

Introduction of Fast Fourier Transform¹

Tobramycin

NCUT

2022.7



¹J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. of Computation, vol. 19, pp. 297–301, 1965

Contents

1 FFT Experiment

2 FFT Introduction

FFT Experiment

FFT Input and Output

真实的函数：

$$S = 2 + 3 * \cos(2 * \pi * 50 * t - \pi * 30 / 180) + 1.5 * \cos(2 * \pi * 75 * t + \pi * 90 / 180) \quad (1)$$

采样点数：设 1s 内采样点为 256 个，则采样频率为 256Hz，分辨 1Hz 频率

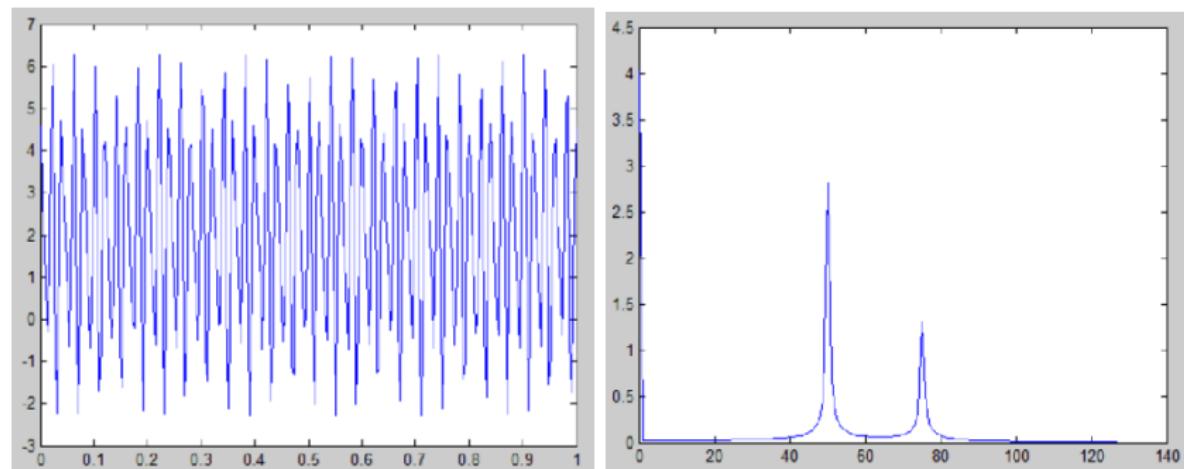


Figure: Origin and Spectrum

Output 计算公式

FFT 幅度：毕达哥拉斯公式

$$\text{Magnitude} = \sqrt{\text{Real}^2 + \text{Imaginary}^2} \quad (2)$$

FFT 相位：反正切规则

$$\text{Phase} = \tan^{-1} \left(\frac{\text{Imaginary}}{\text{Real}} \right) \quad (3)$$

Output 计算结果

fft input: [256,], fft output: [256,], 采样定理: 采样频率 $\leq 2 \times$ 信号频率



Figure: FFT Result

- 0Hz 分量复数表示: 511.999
- 50Hz 分量复数表示: $332 - 191j$
- 75Hz 分量复数表示: $3.44e^{12} + 191j$
- 0Hz 模值表示: $511.999 * \frac{2}{N} = 4$
- 50Hz 模值表示: $384 * \frac{2}{N} = 3$
- 75Hz 模值表示: $192 * \frac{2}{N} = 1.5$

类似, 相位计算得: 0, -30, 90

因此, 原函数

$$S = 2 + 3 * \cos(2 * \pi * 50 * t - \pi * 30 / 180) + 1.5 * \cos(2 * \pi * 75 * t + \pi * 90 / 180)$$

Tobramycin (mail)

FFT Introduction

FFT Introduction

傅立叶变换 (FT): 连续时间 & 连续频率

$$X(f) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-i2\pi ft} \cdot dt \quad (4)$$

离散傅立叶变换 (DFT): 离散时间 & 离散频率

$$X(k) = \sum_{n=0}^{N-1} x(n) \times e^{-i2\pi \frac{k}{N} n}, k = 0, 1, \dots, N - 1 \quad (5)$$

n 表示离散时间, k 表示离散频率

1. Vandermonde Matrix

$(d + 1)$ points uniquely define a degree d polynomial

$$\{(x_0, P(x_0)), (x_1, P(x_1)), \dots, (x_d, P(x_d))\}$$

$$P(x) = p_0 + p_1x + p_2x^2 + \cdots + p_dx^d$$

$$\begin{bmatrix} P(x_0) \\ P(x_1) \\ \vdots \\ P(x_d) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^d \\ 1 & x_1 & x_1^2 & \cdots & x_1^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_d & x_d^2 & \cdots & x_d^d \end{bmatrix}}_M \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_d \end{bmatrix}$$

M

M is invertible for unique x_0, x_1, \dots, x_d

Figure: Vandermonde Matrix²

²https://www.youtube.com/watch?v=h7apO7q16V0&feature=emb_logo

1. Vandermonde Matrix

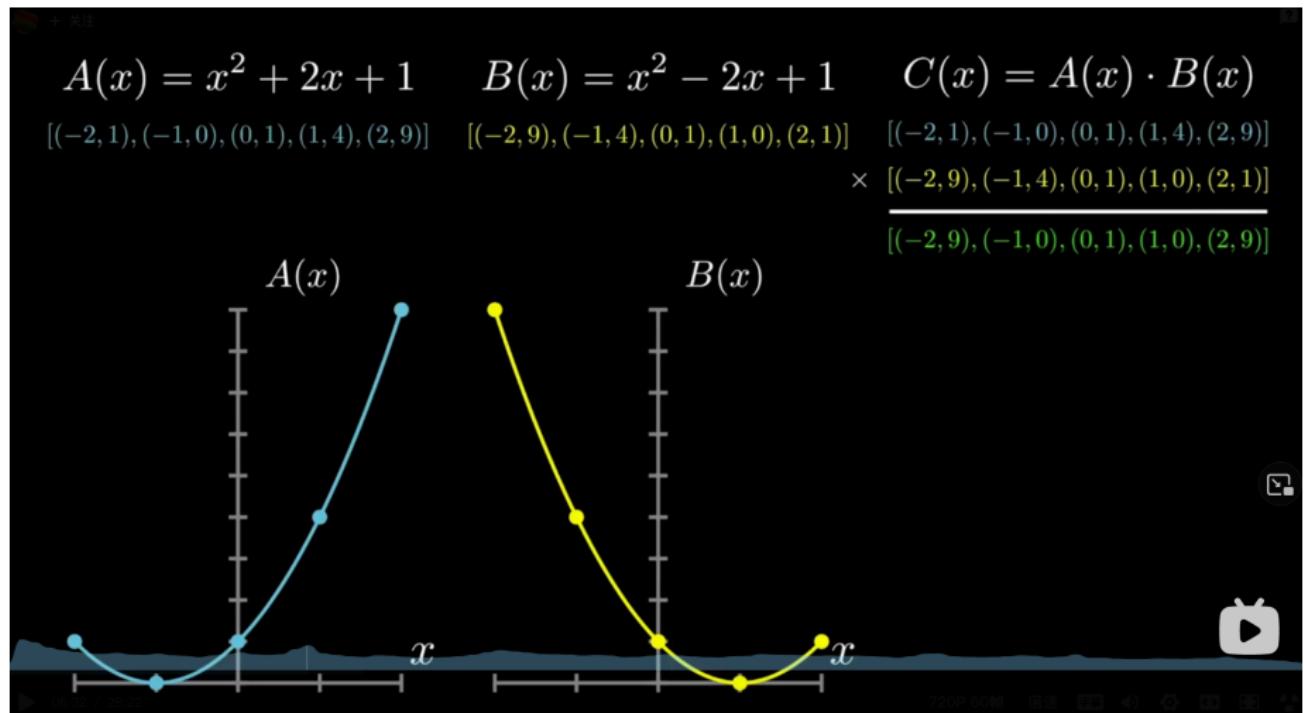


Figure: 点值表示法³

2. 对称结构

$$P(x) = p_0 + p_1x + p_2x^2 + \cdots + p_{n-1}x^{n-1}$$

Evaluate at n points $\pm x_1, \pm x_2, \dots, \pm x_{n/2}$

$$P(x) = P_e(x^2) + xP_o(x^2)$$

$$\left. \begin{array}{l} P(x_i) = P_e(x_i^2) + x_i P_o(x_i^2) \\ P(-x_i) = P_e(x_i^2) - x_i P_o(x_i^2) \end{array} \right\} \text{Lot of overlap!}$$

$P_e(x^2)$ and $P_o(x^2)$ have degree $n/2 - 1$!

Evaluate $P_e(x^2)$ and $P_o(x^2)$ each at $x_1^2, x_2^2, \dots, x_{n/2}^2$ ($n/2$ points)

Same process on simpler problem

Figure: 对称结构⁴

⁴https://www.youtube.com/watch?v=h7apO7q16V0&feature=emb_logo

2. 对称结构

Evaluate $P(x) : [p_0, p_1, \dots, p_{n-1}]$
 $[\pm x_1, \pm x_2, \dots, \pm x_{n/2}]$

Evaluate $P_e(x^2) : [p_0, p_2, \dots, p_{n-2}]$
 $[x_1^2, x_2^2, \dots, x_{n/2}^2]$
 $[P_e(x_1^2), P_e(x_2^2), \dots, P_e(x_{n/2}^2)]$

Evaluate $P_o(x^2) : [p_1, p_3, \dots, p_{n-1}]$
 $[x_1^2, x_2^2, \dots, x_{n/2}^2]$
 $[P_o(x_1^2), P_o(x_2^2), \dots, P_o(x_{n/2}^2)]$

Points $[\pm x_1, \pm x_2, \dots, \pm x_{n/2}]$ are \pm paired.

Points $[x_1^2, x_2^2, \dots, x_{n/2}^2]$ are not \pm paired.

Recursion breaks!

Is it possible to make $[x_1^2, x_2^2, \dots, x_{n/2}^2]$ \pm paired?

Figure: 对称结构⁵

⁵https://www.youtube.com/watch?v=h7apO7q16V0&feature=emb_logo

3. 复数引入

$$P(x) = x^3 + x^2 - x - 1$$

Alternative Perspective

Solution to $x^4 = 1$

Points are 4th roots of unity!

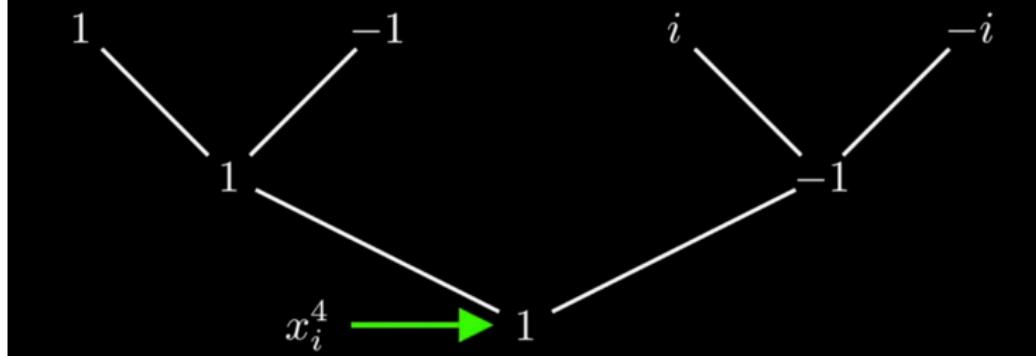


Figure: 复数引入⁶

3. 复数引入

$$P(x) = x^5 + 2x^4 - x^3 + x^2 + 1$$

Need $n \geq 6$ points → let $n = 8$ (powers of 2 are convenient)

Points are 8th roots of unity!

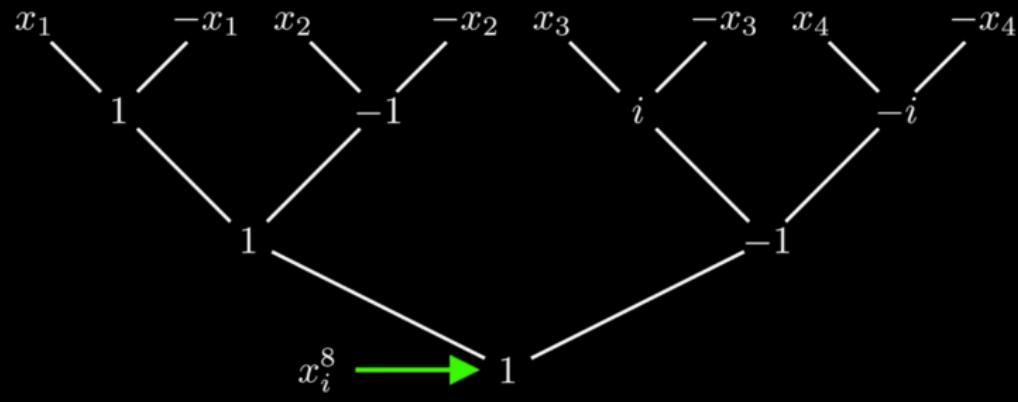


Figure: 复数引入⁷

⁷https://www.youtube.com/watch?v=h7apO7q16V0feature=emb_logo

4. 如何选取复数采样点

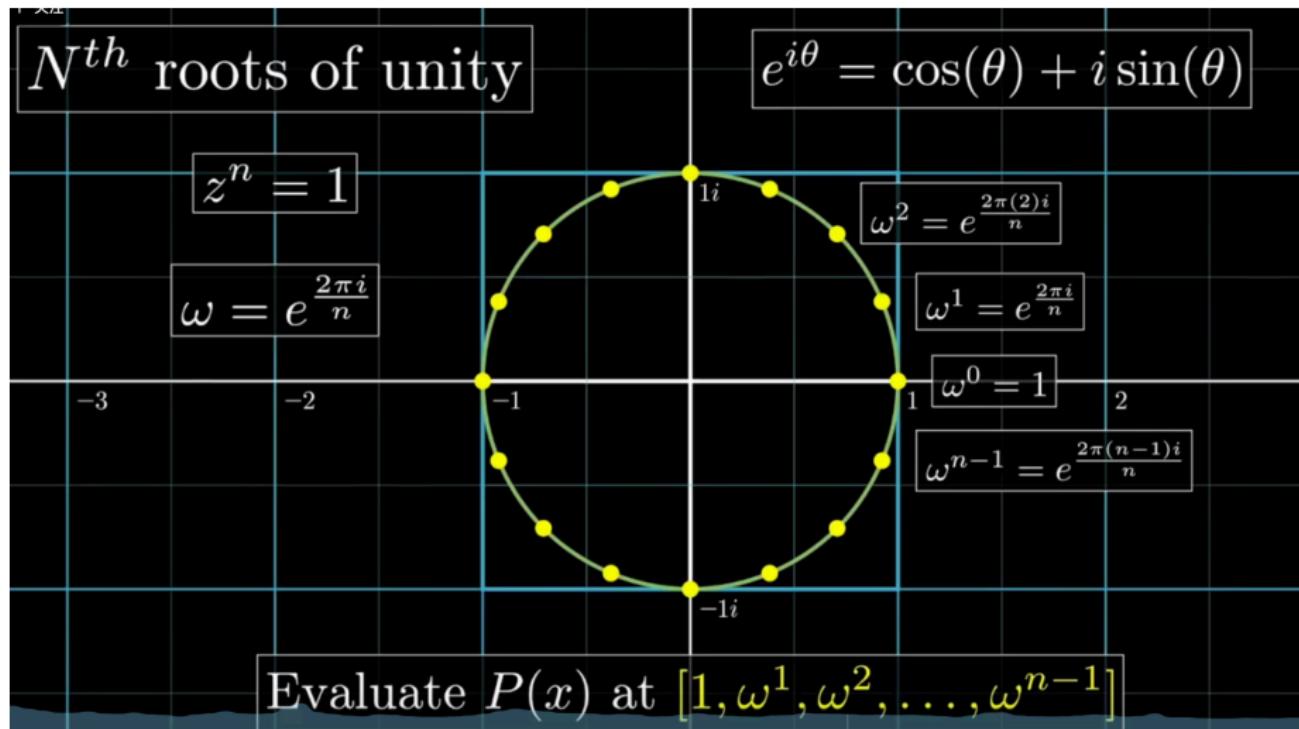
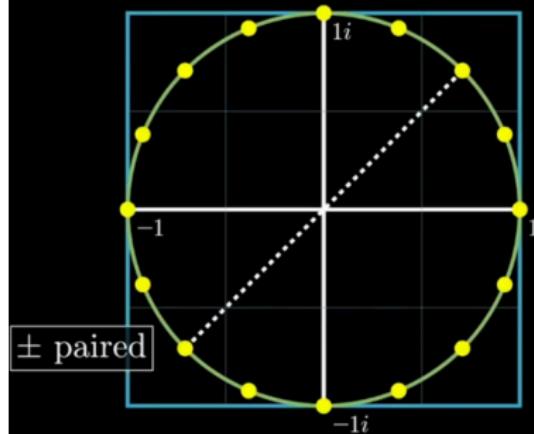


Figure: 复数引入⁸

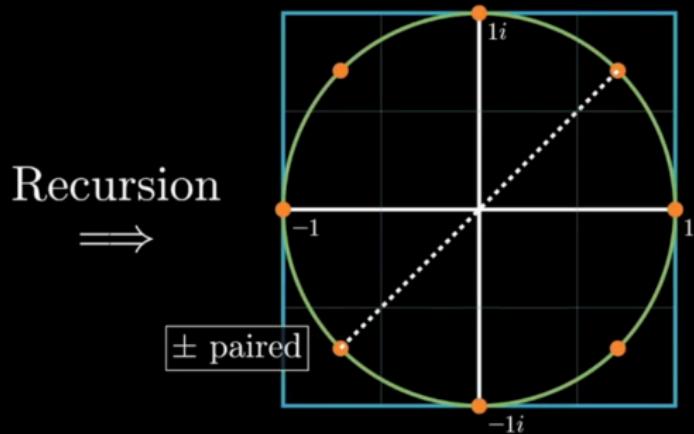
4. 如何选取复数采样点

Why does this work?

$\omega^{j+n/2} = -\omega^j \rightarrow (\omega^j, \omega^{j+n/2})$ are \pm paired



Evaluate $P(x)$ at $[1, \omega^1, \omega^2, \dots, \omega^{n-1}]$
 n roots of unity



Evaluate $P_e(x^2)$ and $P_o(x^2)$ at
 $[1, \omega^2, \omega^4, \dots, \omega^{2(n/2-1)}]$
 $(n/2)$ roots of unity

FFT Pseudocode

```
def FFT(P) :  
    # P - [ $p_0, p_1, \dots, p_{n-1}$ ] coeff representation  
    n = len(P) # n is a power of 2  
    if n == 1:  
        return P  
     $\omega = e^{\frac{2\pi i}{n}}$   
    Pe, Po = [ $p_0, p_2, \dots, p_{n-2}$ ], [ $p_1, p_3, \dots, p_{n-1}$ ]  
    ye, yo = FFT(Pe), FFT(Po)  
    y = [0] * n  
    for j in range(n/2):  
        y[j] = ye[j] +  $\omega^j y_o[j]$   
        y[j + n/2] = ye[j] -  $\omega^j y_o[j]$   
    return y
```

Figure: FFT Pseudocode¹⁰

¹⁰https://www.youtube.com/watch?v=h7apO7q16V0feature=emb_logo

IFFT Vandermonde Matrix

Evaluation (FFT)

$$\text{FFT}([p_0, p_1, \dots, p_{n-1}]) \rightarrow [P(\omega^0), P(\omega^1), \dots, P(\omega^{n-1})]$$

$$\begin{bmatrix} P(\omega^0) \\ P(\omega^1) \\ P(\omega^2) \\ \vdots \\ P(\omega^{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{n-1} \end{bmatrix}$$

FFT(<coeffs>) defined $\omega = e^{\frac{2\pi i}{n}}$

Interpolation (Inverse FFT)

$$\text{IFFT}([P(\omega^0), P(\omega^1), \dots, P(\omega^{n-1})]) \rightarrow [p_0, p_1, \dots, p_{n-1}]$$

$$\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{n-1} \end{bmatrix} = \frac{1}{n} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} P(\omega^0) \\ P(\omega^1) \\ P(\omega^2) \\ \vdots \\ P(\omega^{n-1}) \end{bmatrix}$$

]

Figure: IFFT Vandermonde Matrix ¹¹

¹¹https://www.youtube.com/watch?v=h7apO7q16V0feature=emb_logo

IFFT Pseudocode



IFFT(<values>) \Leftrightarrow FFT(<values>) with $\omega = \frac{1}{n} e^{-\frac{2\pi i}{n}}$

```

def FFT(P) :
    # P - [p0, p1, ..., pn-1] coeff rep
    n = len(P) # n is a power of 2
    if n == 1:
        return P
    ω =  $e^{\frac{2\pi i}{n}}$ 
    Pe, Po = P[::2], P[1::2]
    ye, yo = FFT(Pe), FFT(Po)
    y = [0] * n
    for j in range(n/2):
        y[j] = ye[j] + ωj yo[j]
        y[j + n/2] = ye[j] - ωj yo[j]
    return y

def IFFT(P) :
    # P - [P( $\omega^0$ ), P( $\omega^1$ ), ..., P( $\omega^{n-1}$ )] value rep
    n = len(P) # n is a power of 2
    if n == 1:
        return P
    ω = (1/n) *  $e^{-\frac{2\pi i}{n}}$ 
    Pe, Po = P[::2], P[1::2]
    ye, yo = IFFT(Pe), IFFT(Po)
    y = [0] * n
    for j in range(n/2):
        y[j] = ye[j] + ωj yo[j]
        y[j + n/2] = ye[j] - ωj yo[j]
    return y

```

Figure: FFT Pseudocode¹²

¹²https://www.youtube.com/watch?v=h7apO7q16V0&feature=emb_logo

复数采样点的另一个优势

Interpolation

Alternative Perspective on Evaluation/FFT

$$P(x) = p_0 + p_1x + p_2x^2 + \cdots + p_{n-1}x^{n-1}$$

$$\begin{bmatrix} P(\omega^0) \\ P(\omega^1) \\ P(\omega^2) \\ \vdots \\ P(\omega^{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{n-1} \end{bmatrix}$$

$$x_k = \omega^k \text{ where } \omega = e^{\frac{2\pi i}{n}}$$

Figure: 另一个优势¹³

¹³https://www.youtube.com/watch?v=h7apO7q16V0&feature=emb_logo

Thank you!