider

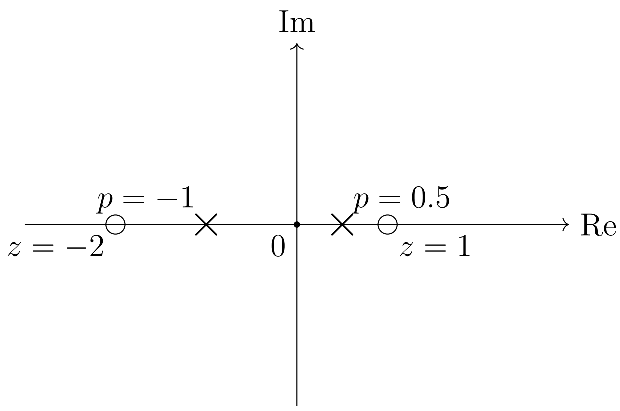
$$F(s) = \frac{(s-1)(s+2)}{(s-0.5)(s+1)}.$$

Zeroes: $s = 1$, $s = -2$ Poles: $s = 0.5$, $s = -1$

4.11.2 Pole–Zero Diagram

Poles and zeroes are plotted in the complex plane (often the s-plane or z-plane):

- Zeroes are marked with a $\circ$ (circle).
- Poles are marked with a $\times$ (cross).



4.11.3 Interpretation

- Zeroes indicate where the system response is **cancelled**.
- Poles indicate where the system response is **resonant** or tends to infinity.
- The relative locations of poles and zeroes determine the stability and frequency response of a system.

Fourier Series

Any periodic function $f(t)$ with period T can be expanded into a Fourier series:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{2\pi n}{T} t\right) + b_n \sin\left(\frac{2\pi n}{T} t\right)\right).$$

5.1 Fourier Coefficients

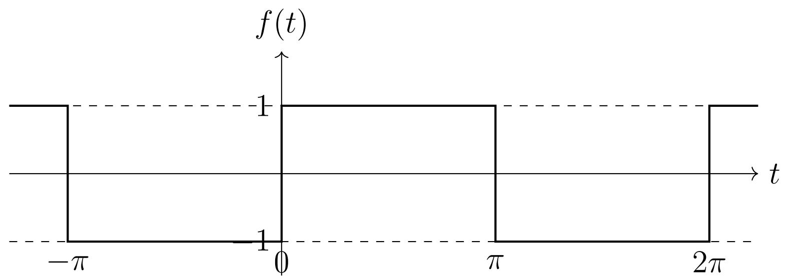
$$a_0 = \frac{2}{T}\int_0^T f(t)\,dt, \quad a_n = \frac{2}{T}\int_0^T f(t)\cos\left(\tfrac{2\pi n}{T}t\right)\,dt,$$

$$b_n = \frac{2}{T}\int_0^T f(t)\sin\left(\tfrac{2\pi n}{T}t\right)\,dt.$$

5.2 Example: Square Wave

Consider a square wave $f(t)$ with period $T = 2\pi$ defined as

$$f(t) = \begin{cases} 1, & 0 < t < \pi, \\ -1, & -\pi < t < 0, \end{cases} \quad \text{and extended periodically.}$$



5.3 Step 1: Compute a_0

$$a_0 = \frac{2}{2\pi}\int_{-\pi}^{\pi} f(t)\,dt = \frac{1}{\pi}\left(\int_{-\pi}^{0} -1\,dt + \int_0^{\pi} 1\,dt\right) = 0.$$

5.4 Step 2: Compute a_n

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\cos(nt)\,dt.$$

Since $f(t)$ is odd and $\cos(nt)$ is even, their product is odd. Thus

$$a_n = 0.$$

5.5 Step 3: Compute b_n

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\sin(nt)\,dt.$$

Now $f(t)$ is odd and $\sin(nt)$ is odd $\implies$ product is even:

$$b_n = \frac{2}{\pi}\int_{0}^{\pi} (1)\sin(nt)\,dt.$$

$$b_n = \frac{2}{\pi}\left[\frac{1-\cos(n\pi)}{n}\right] = \frac{2}{n\pi}(1-(-1)^n).$$

So:

$$b_n = \begin{cases} \frac{4}{n\pi}, & n \text{ odd},\\ 0, & n \text{ even}.\end{cases}$$

5.6 Final Fourier Series (Sine Form)

$$f(t) = \frac{4}{\pi}\left(\sin t + \frac{1}{3}\sin 3t + \frac{1}{5}\sin 5t + \cdots\right).$$

5.7 Complex Form of Fourier Series

We can also write the Fourier series using complex exponentials:

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{int}, \quad c_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(t)e^{-int}\,dt.$$

5.8 Step 1: Compute c_n

For the square wave:

$$c_n = \frac{1}{2\pi}\left(\int_0^{\pi} 1\cdot e^{-int}\,dt + \int_{-\pi}^{0} -1\cdot e^{-int}\,dt\right).$$

$$c_n = \frac{1}{2\pi}\left(\frac{1-e^{-in\pi}}{-in} - \frac{e^{in\pi}-1}{-in}\right).$$

$$c_n = \frac{1}{\pi i n}(1-(-1)^n).$$

Thus:

$$c_n = \begin{cases} \frac{2}{i\pi n}, & n \text{ odd},\\ 0, & n \text{ even}.\end{cases}$$

5.9 Step 2: Final Complex Series

$$f(t) = \sum_{\substack{n=-\infty\\ n \text{ odd}}}^{\infty} \frac{2}{i\pi n} e^{int}.$$

5.10 Interpretation

- The sine form shows explicitly the harmonic content (only odd harmonics).
- The complex form is more compact and symmetric, useful in analysis and physics.
- Both represent the same square wave in the limit of infinite harmonics.

Fourier Transforms and the DFT

The Fourier Transform (FT) converts a time-domain signal into its frequency-domain representation.

6.1 From Fourier Series to Fourier Transform (Step by Step)

A periodic signal $x_T(t)$ of period T can be expressed as a Fourier Series:

$$x_T(t) = \sum_{n=-\infty}^{\infty} c_n e^{in\omega_0 t}, \qquad \omega_0 = \frac{2\pi}{T}.$$

The coefficients are

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} x_T(t)\, e^{-in\omega_0 t}\, dt.$$

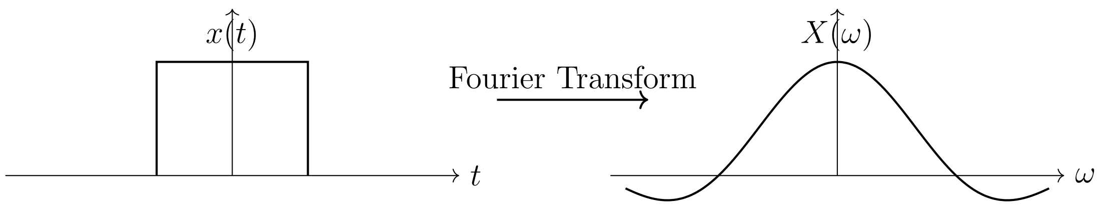
1. As $T \to \infty$, $x_T(t) \to x(t)$ (aperiodic signal).
2. The spacing between harmonics $\omega_0 = 2\pi/T \to 0$, giving a continuous frequency axis ω.
3. The discrete coefficients c_n become continuous spectrum values $X(\omega)$.
4. The summation turns into an integral.

Thus, the Fourier Transform pair is obtained:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)\, e^{-i\omega t}\, dt, \qquad x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)\, e^{i\omega t}\, d\omega.$$

6.2 Time and Frequency Domains

- **Time domain** $x(t)$**:** signal evolution with respect to time.
- **Frequency domain** $X(\omega)$**:** how much of each sinusoidal frequency is present.



A localised rectangular pulse in time spreads into a sinc in frequency. Conversely, a single sinusoid in time becomes a delta spike in frequency.

6.3 Fourier Transform Properties: Shifts

- **Time Shift Theorem:** If $x(t) \leftrightarrow X(\omega)$, then

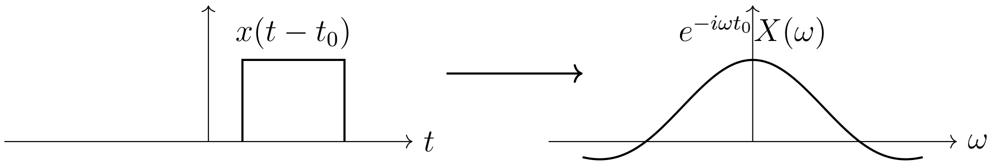
  $$x(t - t_0) \quad \Longleftrightarrow \quad e^{-i\omega t_0} X(\omega).$$

  (Time delay $\to$ frequency phase shift)

- **Frequency Shift Theorem:** If $x(t) \leftrightarrow X(\omega)$, then

$$e^{i\omega_0 t}x(t) \quad \Longleftrightarrow \quad X(\omega - \omega_0).$$

(Modulation in time $\to$ spectrum shift)



6.4 Worked Example: Rectangular Pulse

Let

$$x(t) = \begin{cases} 1, & |t| \le \frac{T}{2}, \\ 0, & |t| > \frac{T}{2}. \end{cases}$$

Fourier Transform:

$$X(\omega) = \int_{-T/2}^{T/2} e^{-i\omega t}\,dt = \frac{2\sin(\omega T/2)}{\omega}.$$

Thus

$$X(\omega) = T \cdot \operatorname{sinc}\left(\frac{\omega T}{2\pi}\right).$$

Derivation of the Discrete Fourier Transform (DFT)

7.1 Step 1: Continuous-Time Fourier Transform (CTFT)

For a continuous-time signal $x(t)$, the Fourier Transform is

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-i\omega t}\,dt.$$

7.2 Step 2: Sampling the Signal

We sample $x(t)$ every T_s seconds:

$$x[n] = x(nT_s), \quad n \in \mathbb{Z}.$$

This produces a discrete-time sequence.

7.3 Step 3: Discrete-Time Fourier Transform (DTFT)

The DTFT is

$$X(e^{i\omega}) = \sum_{n=-\infty}^{\infty} x[n]\,e^{-i\omega n}.$$

Properties:

- $X(e^{i\omega})$ is continuous in ω.
- $X(e^{i\omega})$ is 2π-periodic.

7.4 Step 4: Finite-Length Signals

In practice, we only keep N samples:

$$x[n] = 0 \quad \text{for } n < 0 \text{ or } n \geq N.$$

Equivalently, we treat $x[n]$ as N-periodic:

$$x[n+N] = x[n].$$

7.5 Step 5: Fourier Series Representation

Since $x[n]$ is N-periodic, it has a discrete Fourier series:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{i2\pi kn/N}.$$

7.6 Step 6: Coefficients (DFT)

The coefficients are

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-i2\pi kn/N}, \quad k = 0, 1, \ldots, N-1.$$

This is the **Discrete Fourier Transform (DFT)**.

7.7 Step 7: Inverse DFT

We recover $x[n]$ from $X[k]$:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{i2\pi kn/N}.$$

7.8 Step 8: Summary of the DFT Pair

$$\boxed{X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-i2\pi kn/N}, \qquad x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{i2\pi kn/N}}$$

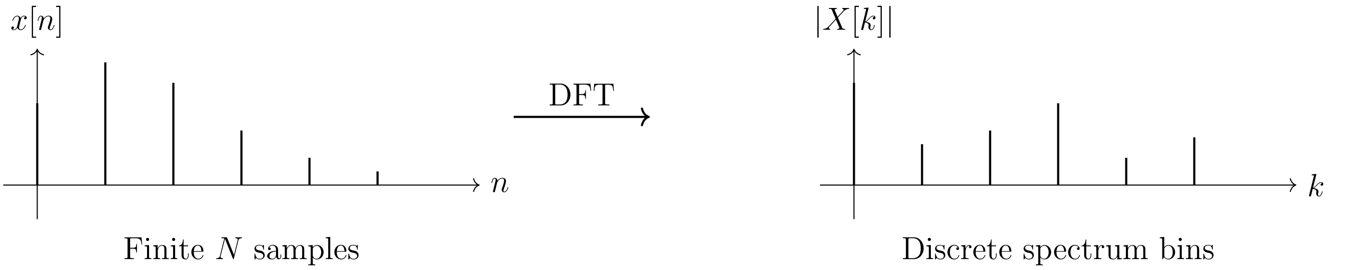
7.9 Step 9: Conditions for Validity

The DFT applies under the following assumptions:

1. The signal $x[n]$ is of finite length (N samples).
2. Outside $0 \leq n < N$, we assume $x[n] = 0$ or extend it periodically with period N.
3. The basis functions $e^{-i2\pi kn/N}$ are orthogonal:

$$\sum_{n=0}^{N-1} e^{-i2\pi(k-m)n/N} = \begin{cases} N, & k = m, \\ 0, & k \neq m. \end{cases}$$

7.10 Step 10: Diagram — Time vs Frequency Domain



This shows that:

- A finite-length time signal $x[n] \to$ a set of N discrete frequency bins $X[k]$.
- Increasing N improves frequency resolution.
- The spectrum is periodic with period N in the frequency index k.

7.11 Multiplication and Convolution in DSP

7.11.1 Convolution in DSP

In Digital Signal Processing (DSP), convolution describes how an input signal passes through a Linear Time-Invariant (LTI) system.

$$y[n] = (x * h)[n] = \sum_{m=-\infty}^{\infty} x[m]\, h[n-m],$$

where

- $x[n]$ = input signal,
- $h[n]$ = impulse response of the system,
- $y[n]$ = output signal.

Meaning: The output is obtained by "smearing" the input $x[n]$ according to how the system responds to an impulse.

Examples:

- Convolution with a moving-average filter smooths a signal.
- Convolution with $\delta[n]$ (the unit impulse) leaves the signal unchanged.

7.11.2 Multiplication in DSP

Multiplication refers to pointwise product in time:

$$y[n] = x[n] \cdot w[n],$$

where

- $x[n]$ = signal,
- $w[n]$ = window or modulating sequence.

Examples:

- Applying a window $w[n]$ before computing the DFT.
- Multiplying by a sinusoid shifts the spectrum in frequency (modulation).

7.11.3 Duality: Multiplication vs Convolution

Fourier analysis reveals a fundamental duality:

$$x[n] \cdot w[n] \iff \frac{1}{2\pi}\left(X(e^{i\omega}) * W(e^{i\omega})\right)$$

$$x[n] * h[n] \iff X(e^{i\omega}) \cdot H(e^{i\omega})$$

7.11.4 Interpretation

- **Multiplication in time → Convolution in frequency:** Truncating or windowing a signal in time spreads its spectrum (spectral leakage).
- **Convolution in time → Multiplication in frequency:** Passing a signal through a filter shapes its spectrum according to the frequency response $H(e^{i\omega})$.

7.12 DFT in Matrix Form

Let $x[n]$ be a discrete signal of length N:

$$\mathbf{x} = \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}.$$

The DFT can be written in matrix form as:

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

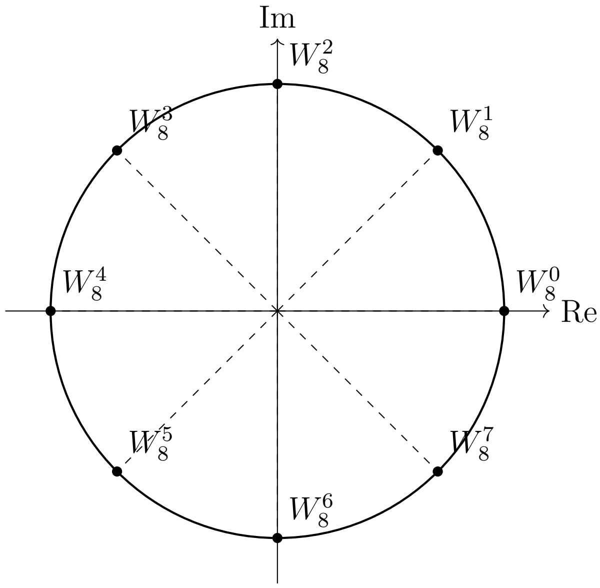
where $\mathbf{X} = [X[0], X[1], \ldots, X[N-1]]^T$ and $\mathbf{W}_N$ is the $N \times N$ DFT matrix:

$$\mathbf{W}_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix},$$

with

$$W_N = e^{-i\frac{2\pi}{N}} \quad \text{(primitive } N\text{th root of unity).}$$

7.13 Roots of Unity Diagram



This shows how the N roots of unity lie evenly spaced on the unit circle.

7.14 Worked Example: $N = 4$

Let $x[n] = [1, 2, 3, 4]^T$ with $N = 4$. Then

$$W_4 = e^{-i2\pi/4} = e^{-i\pi/2} = -i.$$

The DFT matrix is:

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

Compute

$$\mathbf{X} = \mathbf{W}_4 \, \mathbf{x}.$$

Step-by-step:

$$\begin{aligned} X[0] &= 1 + 2 + 3 + 4 = 10 \\ X[1] &= 1 + 2(-i) + 3(-1) + 4(i) = 1 - 2i - 3 + 4i = -2 + 2i \\ X[2] &= 1 + 2(-1) + 3(1) + 4(-1) = 1 - 2 + 3 - 4 = -2 \\ X[3] &= 1 + 2(i) + 3(-1) + 4(-i) = 1 + 2i - 3 - 4i = -2 - 2i \end{aligned}$$

So the DFT is:

$$\mathbf{X} = [10, -2 + 2i, -2, -2 - 2i]^T.$$

7.15 Definitions and DSP Context

- f_0: fundamental frequency of a periodic signal. For discrete signals of length N sampled at T_s, $f_0 = 1/(NT_s)$.
- Frequency (f): number of oscillations per second (Hz).
- Sampling: measuring the signal at discrete intervals T_s.
- Sampling frequency $f_s = 1/T_s$.

- Nyquist Criterion: $f_s \geq 2f_{\max}$ to avoid aliasing.
- Discrete: signal defined only at integer steps n.
- Periodic: signal repeats every N samples, $x[n+N] = x[n]$.
- N: number of samples (length of signal / DFT size).
- T_s: sampling interval in seconds.
- n: time index (integer, $0 \leq n < N$).
- k: frequency index (integer, $0 \leq k < N$), represents the kth DFT coefficient / bin.

Fourier Transforms and the DFT

Fast Fourier Transform (FFT)

9.1 Derivation of FFT from DFT

The DFT of length N is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-i2\pi kn/N}, \quad k = 0, 1, \ldots, N-1$$

Step 1: Split sum into even and odd indices (for Radix-2, N even):

$$X[k] = \sum_{n=0}^{N/2-1} x[2n]e^{-i2\pi k(2n)/N} + \sum_{n=0}^{N/2-1} x[2n+1]e^{-i2\pi k(2n+1)/N}$$

Step 2: Factor out common terms:

$$X[k] = \underbrace{\sum_{n=0}^{N/2-1} x[2n]e^{-i2\pi kn/(N/2)}}_{\text{DFT of even samples } X_{\text{even}}[k]} + e^{-i2\pi k/N} \underbrace{\sum_{n=0}^{N/2-1} x[2n+1]e^{-i2\pi kn/(N/2)}}_{\text{DFT of odd samples } X_{\text{odd}}[k]}$$

Step 3: Recursive decomposition:

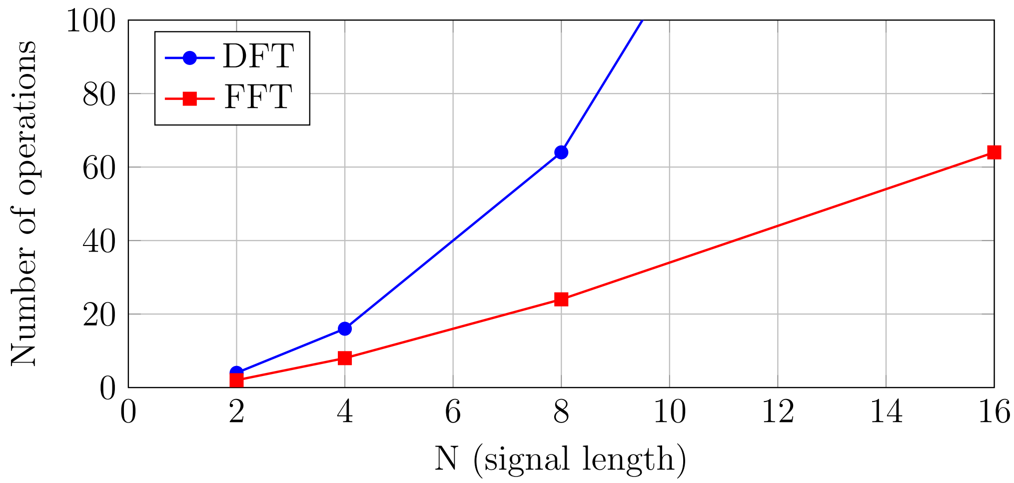
$$X[k] = X_{\text{even}}[k] + W_N^k X_{\text{odd}}[k], \quad k = 0, 1, \ldots, N-1$$

Where $W_N^k = e^{-i2\pi k/N}$ is the **twiddle factor**.

Step 4: Recursion reduces computation from $O(N^2)$ to $O(N \log_2 N)$.

9.2 DFT vs FFT Computational Complexity

- DFT: N^2 complex multiplications and $N(N-1)$ additions
- FFT: $N/2 \log_2 N$ complex multiplications, $N \log_2 N$ additions



This clearly shows FFT is much more efficient than DFT as N increases.

9.3 Radix-2 FFT (Decimation in Time)

Decimation-in-time (DIT): Split input into even and odd indices repeatedly. The process recursively reduces an N-point DFT into two $N/2$-point DFTs.

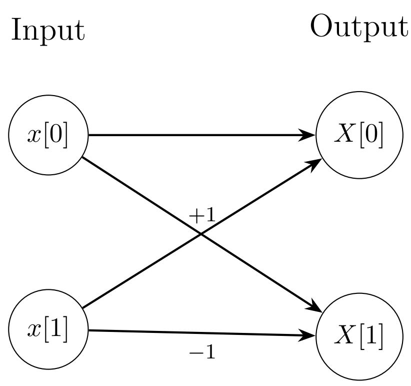
Step-by-step:

1. Split sequence into even-index and odd-index subsequences.
2. Compute DFTs of smaller sequences recursively.
3. Combine results using twiddle factors $W_N^k = e^{-i2\pi k/N}$.
4. Continue until 2-point DFTs (trivial addition/subtraction).

9.4 Butterfly Operation

2-point FFT: x_0, x_1

$$X[0] = x_0 + x_1, \quad X[1] = x_0 - x_1$$



Twiddle factor:
$W_2^0 = e^{-j2\pi \cdot 0/2} = 1$
(No twiddle factor applied in 2-point FFT)

Worked Example for 2-point FFT
Let the input signals be $x[0] = 1$ and $x[1] = 2$. The 2-point FFT computes the outputs $X[0]$ and $X[1]$ using the butterfly diagram above. The equations for a 2-point FFT are:

$$X[0] = x[0] + x[1], \quad X[1] = x[0] - x[1]$$

Substituting the input values:

$$X[0] = x[0] + x[1] = 1 + 2 = 3$$

$$X[1] = x[0] - x[1] = 1 - 2 = -1$$

Thus, the outputs are:

$$X[0] = 3, \quad X[1] = -1$$

Explanation: The butterfly diagram shows that $X[0]$ is the sum of the inputs $x[0]$ and $x[1]$ (both with weight $+1$), while $X[1]$ is the difference, where $x[0]$ is multiplied by $+1$ and $x[1]$ by -1. No twiddle factor is needed in a 2-point FFT since $W_2^0 = 1$.

9.5 6. Key Terms

- **Decimation-in-time:** Rearranging input into even/odd indices for recursive DFT computation.
- **Twiddle factor:** $W_N^k = e^{-i2\pi k/N}$, a complex rotation used to combine smaller DFTs.
- **Normalized angular frequency:** $\omega = 2\pi f/f_s$, represents frequency in radians/sample.
- **Binary bit-reversal:** Reordering input indices according to reversed binary representation for FFT algorithm efficiency.

  Example (N=8):

  $$n = 0\ (000)_2 \to 0,\ n = 1\ (001)_2 \to 4,\ n = 2\ (010)_2 \to 2, \ldots$$