

信号与系统

王洪广

wanghg@mail.xjtu.edu.cn

wanghg.gr.xjtu.edu.cn

7.1 离散傅立叶变换 (DFT) : (Discrete Fourier Transform)

- 离散时间傅立叶变换是连续函数。为了在频域进行数字处理，需要将其离散化，即需要一种时域离散、频域也离散的关系，DFS正满足这一点。

- DFS的本质是将时域N个独立的点变换为频域N个独立的点，将一个周期序列取主周期，即得N点有限长序列，对 A_k 取其主周期也是N个独立的点。

可以认为时域N点有限长序列与频域N点有限长序列之间有一种变换关系，这种关系就称为DFT。

一. 从DFS到 DFT:

若 $x(n)$ 为有限长序列, 将其周期性延拓成以 N 为周期的序列 $\tilde{x}(n)$ 则:

$$\tilde{x}(n) = \sum_{k=-\infty}^{\infty} x(n - kN) \quad \text{或表示为} \quad \tilde{x}(n) = x((n))_N$$

若 $R_N(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \textit{otherwise} \end{cases}$ 则:

$$x(n) = \tilde{x}(n)R_N(n)$$

由DFS有:

$$\tilde{x}(n) = \sum_{k=\langle N \rangle} \dot{A}_k e^{j\frac{2\pi}{N}kn} = \sum_{k=0}^{N-1} \dot{A}_k e^{j\frac{2\pi}{N}kn}$$

$$\dot{A}_k = \frac{1}{N} \sum_{n=\langle N \rangle} \tilde{x}(n) e^{-j\frac{2\pi}{N}kn} = \frac{1}{N} \sum_{n=0}^{N-1} \tilde{x}(n) e^{-j\frac{2\pi}{N}kn}$$

将 $N\dot{A}_k$ 记为 $\tilde{X}(k)$, 且 $W_N = e^{-j\frac{2\pi}{N}}$, 则DFS可表示为:

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

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

—— DFS的另一种表示形式

如果取 $\tilde{X}(k)$ 的主值周期 $k = 0 \sim N-1$, 并记为 $X(k)$, 同时取 $\tilde{x}(n)$ 的主值周期 $n = 0 \sim N-1$, 并记为 $x(n)$ 则有:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}, \quad 0 \leq n \leq N-1$$
$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad 0 \leq k \leq N-1$$

这一对关系就是有限长序列与它的DFT.

DFT表明：时域的 N 点有限长序列 $x(n)$ ，可以变换为频域的 N 点有限长序列 $X(k)$ 。

显然有： $X(k) = \tilde{X}(k) R_N(k)$

$\tilde{X}(k) = X((k))_N$ —— DFT与DFS的关系

二. DFT与频域采样的关系:

对 N 点有限长序列 $x(n)$ 有:

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x(n)e^{-j\omega n} \quad \text{令 } \omega = \frac{2\pi}{N}k \text{ 则有:}$$

$$X(e^{j\frac{2\pi}{N}k}) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn}$$

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

$$\therefore X(k) = X(e^{j\omega}) \Big|_{\omega = \frac{2\pi}{N}k} \quad k = 0 \sim N-1$$

表明：有限长序列的DFT就是对其离散时间傅立叶变换在一个周期内等间隔采样的样本。

应该强调指出：**DFT并不是 $x(n)$ 的频谱，只是其频谱的样本。** DFT在一定程度上反映了 $x(n)$ 的频谱。只有在满足频域采样的要求时，DFT才可以完全代表信号的频谱。

频域采样 \longrightarrow 时域周期性延拓，要求 $x(n)$ 时限。若 $x(n)$ 有M点，对其频谱在一个周期内采样N点，那么， $x(n)$ 将以N为周期延拓。**只有当 $N \geq M$ 时，这种延拓才不会发生混叠，**即在一个周期内至少要采样M点才能恢复原信号。

若 $x(n)$ 非时限，则不论在频谱的一个周期采样多少点，都无法从周期延拓的信号中恢复原 $x(n)$ 。所以，**DFT 只能对应有限长序列。**

• 考察有限长系列当满足频域采样要求时，在时域和频域的恢复过程：

若 $x(n)$ 是一个 M 点的序列，在频域对其采样 N 点，则有：

$$X(k) = \sum_{n=0}^{M-1} x(n)W_N^{kn} = \sum_{n=-\infty}^{\infty} x(n)W_N^{kn}$$

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

$$\tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{m=-\infty}^{\infty} x(m) W_N^{km} \cdot W_N^{-kn}$$

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

$$\sum_{k=0}^{N-1} W_N^{k(m-n)} = \begin{cases} N, & m = n + rN, r \text{ 为整数} \\ 0, & \text{其他 } m \end{cases}$$

$$\therefore \tilde{x}(n) = \sum_{r=-\infty}^{\infty} x(n + rN)$$

当 $N \geq M$ 时可以通过矩形窗从 $\tilde{x}(n)$ 恢复 $x(n)$ 。

$$x(n) = \tilde{x}(n) \cdot R_N(n)$$

频谱的恢复:

$$X(e^{j\omega}) = \frac{1}{2\pi} \tilde{X}(e^{j\omega}) \otimes \phi(e^{j\omega}) \quad \phi(e^{j\omega}) \leftrightarrow R_N(n)$$

$$\tilde{X}(e^{j\omega}) = \frac{2\pi}{N} \sum_{k=-\infty}^{\infty} \tilde{X}(k) \delta\left(\omega - \frac{2\pi}{N}k\right)$$

————— $\tilde{x}(n)$ 的DTFT

$$\phi(e^{j\omega}) = \sum_{n=0}^{N-1} e^{-j\omega n} = \frac{\sin \omega N/2}{\sin \omega/2} e^{-j\frac{N-1}{2}\omega}$$

做周期卷积时, 将积分区间取为 $0 \sim 2\pi$, 在此区间内

$$\tilde{X}(k) = X(k)$$

于是：

$$X(e^{j\omega}) = \sum_{k=0}^{N-1} X(k) \frac{1}{N} \phi(e^{j(\omega - \frac{2\pi}{N}k)})$$

令：

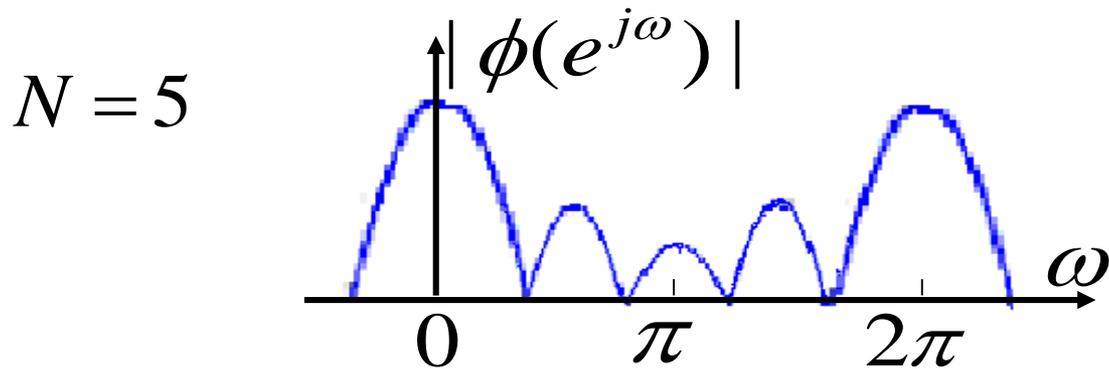
$$\Phi_k(e^{j\omega}) = \frac{1}{N} \phi(e^{j(\omega - \frac{2\pi}{N}k)})$$

$$= \frac{1}{N} \frac{\sin(\omega N / 2)}{\sin[(\omega - \frac{2\pi k}{N}) / 2]} e^{-j(\frac{N-1}{2}\omega + \frac{k\pi}{N})}$$

则：

$$X(e^{j\omega}) = \sum_{k=0}^{N-1} X(k) \Phi_k(e^{j\omega}) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \phi(e^{j(\omega - \frac{2\pi}{N}k)})$$

表明：以矩形窗的频谱为内插函数恢复成 $X(e^{j\omega})$ 。

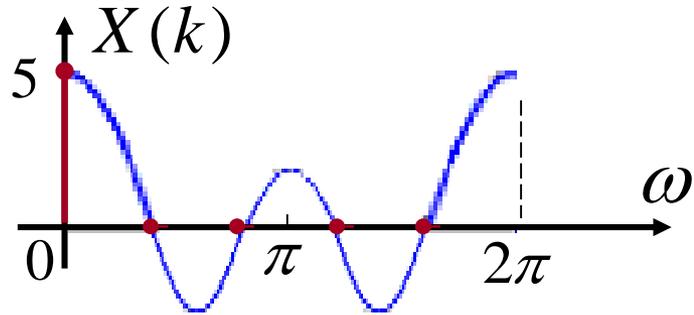
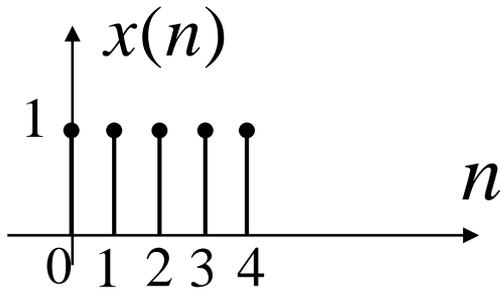


例: $x(n) = R_N(n) \quad 0 \leq n \leq 4, \quad N = 5$

$$X(k) = \sum_{n=0}^4 W_5^{kn} = \sum_{n=0}^4 e^{-j\frac{2\pi}{5}kn} = \begin{cases} 5, & k = 0 \\ 0, & k = 1 \sim 4 \end{cases}$$

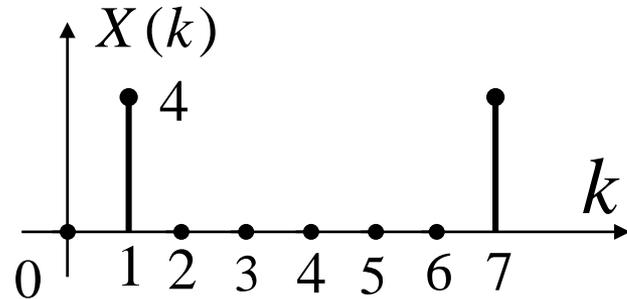
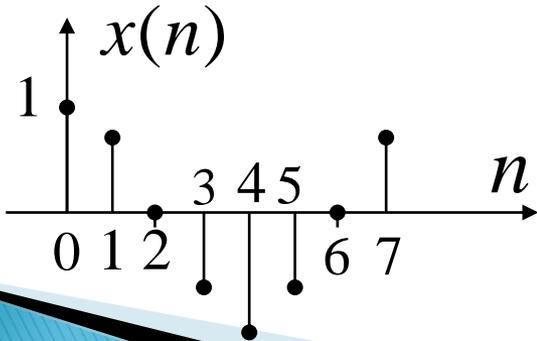
$$X(e^{j\omega}) = \sum_{n=0}^4 e^{-j\omega n} = \frac{\sin(5\omega/2)}{\sin(\omega/2)} e^{-j2\omega}$$

显然有: $X(k) = X(e^{j\omega}) \Big|_{\omega = \frac{2\pi}{5}k} = \begin{cases} 5, & k = 0 \\ 0, & 1 \leq k \leq 4 \end{cases}$



例: $x(n) = \cos\left(\frac{\pi}{4}n\right)$ $0 \leq n \leq 7, \quad N = 8$

$$X(k) = \sum_{n=0}^{7} \cos\left(\frac{\pi}{4}n\right) W_N^{kn} = \begin{cases} 4, & k = 1, 7 \\ 0, & \text{otherwise} \end{cases}$$



7.2 离散傅立叶变换(DFT)的性质:

1. 线性:

如果 $x(n), y(n)$ 均为N点有限长序列, 且:

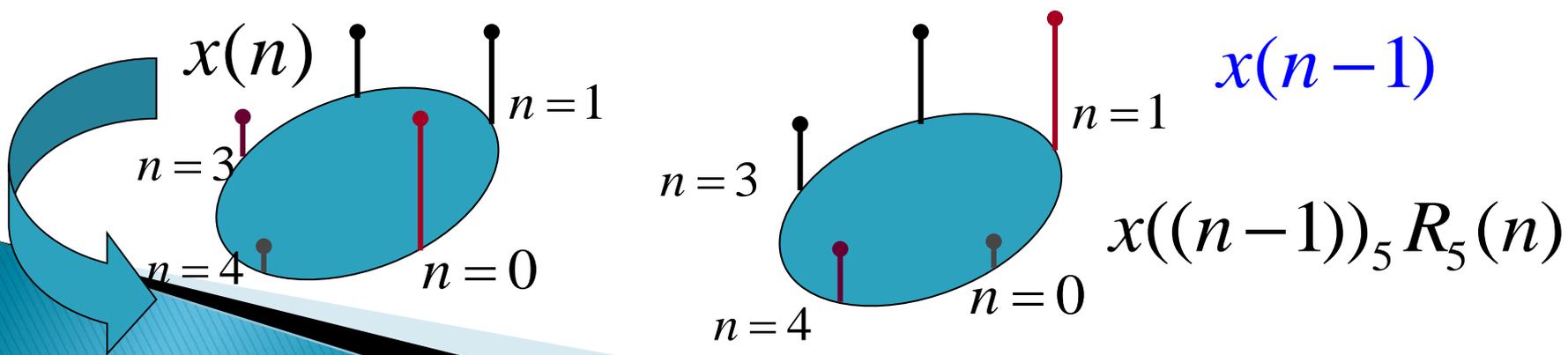
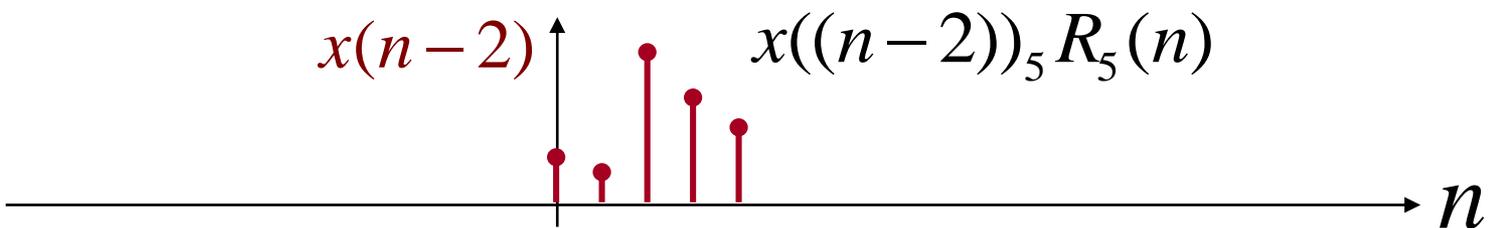
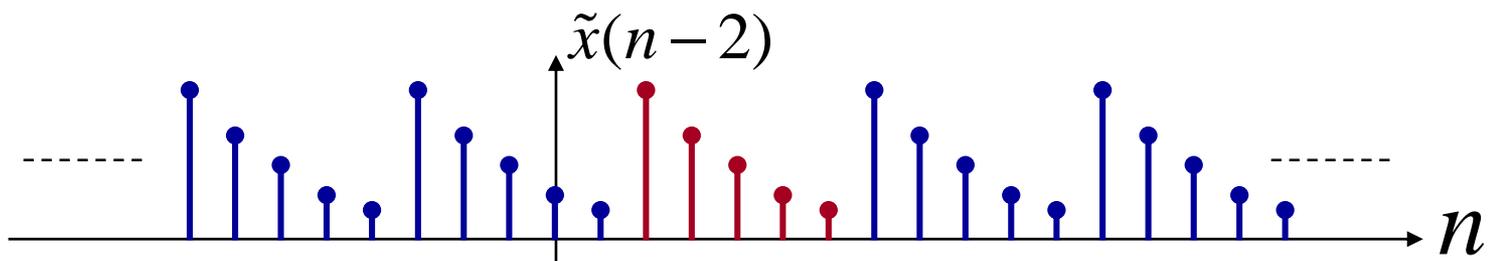
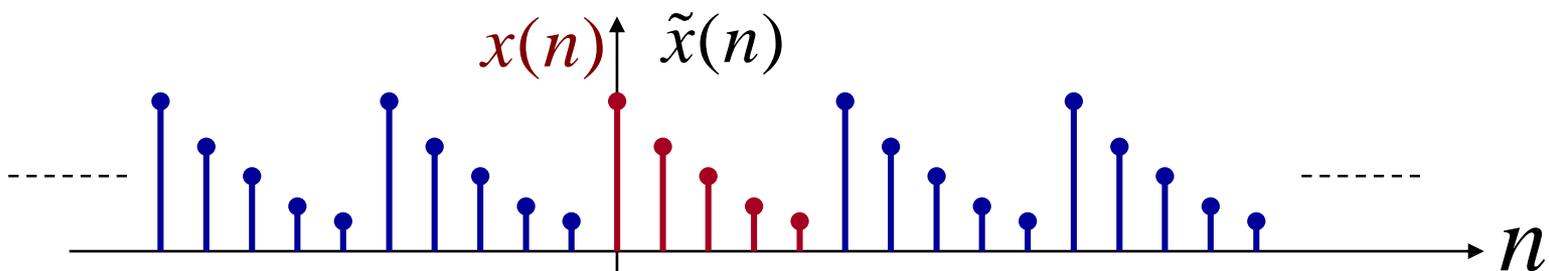
$$x(n) \xleftrightarrow{\text{DFT}} X(k) \quad y(n) \xleftrightarrow{\text{DFT}} Y(k)$$

则: $ax(n) + by(n) \xleftrightarrow{\text{DFT}} aX(k) + bY(k)$

2. 圆周移位:

$x((n - n_0))_N R_N(n)$ 称为 $x(n)$ 的**圆周移位**。

先将 $x(n)$ 以N为周期延拓, 再移位 n_0 , 然后取主值周期。本质上是周期信号移位。这相当于将序列置于圆周上随 n_0 转动, 故称为**圆周移位**。



若：
$$x_1(n) = x((n - n_0))_N R_N(n)$$

$$= \tilde{x}(n - n_0) R_N(n)$$

则：
$$X_1(k) = W_N^{kn_0} X(k)$$

$$\because \tilde{x}(n - n_0) \xleftrightarrow{\text{DFS}} W_N^{kn_0} \tilde{X}(k)$$

$$X_1(k) = \text{DFT}[x_1(n)] = \text{DFT}[\tilde{x}(n - n_0) R_N(n)]$$

$$= W_N^{kn_0} \tilde{X}(k) R_N(k) = W_N^{kn_0} X(k)$$

频域圆周移位：

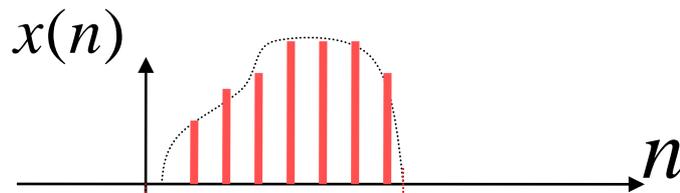
与时域对应，如果有限长序列的DFT在频域圆周

移位，则有：
$$W_N^{-k_0 n} x(n) \leftrightarrow X((k - k_0))_N R_N(k)$$

实现圆周移位的步骤:

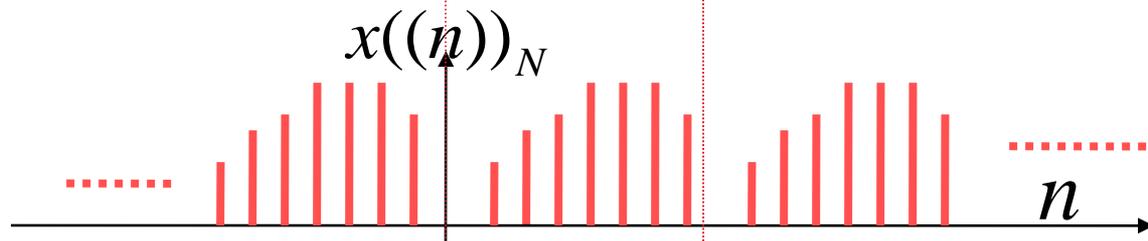
- 有限长序列

$$x(n) \quad 0 \leq n \leq N-1$$



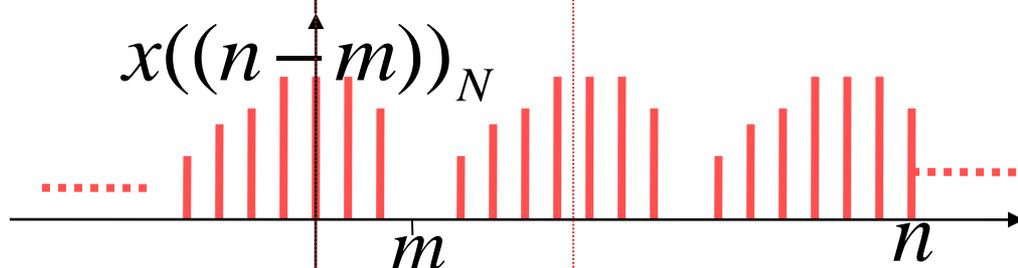
- 周期延拓

$$x((n))_N$$



- 线性位移

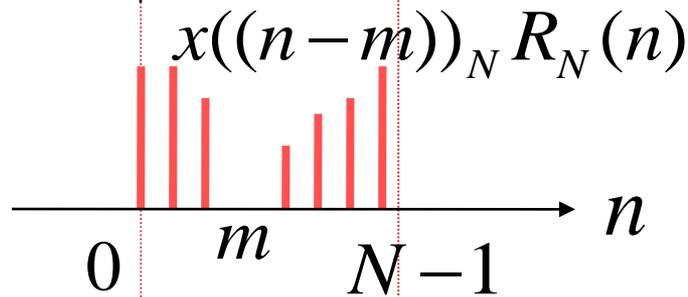
$$x((n-m))_N$$



- 加窗

- 得到圆周位移序列

$$x((n-m))_N R_N(n)$$



3. 周期卷积与圆周卷积:

如果 $\tilde{x}(n)$, $\tilde{y}(n)$ 以 N 为周期, 则将

$$\tilde{f}(n) = \tilde{x}(n) \otimes \tilde{y}(n) = \sum_{m=0}^{N-1} \tilde{x}(m) \tilde{y}(n-m)$$

$$= \sum_{m=0}^{N-1} \tilde{x}(n-m) \tilde{y}(m) \text{ 定义为 } \tilde{x}(n) \text{ 与 } \tilde{y}(n) \text{ 的周期卷积。}$$

显然 $\tilde{f}(n)$ 也是以 N 为周期的。

$$\text{若 } x(n) = \tilde{x}(n)R_N(n) \quad y(n) = \tilde{y}(n)R_N(n)$$

上式可以写成:

$$\tilde{f}(n) = \sum_{m=0}^{N-1} x(m) y((n-m))_N = \sum_{m=0}^{N-1} x((n-m))_N y(m)$$

定义: 两个长度相同序列的圆周卷积:

$$f(n) = x(n) \otimes y(n) = \sum_{m=0}^{N-1} x(m) y((n-m))_N \cdot R_N(n)$$

$$= \sum_{m=0}^{N-1} x((n-m))_N y(m) \cdot R_N(n) \quad \text{称为**圆周卷积**}$$

- **圆周卷积的实质**, 是先将两个有限长序列延拓成周期序列, 做周期卷积, 然后对卷积结果取主值周期, 即:

$$f(n) = \tilde{f}(n) \cdot R_N(n)$$

若 $f(n) = x(n) \otimes y(n)$ 则:

$$F(k) = X(k) \cdot Y(k)$$

$$\because \tilde{f}(n) = \tilde{x}(n) \otimes \tilde{y}(n)$$

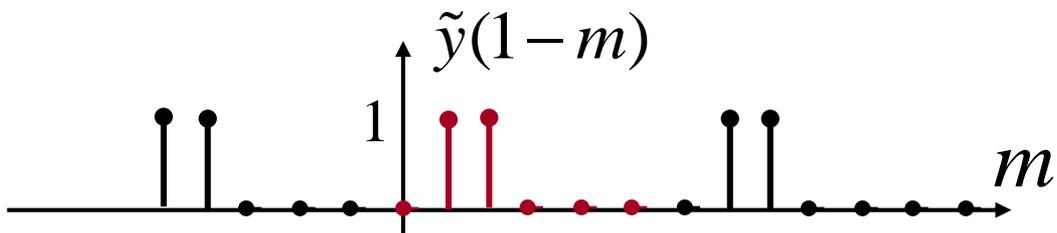
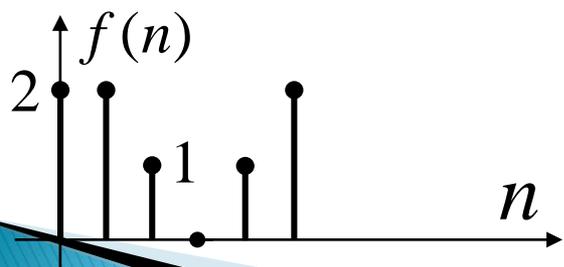
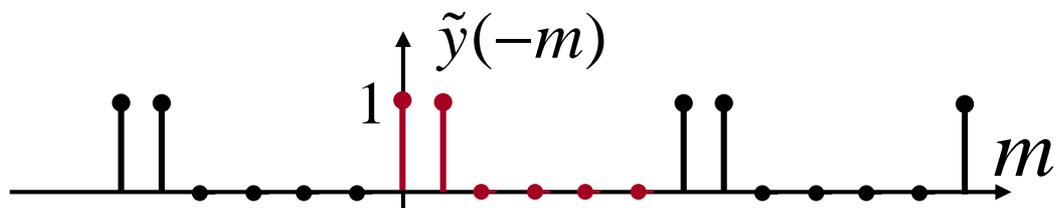
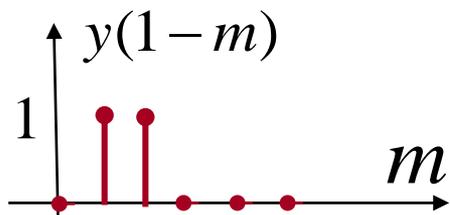
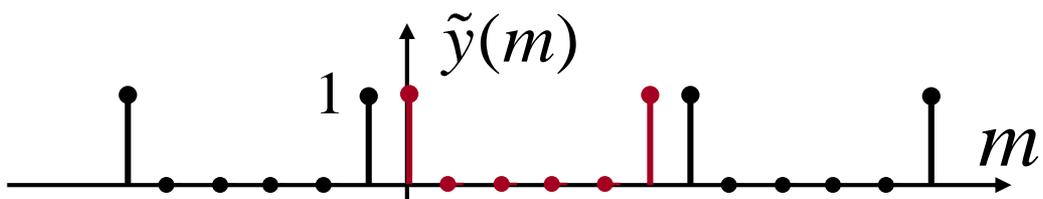
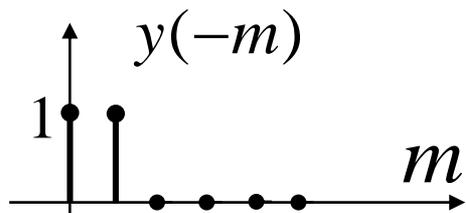
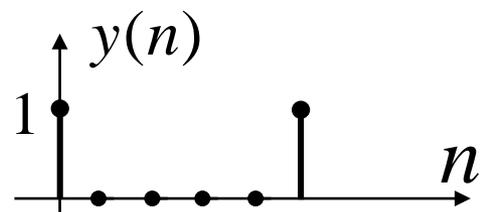
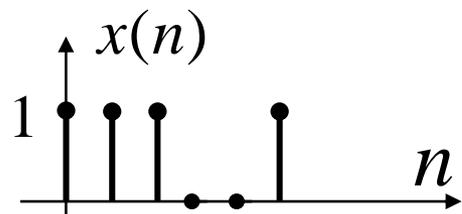
$$\tilde{F}(k) = \tilde{X}(k) \tilde{Y}(k)$$

$$\begin{aligned} \text{DFS: } \quad \dot{C}_k &= N \dot{A}_k \dot{B}_k \\ \tilde{X}(k) &= N \dot{A}_k \quad \tilde{Y}(k) = N \dot{B}_k \\ \tilde{F}(k) &= N \dot{C}_k \end{aligned}$$

$$\begin{aligned} \therefore F(k) &= \tilde{F}(k) R_N(k) = \tilde{X}(k) \tilde{Y}(k) R_N(k) \\ &= \tilde{X}(k) R_N(k) \tilde{Y}(k) R_N(k) = X(k) Y(k) \end{aligned}$$

频域卷积: 若 $f(n) = x(n) y(n)$ 则:

$$F(k) = \frac{1}{N} X(k) \otimes Y(k)$$



4. 共轭对称性：

若 $x(n) \xleftrightarrow{\text{DFT}} X(k)$ 则：

$x^*(n) \xleftrightarrow{\text{DFT}} X^*(N-k)$ 有时也记为 $X^*(-k)$

$$\sum_{n=0}^{N-1} x^*(n) W_N^{kn} = \left[\sum_{n=0}^{N-1} x(n) W_N^{-kn} \right]^* = X^*((-k))_N R_N(k)$$

$$= X^*((N-k))_N R_N(k) = X^*(N-k) \quad k = 0 \sim N-1$$

当 $k=0$ 时， $X^*(N) = X^*(0)$

1. 由 $x(n) = x_r(n) + jx_i(n)$ 可得:

$$x_r(n) = \frac{1}{2}[x(n) + x^*(n)] \quad \text{—— 实部}$$

$$jx_i(n) = \frac{1}{2}[x(n) - x^*(n)] \quad \text{—— 虚部}$$

若 $x_r(n) \xleftrightarrow{\text{DFT}} X_e(k) \quad jx_i(n) \xleftrightarrow{\text{DFT}} X_o(k)$

则 $X_e(k) = \frac{1}{2}[X(k) + X^*(N-k)]$

$$X_o(k) = \frac{1}{2}[X(k) - X^*(N-k)]$$

显然, $X(k) = X_e(k) + X_o(k)$

$$\text{由于 } X_e^*(N-k) = \frac{1}{2}[X(N-k) + X^*(k)]^* = X_e(k)$$

所以称其为**圆周共轭偶对称**，或称 $X_e(k)$ 是 $X(k)$ 的**圆周共轭偶部**。由此，进一步可得：

$$\text{Re}[X_e(k)] = \text{Re}[X_e(N-k)]$$

$$\text{Im}[X_e(k)] = -\text{Im}[X_e(N-k)]$$

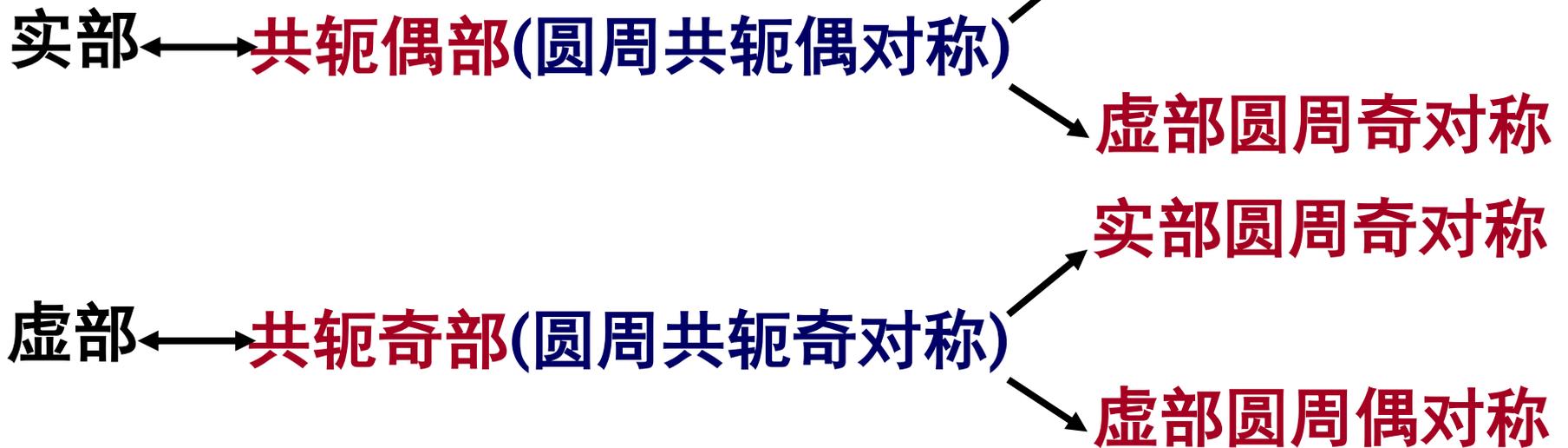
同理可得： $X_o(k) = -X_o^*(N-k)$ **圆周共轭奇对称**，

$X_o(k)$ 是 $X(k)$ 的**圆周共轭奇部**。

进一步可得： $\text{Re}[X_o(k)] = -\text{Re}[X_o(N-k)]$

$$\text{Im}[X_o(k)] = \text{Im}[X_o(N-k)]$$

结论：



2. 若 $x(n)$ 是实序列， 则

$$X(k) = X_e(k) \text{ 只有共轭偶部}$$

若 $x(n)$ 是纯虚序列， 则

$$X(k) = X_o(k) \text{ 只有共轭奇部}$$

在这两种情况下， 只要知道 $X(k)$ 的一半序列值，
即可得出另一半序列值。

3. 由于 $x(n) = x_e(n) + x_o(n)$

$$x(N) = x(0)$$

$$x_e(n) = \frac{1}{2}[x(n) + x^*(N-n)] \quad \text{—— 圆周共轭偶部}$$

$$x_o(n) = \frac{1}{2}[x(n) - x^*(N-n)] \quad \text{—— 圆周共轭奇部}$$

$$x_e(n) \xleftrightarrow{\text{DFT}} \frac{1}{2}[X(k) + X^*(k)] = \text{Re}[X(k)]$$

$$x_o(n) \xleftrightarrow{\text{DFT}} \frac{1}{2}[X(k) - X^*(k)] = j \text{Im}[X(k)]$$

$$x^*(n) \leftrightarrow X^*(N-k) = X^*(-k)$$

$$x^*(-n) = x^*(N-n) \leftrightarrow X^*(k)$$

6. Parseval定理:

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

$$\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2 = \sum_{k=0}^{N-1} |\dot{A}_k|^2$$

在这里 $X(k) = N\dot{A}_k$, $k = 0 \sim N-1$

其实就是DFS的Parseval定理的变形。

一般情况下有:

$$\sum_{n=0}^{N-1} x(n)y^*(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)Y^*(k)$$

当 $x(n) = y(n)$ 时, 就成为前面的形式。

7.4 利用DFT计算线性卷积

若 $x(n)$ 是 N 点序列, $y(n)$ 是 M 点序列,

$$\text{线性卷积: } f(n) = x(n) * y(n) = \sum_{m=-\infty}^{\infty} x(m)y(n-m)$$

$x(m)$ 的非零区间: $0 \leq m \leq N-1$

$y(n-m)$ 的非零区间: $0 \leq n-m \leq M-1$

因此, $f(n)$ 的非零区间为: $0 \leq n \leq N+M-2$

即: **线性卷积** 的长度为 $N+M-1$ 点。

将 $x(n)$ 、 $y(n)$ 均补零加长到 L 点, 做圆周卷积有:

$$\tilde{f}(n) = \tilde{x}(n) \otimes \tilde{y}(n) \cdot R_L(n) = \sum_{m=0}^{L-1} x(m) \tilde{y}(n-m) \cdot R_L(n)$$

$$\tilde{y}(n-m) = y((n-m))_L = \sum_{k=-\infty}^{\infty} y(n-m-kL)$$

$$\begin{aligned} \text{代入: } \tilde{f}(n) &= \sum_{m=0}^{L-1} x(m) \sum_{k=-\infty}^{\infty} y(n-m-kL) \cdot R_L(n) \\ &= \sum_{k=-\infty}^{\infty} \sum_{m=0}^{L-1} x(m) y(n-m-kL) \cdot R_L(n) \\ &= \sum_{k=-\infty}^{\infty} f(n-kL) \cdot R_L(n) \end{aligned}$$

表明：圆周卷积是线性卷积周期延拓后取的主值周期。

• 显然，当 $L \geq N + M - 1$ 时，

$f(n)$ 周期延拓时不发生重叠，可以通过取 $\tilde{f}(n)$ 的主值周期，即通过圆周卷积得到 $f(n)$ 。这就是**通过计算圆周卷积求得线性卷积必须满足的条件**。

• 当 $L < N + M - 1$ 时，由于周期性延拓时必然发生重叠，无法通过圆周卷积取得线性卷积的结果。

7.4 DFT应用中的几个具体问题

一. 频谱的混叠现象：(spectrum aliasing)

实际应用中的多数信号并不存在数学解析式，信号的频谱无法用定义直接计算，通常是采用数值方法进行近似的计算。在时域将连续时间信号通过采样离散化时，但在离散化处理时要遵循采样定理，否则会出现频谱混叠，此时样本序列不能代表原信号，对其作DFT也就失去意义了。但采样频率过高则会增大数据量影响运算速度，且要求较大内存量。

对非带限信号，频谱混叠是必然的。只能通过提高采样频率减轻频谱混叠程度，借以减小误差。

若不知道信号的最高频率，只有已经记录下来的波形或数据，则可以从波形或数据中找出变化规律最快的相邻两点，以这两点的时间间隔 t_d 为依据，按照

$$f_m = \frac{1}{2t_d} \text{ 近似确定最高频率 } f_m \text{。}$$

二. 信号截断与频谱泄漏:

带限信号一定非时限。在时域将连续时间信号离散化时，为满足Nyquist 定理要求，信号必须带限，那么时域序列必为无限长。要对它做DFT就必须将其截断为有限长序列，为此要乘上矩形窗口函数

$$R_N(n) \leftrightarrow W_N(e^{j\omega})$$

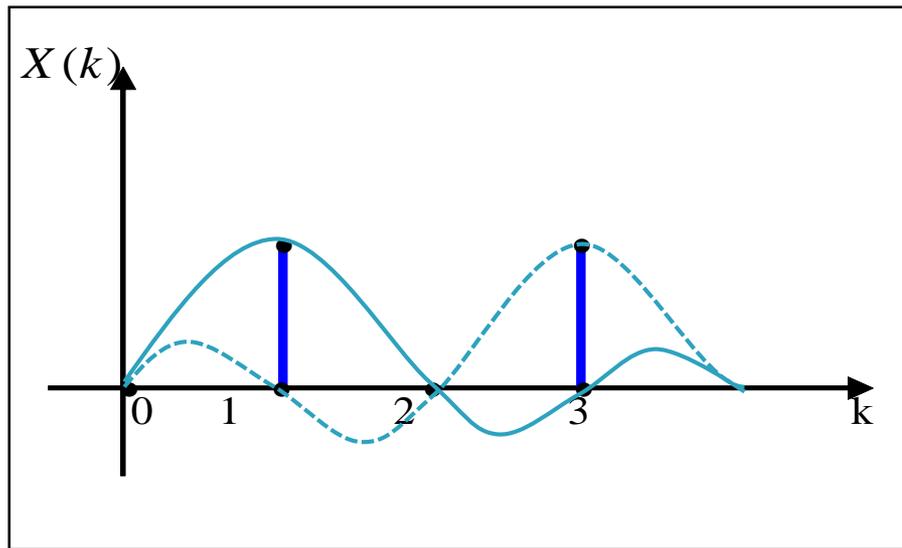
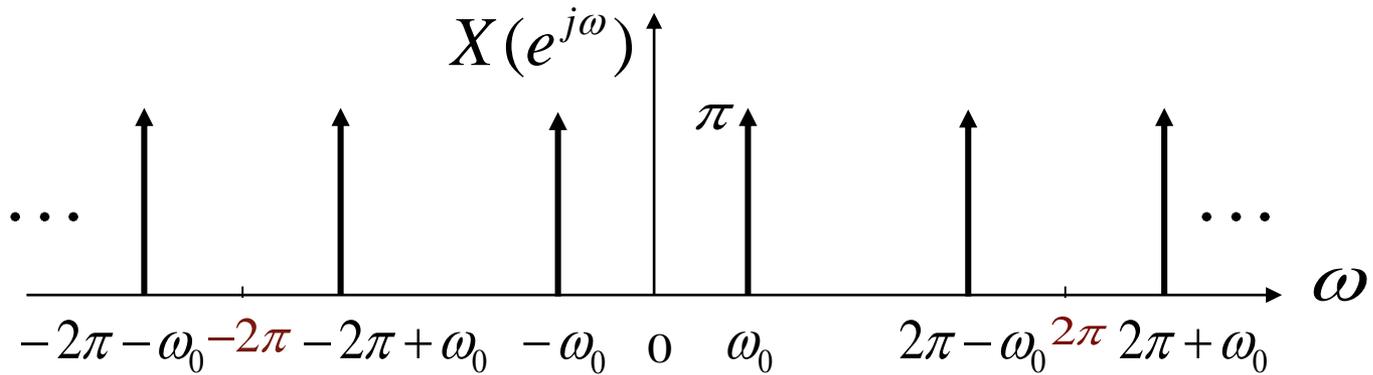
截断后的有限长序列的频谱：

$$\hat{X}(e^{j\omega}) = \frac{1}{2\pi} X(e^{j\omega}) \otimes W_N(e^{j\omega})$$

由于 $W_N(e^{j\omega})$ 的引入可能会产生**频谱泄漏**，导致在DFT运算中产生出信号中本来没有的频率分量。

例： $x(n) = \cos \frac{\pi}{2} n$ $N_0 = 4$

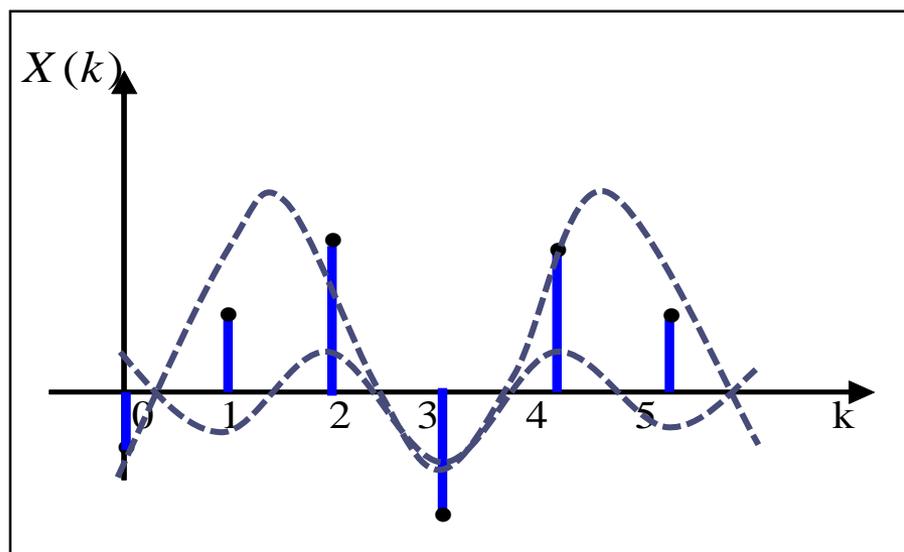
分别以 $N = 4$ 和 $N = 6$ 将其截断，并分别对截断的有限长序列做4点和6点DFT，考察其结果。



N=4



Dft.exe



$N=6$

结论： 对周期信号截断时，必须保证 N 为信号周期的整数倍；对非周期信号，先截取 N 点做 N 点DFT，再取 $2N$ 点做 $2N$ 点DFT.....依次类推，直到两次DFT的结果非常接近时为止。以此确定截断长度。

三. 频率分辨率:

若模拟信号 $x_a(t)$ 带限于 Ω_M ，采样频率 $\Omega_s = 2\pi f_s$ ，在 t_p 一段时间内采样 N 点，则采样间隔:

$$T = 2\pi / \Omega_s = 1 / f_s \quad t_p = NT = N / f_s$$

由于 Ω_s 总对应于离散域的频率 2π ，因此，做 N 点 DFT 后，在数字域的频率分辨率为： $\Delta\omega = 2\pi / N$

连续域频率分辨率为： $\Delta f = f_s / N = 1 / t_p$ (Hz)

这表明：在 t_p 一定时，模拟域频率分辨率与采样点数无关。不能通过增加采样的点数或提高采样频率来改善模拟域的频率分辨率。**要提高模拟域的频率分辨率，只有增大采样信号的时间区段 t_p 才能奏效。**

在工程应用中，通常是根据对模拟域频率分辨率 Δf 的要求，来决定采样信号的时间区段 t_p ，再根据采样定理选定 Ω_s ，进而决定采样点数 N ，这就是做DFT的点数。

四. 栅栏效应:

DFT是信号频谱的等间隔样本，相当于通过栅栏观察信号的频谱。因此，必定有一些地方会被挡住（即采样时采不到的那些点），而在**DFT**的结果中无法体现。这种现象就称为**栅栏效应**。

在工程应用中，可以通过在序列的尾部补零，加长原序列的长度，从而增加做**DFT**的点数，来消除栅栏效应。这相当于调整了原来栅栏的间隙，使频谱中那些在原来采样时采不到的分量得以反映。

7.5 快速傅里叶变换 (FFT) :

一. DFT的运算特点:

运算工作量:
$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad 0 \leq k \leq N-1$$

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

求出 $X(k)$ 的每一个点, 要做 N 次复数乘法, $(N-1)$ 次复数加法。求出全部 $X(k)$ 要做 N^2 次复数乘法, 和做 $N(N-1)$ 次复数加法。可见, DFT 运算具有 N^2 数量级的运算量。当 N 较大时, 运算量是巨大的 (如: $N=1024$ 点时, 运算工作量超过 10^6 数量级)。

正是由于DFT具有巨大的运算量，因此在快速算法产生之前，极大地阻碍了离散时间信号处理技术在工程实际中的应用。

DFT的运算特点：

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

1. W_N^{kn} 对 k, n 均以 N 为周期。

$$W_N^{(k+N)n} = W_N^{(n+N)k} = W_N^{kn}$$

2. W_N^{kn} 有对称性。 $W_N^{k(N-n)} = W_N^{(N-k)n} = W_N^{-kn}$

$$W_N^{\pm N/2} = -1 \quad W_N^{(k \pm N/2)} = -W_N^k$$

$$W_{N/2}^{kn} = W_N^{2kn}$$

最重要的是：如能将N点序列分成几个短序列，
则可以有效地减少运算次数。 N^2 量级运算量

例如将一个N点序列分成两个N/2点序列，就可以减少近一半的运算工作量。

FFT的基本思想：将长序列分成短序列。并利用DFT的运算特点（周期性、对称性）减少运算量。

二. 按时间抽取的FFT算法（Cooley-Tukey算法）

若 $x(n)$ 的长度 $N = 2^M$

把 $x(n)$ 按 n 为奇数、偶数分成两组： $x_1(r) = x(2r)$;

$$x_2(r) = x(2r + 1), \quad (r = 0 \sim \frac{N}{2} - 1)$$

$$\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{kn} \\
&= \sum_{r=0}^{(N/2)-1} x(2r) W_N^{2rk} + \sum_{r=0}^{(N/2)-1} x(2r+1) W_N^{(2r+1)k} \\
&= \sum_{r=0}^{(N/2)-1} x(2r) W_{\frac{N}{2}}^{rk} + \sum_{r=0}^{(N/2)-1} x(2r+1) W_{\frac{N}{2}}^{rk} \cdot W_N^k \\
&= \sum_{n=0}^{(N/2)-1} x_1(n) W_{\frac{N}{2}}^{nk} + \sum_{n=0}^{(N/2)-1} x_2(n) W_{\frac{N}{2}}^{nk} \cdot W_N^k \\
&= X_1(k) + W_N^k X_2(k) \quad 0 \leq k \leq \frac{N}{2} - 1
\end{aligned}$$

这就得到了 $X(k)$ 的前半部分。

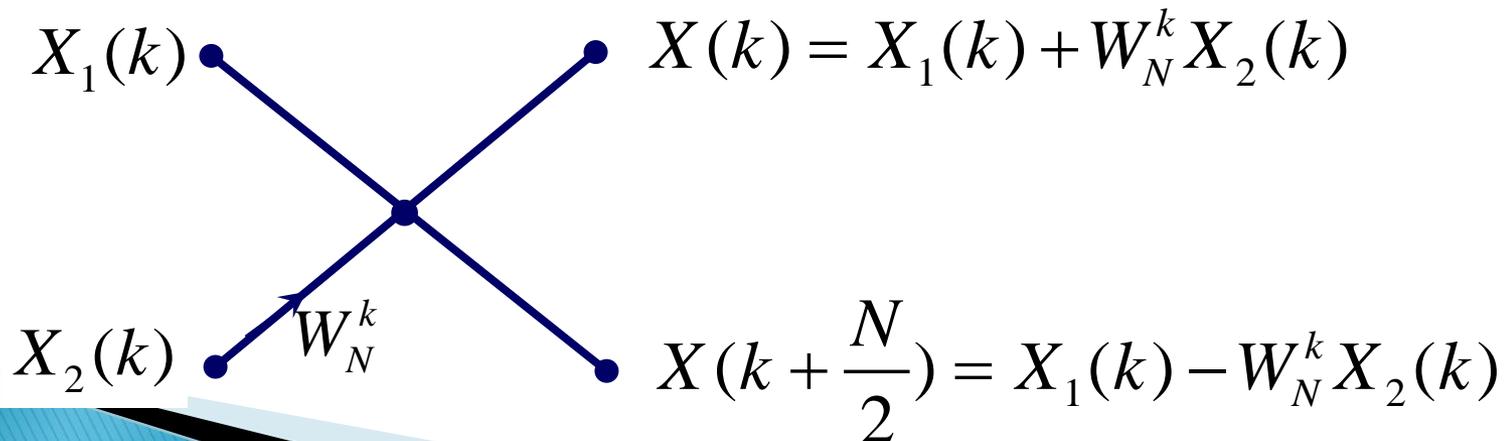
$$X\left(k + \frac{N}{2}\right) = \sum_{n=0}^{(N/2)-1} x_1(n) W_{\frac{N}{2}}^{n(k+\frac{N}{2})} + \sum_{n=0}^{(N/2)-1} x_2(n) W_{\frac{N}{2}}^{n(k+\frac{N}{2})} \cdot W_N^{(k+\frac{N}{2})}$$

$$= X_1(k) - W_N^k X_2(k) \quad \text{所以有:}$$

$$X(k) = X_1(k) + W_N^k X_2(k)$$

$$0 \leq k \leq \frac{N}{2} - 1$$

$$X\left(k + \frac{N}{2}\right) = X_1(k) - W_N^k X_2(k)$$



对于 $N=2^M$ 的序列，经 $M-1$ 次分解，最后分解为每组两点的DFT。对一个两点序列做DFT不需要做乘法，此时：

$$X(k) = \sum_{n=0}^1 x(n)W_2^{kn} \quad \begin{aligned} X(0) &= x(0) + x(1) \\ X(1) &= x(0) - x(1) \end{aligned}$$

由于这种算法在分组时是按时域中序列的奇、偶位分解的，故称**按时间抽取的FFT算法**，也被称为DIT(Decimation in Time)算法。

例：一个8点序列DIT算法的运算流程：



Dft.exe

三. 按频率抽取的FFT算法: (Sand-Tukey 算法)

$N = 2^M$ 将序列每次按前一半后一半分组。

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{kn} = \sum_{n=0}^{(N/2)-1} x(n) W_N^{kn} + \sum_{n=(N/2)}^{N-1} x(n) W_N^{kn} \\ &= \sum_{n=0}^{(N/2)-1} x(n) W_N^{kn} + \sum_{n=0}^{(N/2)-1} x(n + \frac{N}{2}) W_N^{k(n + \frac{N}{2})} \\ &= \sum_{n=0}^{(N/2)-1} [x(n) + (-1)^k x(n + \frac{N}{2})] W_N^{kn} \quad 0 \leq k \leq N-1 \end{aligned}$$

k为偶数 ($k = 2r$) 时:

$$X(2r) = \sum_{n=0}^{(N/2)-1} [x(n) + x(n + \frac{N}{2})] W_N^{2rn} \quad \text{即}$$

$$X(2r) = \sum_{n=0}^{(N/2)-1} [x(n) + x(n + \frac{N}{2})] W_{\frac{N}{2}}^{rn} \quad 0 \leq r \leq \frac{N}{2} - 1$$

k为奇数 ($k = 2r + 1$)时 :

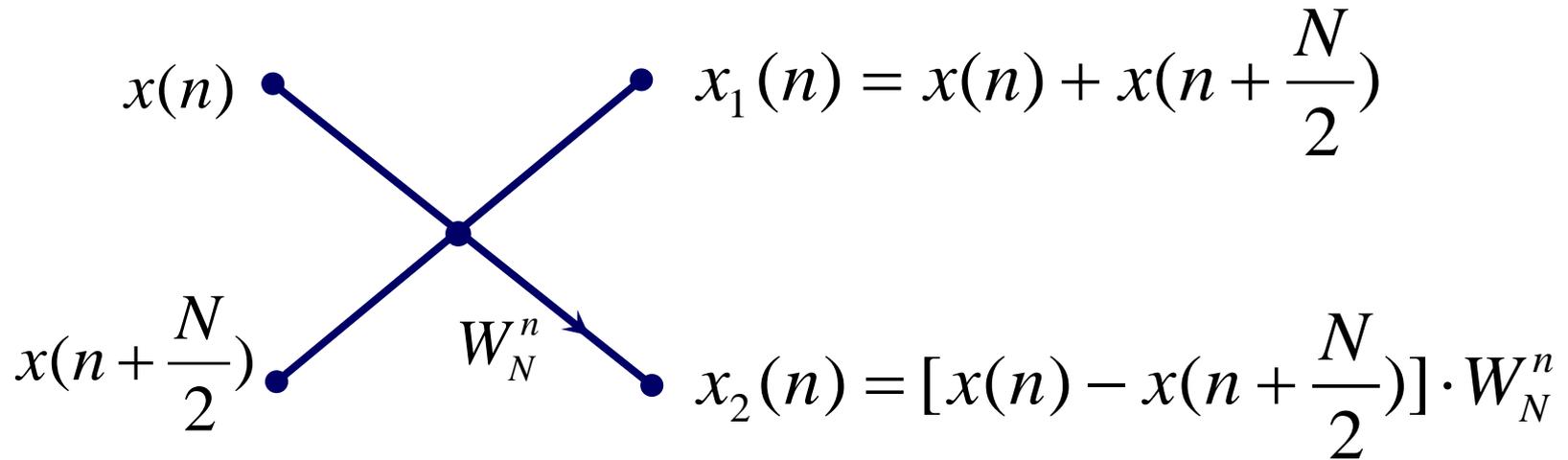
$$\begin{aligned} X(2r+1) &= \sum_{n=0}^{(N/2)-1} [x(n) - x(n + \frac{N}{2})] W_N^{(2r+1)n} \\ &= \sum_{n=0}^{(N/2)-1} [x(n) - x(n + \frac{N}{2})] W_N^n \cdot W_{\frac{N}{2}}^{rn} \end{aligned}$$

令 $x_1(n) = x(n) + x(n + \frac{N}{2})$

$$x_2(n) = [x(n) - x(n + \frac{N}{2})] \cdot W_N^n$$

则 $X(2r) \leftrightarrow x_1(n)$ $X(2r+1) \leftrightarrow x_2(n)$

此时，信号的分组与组合可以用以下蝶形结表示：



因为这种算法的结果，在频域表现为是按频域的奇、偶位分组的，故称**按频率抽取的FFT算法**。也称为**DIF(Decimation in Frequency)算法**。



- DIT与DIF算法的运算工作量完全相同。

复数乘：
$$M \cdot \frac{N}{2} = \frac{N}{2} \log_2 N$$

复数加：
$$N \cdot M = N \log_2 N$$

运算量为： $N \log_2 N$ 数量级。

当 $N = 2^{10} = 1024$ 时运算量大约 10^4 。比直接运算快大约100倍。—— 可见快速算法的重要性。

- DIT与DIF算法都可以进行原位运算，且运算流程十分规则。

- DIT与DIF算法都需要**码位倒置**过程。

一个8点序列的码位倒置过程：

$x(0)$	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	$x(6)$	$x(7)$
000	001	010	011	100	101	110	111
000	100	010	110	001	101	011	111
$x(0)$	$x(4)$	$x(2)$	$x(6)$	$x(1)$	$x(5)$	$x(3)$	$x(7)$

DIT算法与DIF算法的区别：

- 基本蝶形结的结构不同。
- DIT法中DFT运算在第一级完成
- DIF法中DFT运算在最后一级完成。

DIT与DIF算法的根本区别并不在于谁的输入或输出是码位倒置的，而是它们蝶形结的结构不同。

四. IDFT的快速算法：（IFFT）

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

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

IDFT与DFT的区别：

- 1) W_N^{kn} 指数有负号
- 2) 有系数 $1/N$

可以完全照DFT的方法建立相应的按频率抽取，
或按时间抽取的运算流程

由于：

$$\begin{aligned}x(n) &= \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn} = \frac{1}{N} \left[\sum_{k=0}^{N-1} X^*(k) W_N^{kn} \right]^* \\ &= \frac{1}{N} \{ \text{DFT}[X^*(k)] \}^*\end{aligned}$$

先将 $X(k)$ 取复共轭，再对 $X(k)$ 的共轭执行FFT程序，对运算结果再取共轭，并乘以 $1/N$ 即可。完全可以利用FFT程序实现IFFT运算。

工程应用中常用的实现IFFT运算的方法。

7.6 信号频谱分析的Matlab实现

fft(x) fft(x,N) ifft(x) ifft(x,N)

```
%Program 7-1 of DFT using matlab
clc;clear;
t1=0:4;t2=0:7;t3=0:11;t4=0:15;
x1=[1 1 1 1 1];
x2=cos(pi/4*t2);
x3=cos(pi/4*t3);
x4=cos(pi/4*t4);
x1k=fft(x1,5);
x2k=fft(x2,8);
x3k=fft(x3,12);
x4k=fft(x4,16);

subplot(221);stem(t1,abs(x1k));
axis([0 4 0 6]);xlabel('a');title(' 5 Point DFT of 例7.1')
subplot(222);stem(t2,abs(x2k));
axis([0 8 0 6]);xlabel('a');title(' 8 Point DFT of 例7.2')
subplot(223);stem(t3,abs(x3k));
axis([0 12 0 6]);xlabel('a');
subplot(224);stem(t4,abs(x4k));
axis([0 16 0 10]);xlabel('a');
```