NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Module 9: Fourier Transform

Learning Objectives

- Learn discrete Fourier transform (DFT)
- Learn fast Fourier transform (FFT)
- Learn trigonometric interpolation

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Sources

- Textbook (Chapter 10: Trigonometric Interpolation and the FFT)
- Ref: J.Cooley and J.Tukey (1965), "An algorithm for the Machine Calculation of Complex Fourier Series", *Mathematics of Computation* **19**, 297-301.
- Wiki: Fourier Transform https://en.wikipedia.org/wiki/Fourier_transform
- Wiki: Fast Fourier Transform https://en.wikipedia.org/wiki/Fast_Fourier_transform
- Wiki: JPEG https://en.wikipedia.org/wiki/JPEG

Outline

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Introduction

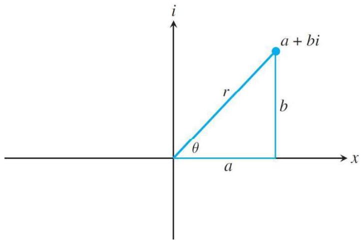
- Sines and Cosines have great impact on modern technology.
- Trigonometric functions of multiple frequencies are natural interpolating functions for periodic data.
- Fourier transform is efficient at carrying out the interpolation and plays an essential role in signal processing.
- Fast Fourier transform makes discrete Fourier transform computationally cheap.

Outline

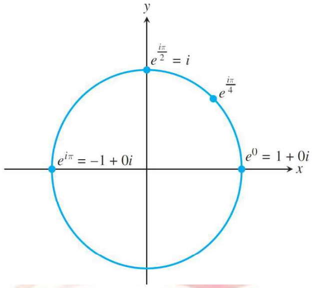
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Complex arithmetic

- **Complex number:** $z = a + bi = re^{i\theta}$
  - $i = \sqrt{-1}$
  - magnitude $r = \sqrt{a^2 + b^2}$
  - angle $\theta = \arctan\dfrac{b}{a}$

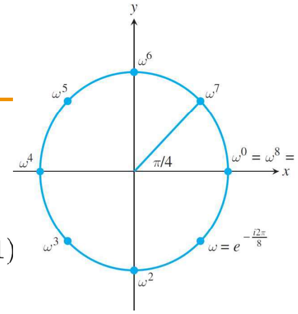


- **Euler formula:** $e^{i\theta} = \cos\theta + i\sin\theta$



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Roots of unity



- **nth root of unity:** solution of $z^n = 1$

  $$\omega_n = e^{-i\frac{2\pi}{n}k}: \quad e^{-i\frac{2\pi}{n}}, e^{-i\frac{4\pi}{n}}, e^{-i\frac{6\pi}{n}}, \cdots, e^{-i\frac{2n\pi}{n}} (= 1)$$

- **Primitive root of unity:** an nth root of unity is primitive if it is not a kth root of unity for any $0 < k < n$. E.g., $\omega = e^{-i\frac{2\pi}{n}}$.

- **Identity:** If ω is a primitive nth root of unity $\omega = e^{-i\frac{2\pi}{n}}$ for $n > 1$, then

$$\begin{aligned}
1 + \omega + \omega^2 + \omega^3 + \cdots + \omega^{n-1} &= 0 \\
1 + \omega^2 + \omega^4 + \omega^6 + \cdots + \omega^{2(n-1)} &= 0 \\
1 + \omega^3 + \omega^6 + \omega^9 + \cdots + \omega^{3(n-1)} &= 0 \\
&\vdots \\
1 + \omega^{n-1} + \omega^{2(n-1)} + \omega^{3(n-1)} + \cdots + \omega^{(n-1)(n-1)} &= 0 \\
1 + \omega^n + \omega^{2n} + \omega^{3n} + \cdots + \omega^{(n-1)n} &= n
\end{aligned}$$

Outline

Discrete Fourier Transform (DFT)

- Let $\omega = e^{-i\frac{2\pi}{n}}$.

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} \xrightarrow[\text{Discrete Fourier Transform}]{y_j = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} x_k \omega^{jk}} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

- In matrix form,

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} a_0 + ib_0 \\ a_1 + ib_1 \\ a_2 + ib_2 \\ \vdots \\ a_{n-1} + ib_{n-1} \end{bmatrix} = \underbrace{\frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix}}_{\text{Fourier Matrix: } F_n} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

Property

- Let $\{y_j\}$ be the DFT of $\{x_j\}$. If all x_j are real number, then

  (1). y_0 is real.

  (2). for $j = 1, \cdots, n-1$, $\boxed{y_{n-j} = \bar{y}_j,}$ i.e., $\boxed{a_{n-j} + ib_{n-j} = a_j - ib_j.}$

  This is because

$$y_{n-j} = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} x_k (\omega^{n-j})^k = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} x_k (\overline{\omega^j})^k = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \overline{x_k (\omega^j)^k} = \overline{y_j}.$$

- For example, if $x_0, \cdots, x_7$ are real numbers, then

$$F_8 \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 + ib_1 \\ a_2 + ib_2 \\ a_3 + ib_3 \\ a_4 \\ a_3 - ib_3 \\ a_2 - ib_2 \\ a_1 - ib_1 \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_{\frac{n}{2}-1} \\ y_{\frac{n}{2}} \\ \overline{y}_{\frac{n}{2}-1} \\ \vdots \\ \overline{y}_1 \end{bmatrix}$$

Example

- Find the DFT of the vector x = $[1,0,-1,0]^T$.

Let ω be the 4th root of unity, or $\omega = e^{-i\pi/2} = \cos(\pi/2) - i\sin(\pi/2) = -i$. Applying the DFT, we get

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

Inverse Discrete Fourier Transform

- DFT can be written in compact form as $y = F_n x$.
- The **Inverse Discrete Fourier Transform** of the vector y can be written as $x = F_n^{-1} y$, where the inverse matrix is

$$F_n^{-1} = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \omega^0 & \omega^{-3} & \omega^{-6} & \cdots & \omega^{-3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)^2} \end{bmatrix}$$

Explanation of DFT

DFT can be viewed as the solution (i.e., the coefficients) of the following interpolating problem:

Given sample data $x_0, x_1, x_2, \cdots, x_{n-1}$ corresponding to parameter values $t = 0, \frac{2\pi}{n}, \frac{4\pi}{n}, \cdots, \frac{2(n-1)\pi}{n}$, find a complex function:

$$Q(t) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} y_k (e^{it})^k$$

such that

$$Q(t_j) = Q(\frac{2\pi j}{n}) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} y_k (e^{i\frac{2\pi j}{n}})^k = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} y_k (\omega^{-j})^k = x_j.$$

In fact, the interpolating conditions are equivalent to $F_n^{-1} y = x$.

Outline

Fast Fourier Transform (FFT)

- The naive DFT method applied to an n-vector requires $O(n^2)$ operations.
- Cooley and Tukey [1965] found the *Fast Fourier Transform (FFT)* to compute the DFT requires only $O(n \log n)$ opertations, which makes the field of signal processing convert from primarily analog to digital.
- The key to the FFT is the fact that

$$F_{2n}[x_0, x_1, x_2, x_3, \cdots, x_{2n-2}, x_{2n-1}]_j = \underline{F_n[x_0, x_2, \cdots, x_{2n-2}]_j} + e^{-i\frac{j\pi}{n}} \underline{F_n[x_1, x_3, \cdots, x_{2n-1}]_j}$$

$$F_{2n}[x_0, x_1, x_2, x_3, \cdots, x_{2n-2}, x_{2n-1}]_{n+j} = \underline{F_n[x_0, x_2, \cdots, x_{2n-2}]_j} - e^{-i\frac{j\pi}{n}} \underline{F_n[x_1, x_3, \cdots, x_{2n-1}]_j}$$

for $j = 0, 1, \cdots, n-1$, where

- $F_n[\cdots]$ represents the vector of scaled DFT of vector $[\cdots]$ with the scale $\sqrt{n}$.
- the sub-indices j simply refer to the j-th entry of the vector.

FFT

That is, the scaled DFT of the $2n$ data $x_0, x_n, x_1, x_{n+1}, \cdots, x_{n-1}, x_{2n-1}$ can be obtained by combining the scaled DFT's of the two n data sets $x_0, x_1, \cdots, x_{n-1}$ and $x_n, x_{n+1}, \cdots, x_{2n-1}$ componentwise.

For example, consider the DFT of vector $[x_0, x_1, x_2, \cdots, x_7]$. We can perform the following calculations in an inverse order:

- $F_8[x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7]$
- $F_4[x_0, x_2, x_4, x_6]$ and $F_4[x_1, x_3, x_5, x_7]$
- $F_2[x_0, x_4]$, $F_2[x_2, x_6]$ and $F_2[x_1, x_5]$, $F_2[x_3, x_7]$
- $F_1[x_0], F_1[x_4], F_1[x_2], F_1[x_6], F_1[x_1], F_1[x_5], F_1[x_3], F_1[x_7]$.

Note that $F_1[x_j] = x_j$. After the 3 rounds of computation, we scale the obtained coefficients with $\frac{1}{\sqrt{8}}$, which gives the DFT of vector $[x_0, x_1, \cdots, x_7]$.

FFT: example

To verify the FFT formula by calculating the scaled DFT of vector $[x_0, x_1, x_2, x_3]$.

Let $\omega_4 = e^{-i\frac{2\pi}{4}} = -i$ and $\omega_2 = e^{-i\frac{2\pi}{2}} = -1$.

- First, calculate the scaled DFT of $[x_0, x_2]$.

$$\begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = \begin{bmatrix} \omega_2^0 & \omega_2^0 \\ \omega_2^0 & \omega_2^1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_0 + x_2 \\ x_0 - x_2 \end{bmatrix}$$

- Second, calculate the scaled DFT of $[x_1, x_3]$.

$$\begin{bmatrix} v_0 \\ v_1 \end{bmatrix} = \begin{bmatrix} \omega_2^0 & \omega_2^0 \\ \omega_2^0 & \omega_2^1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 + x_3 \\ x_1 - x_3 \end{bmatrix}$$

FFT: example (cont)

- Third, calculate the scaled DFT of $[x_0, x_1, x_2, x_3]$.

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \omega_4^0 & \omega_4^0 & \omega_4^0 & \omega_4^0 \\ \omega_4^0 & \omega_4^1 & \omega_4^2 & \omega_4^3 \\ \omega_4^0 & \omega_4^2 & \omega_4^4 & \omega_4^6 \\ \omega_4^0 & \omega_4^3 & \omega_4^6 & \omega_4^9 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_0 + x_1 + x_2 + x_3 \\ x_0 - ix_1 - x_2 + ix_3 \\ x_0 - x_1 + x_2 - x_3 \\ x_0 + ix_1 - x_2 - ix_3 \end{bmatrix}$$

$$= \begin{bmatrix} (x_0 + x_2) + (x_1 + x_3) \\ (x_0 - x_2) - i(x_1 - x_3) \\ (x_0 + x_2) - (x_1 + x_3) \\ (x_0 - x_2) + i(x_1 - x_3) \end{bmatrix} = \begin{bmatrix} u_0 + v_0 \\ u_1 - iv_1 \\ u_0 - v_0 \\ u_1 + iv_1 \end{bmatrix}$$

That is,

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} z_0 \\ z_1 \\ z_{2+0} \\ z_{2+1} \end{bmatrix} = \begin{bmatrix} u_0 + v_0 \\ u_1 + e^{-i\frac{\pi}{2}} v_1 \\ u_0 - v_0 \\ u_1 - e^{-i\frac{\pi}{2}} v_1 \end{bmatrix}$$

Outline

Trigonometric interpolation

Problem: The set of functions $\{1, \cos\theta, \sin\theta, \cos 2\theta, \sin 2\theta, \cdots\}, \theta \in [0, 2\pi]$ spans a space. Given n data points $x_0, x_1, \cdots, x_{n-1}$, let $\Delta\theta = \frac{2\pi}{n}$ and $\theta_j = j\Delta\theta$ for $j = 0, 1, \cdots, n-1$. Now we want to find a function $P_n(\theta)$ as a linear combination of the first n basis functions in the set, such that $P_n(\theta_j) = x_j$.

(1) Denote the DFT of $[x_0, x_1, \cdots, x_{n-1}]$ by $y_j = a_j + ib_j, j = 0, 1, \cdots, n-1$. Construct the complex function:

$$
\begin{aligned}
Q(\theta) &= \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} (a_k + ib_k) e^{ik\theta} = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} (a_k + ib_k)(\cos k\theta + i \sin k\theta) \\
&= \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} (a_k \cos k\theta - b_k \sin k\theta) + i \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} (b_k \cos k\theta + a_k \sin k\theta) \\
&\triangleq P(\theta) + iI(\theta)
\end{aligned}
$$

We have already known that $Q(\theta_j) = x_j$.

Trigonometric interpolation (cont)

(2) If all the x_j are real, the real function

$$P(\theta) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} (a_k \cos k\theta - b_k \sin k\theta)$$

satisfies $P(\theta_j) = x_j$ for $j = 0, 1, \cdots, n-1$. However, $P(\theta)$ contains $2n$ terms.

(3) Assume that n is even. Note that for $j = 0, 1, \cdots, n-1$,

$$\cos k\theta_j = \cos(n-k)\theta_j, \quad \sin k\theta_j = -\sin(n-k)\theta_j, \quad \text{for} \quad k = 1, 2, \cdots, \frac{n}{2} - 1$$

$$0,\ 1,\ 2,\ \cdots,\ \tfrac{n}{2}-1,\ \tfrac{n}{2},\ \tfrac{n}{2}+1, \cdots, n-2,\ n-1$$

Hence construct

$$P_n(\theta) = \frac{1}{\sqrt{n}} a_0 + \frac{1}{\sqrt{n}} \sum_{k=1}^{\frac{n}{2}-1} [(a_k + a_{n-k}) \cos k\theta - (b_k - b_{n-k}) \sin k\theta] + \frac{1}{\sqrt{n}} a_{\frac{n}{2}} \cos \frac{n}{2}\theta$$

satisfying $P_n(\theta_j) = P(\theta_j) = x_j$.

Trigonometric interpolation (cont)

(4) For real x_j, we have $a_{n-j} = a_j, b_{n-j} = -b_j$. Thus

$$P_n(\theta) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{\frac{n}{2}-1} [a_k \cos k\theta - b_k \sin k\theta] + \frac{a_{\frac{n}{2}}}{\sqrt{n}} \cos \frac{n}{2}\theta$$

(5) Similar development for odd n.

(6) By reparameterization, we obtain general trigonometric interpolation scheme. That is, given an interval $[c, d]$ and positive integer n, let $t_j = c + j(d-c)/n$ for $j = 0, \cdots, n-1$. The vector $x = (x_0, \cdots, x_{n-1})$ consists of n *real* numbers. Let $\{a_j + ib_j\}$ denote the DFT $F_n x$ of x. Then an *oder* n *trigonometric interpolating function* is

$$P_n(t) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{\frac{n}{2}-1} \left[a_k \cos \frac{2\pi(t-c)}{d-c} k - b_k \sin \frac{2\pi(t-c)}{d-c} k \right] + \frac{a_{\frac{n}{2}}}{\sqrt{n}} \cos \frac{\pi(t-c)}{d-c} n$$

for $t \in [c, d]$.

Extensions

- Extension to least squares fitting with trigonometric functions
- Extension to Discrete Cosine Transform (DCT)
- Extension to 2-dimensional DCT
- Applications in image compression (such as JPEG compression)
- …

Outline

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Homework

Find an order 8 trigonometric interpolant for data $[0, 1, 0, -1, 0, 1, 0, -1]$ on the interval $[0, 2\pi]$.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Outline

§1. Introduction
§2. Complex arithmetic
§3. Discrete Fourier transform (DFT)
§4. Fast Fourier transform (FFT)
§4. Trigonometric interpolation
§7. Homework
§8. Summary

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Summary

- Discrete Fourier transform
  - provide a tool for many applications such as trigonometric interpolation
- Fast Fourier transform
  - provide a fast way to calculate DFT
- Trigonometric interpolation

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

End

$$\{1, \cos(2\pi u), \sin(2\pi u), \cos(4\pi u), \sin(4\pi u), \cos(6\pi u), \sin(6\pi u), \cos(8\pi u)\}$$