



北京理工大学  
Beijing Institute of Technology

## 本科生实验报告

课程名称：高级数字信号处理

实验名称：DFT 实验原理及验证

任课教师：	田黎育			实验教师：	田黎育
实验日期：	2025 年 9 月 10 日			实验地点：	
实验类型：	<input checked="" type="checkbox"/> 原理验证 <input type="checkbox"/> 综合设计 <input type="checkbox"/> 自主创新				
学生姓名：	闫子易	班级：	13212302	学号：	1120230621
学 院：	集成电路与电子学院			专 业：	电子科学与技术
组 号：		同 组 同 学：			
成 绩：					



集成电路与电子学院  
SCHOOL OF INTEGRATED CIRCUITS  
AND ELECTRONICS

## 实验一、二、三 实数复数信号的离散傅里叶变换

### 一、实验目的

#### 1. 实数信号

$N=8$

$\text{Fre}=1$

手算第 1, 2, 3 点, 这里 1、2、3 点指  $X[0], X[1], X[2]$ , 下面的题以此类推, 第一个点是  $X[0]$

画复数平面图

#### 2. 实数信号

$N=8$

$\text{Fre}=2$ , 加初相  $45^\circ$

手算第 2, 3, 4 点

画复数平面图

#### 3. 复数信号 DFT

$N=8$

$\text{Fre}=3$ ,

手算第 2, 3, 4 点

实数与复数信号离散傅里叶变换 (DFT) 实验的目的为: 理解 DFT 的基本定义与物理意义, 明确信号点数  $N=8$ 、频率  $\text{Fre}$  (1、2、3)、初相 ( $45^\circ$ ) 及信号类型 (无初相实数、含初相实数、复数) 对 DFT 结果的影响逻辑; 掌握不同类型信号 DFT 的手工计算方法, 能准确求解特定频率点 (如无初相实数信号的  $X[0]$ 、 $X[1]$ 、 $X[2]$ , 含  $45^\circ$  初相实数信号的  $X[1]$ 、 $X[2]$ 、 $X[3]$ , 复数信号的  $X[1]$ 、 $X[2]$ 、 $X[3]$ ) 的实部与虚部; 学会将 DFT 复数结果转化为复数平面图, 通过图形直观理解频谱分布及初相对

相位偏移的作用；区分实数信号（含无初相、有初相）与复数信号的 DFT 差异，验证实数信号 DFT 的共轭对称性，认识复数信号 DFT 无共轭对称的特征及独立表征正负频率分量的优势；建立时域信号参数（频率、初相、信号类型）与频域 DFT 结果的对应关系，对比不同信号的频谱特征，为后续复杂信号的 DFT 分析奠定理论与实践基础。

## 二、实验原理

**傅里叶变换**（Fourier transform）是一种线性的积分变换，就是一种从时间到频率的变化或其相互转化。

连续周期信号（Periodic Continuous）：

$$\begin{cases} x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} \\ c_k = \frac{1}{T_0} \int_{T_0} x(t) e^{-jk\omega_0 t} dt \end{cases}$$

连续非周期信号（Non-periodic Continuous）：

$$\begin{cases} x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega t} d\omega \\ X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt \end{cases}$$

离散周期信号（Periodic Discrete）：

$$\begin{cases} x[n] = \sum_{k=\langle N \rangle} c_k e^{jk\frac{2\pi}{N}n} \\ c_k = \frac{1}{N} \sum_{n=\langle N \rangle} x[n] e^{-jk\frac{2\pi}{N}n} \end{cases}$$

离散非周期信号(Non- periodic Discrete)：

$$\begin{cases} x[n] = \frac{1}{2\pi} \int_{2\pi} X(\Omega) e^{j\Omega n} d\Omega \\ X(\Omega) = \sum_{k=-\infty}^{+\infty} x[n] e^{-j\Omega n} \end{cases}$$

其中，我们研究的数字信号，周期离散信号的傅里叶变换

Discrete Fourier Series



Discrete Fourier Transform



$$\begin{cases} X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \cdot \frac{2\pi}{N} \cdot k \cdot n} \\ x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j \cdot \frac{2\pi}{N} \cdot k \cdot n} \end{cases}$$



$$\begin{cases} X(kf_1) = DFT[x(nT_s)] = \sum_{n=0}^{N-1} x(nT_s) e^{-j(\frac{2\pi}{N})nk} \\ x(nT_s) = IDFT[X(kf_1)] = \frac{1}{N} \sum_{nk=0}^{N-1} X(kf_1) e^{j(\frac{2\pi}{N})nk} \end{cases}$$

$$f_1 = \frac{1}{T_1}, T_s = \frac{T_1}{N}, W = e^{-j(\frac{2\pi}{N})}$$

其中，T1 为信号时间，N 为信号采样数，Ts 为信号采样间隔时间，f1 为频率间隔。这样就能很清晰的看到，变换后的坐标轴都是 f1 的整数倍。由时域采样间隔 Ts，确定频域重复周期 fs，而 df→f1=fs/N，即确定了频域的横轴坐标。

这个变换可以写成矩阵形式，如下所示：

DFT:

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & \dots & W^0 \\ W^0 & W^1 & \dots & W^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ W^0 & W^{N-1} & \dots & W^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}$$

IDFT:

$$\begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} = \frac{1}{N} \begin{bmatrix} W^0 & W^0 & \dots & W^0 \\ W^0 & W^{-1} & \dots & W^{-(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ W^0 & W^{-(N-1)} & \dots & W^{-(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix}$$

其中,  $W = e^{-j(\frac{2\pi}{N})}$

这是一个  $N \times N$  方阵, 矩阵的基本元素由  $W = e^{-j(\frac{2\pi}{N})}$  来构成, 每个位置元素  $a_{ij} = W^{(i-1)(j-1)} = e^{-j \cdot 2\pi \frac{(i-1)(j-1)}{N}}$ 。即矩阵元素的索引是从自然数 1 开始的, 但指数运算需从 0 开始。

这个矩阵是主对角对称矩阵。即

$$W_T = W$$

$$W_T W = NI$$

DFT 矩阵有以下性质:

1. 线性变换
2. 满秩矩阵 (每一行一列不能由其他行、列线性表出) 信息不丢失
3. 特殊结构 范德蒙矩阵 DFT  $\rightarrow$  FFT
4. 正交性

相位校正  $\rightarrow$  SNR 上升

信号连续  $\rightarrow$  离散  $t \rightarrow n\Delta T \quad 1/\Delta t = f$

DFT (离散傅里叶变换) 矩阵有多个关键性质。首先, 它具有**线性变换特性**, 能够遵循线性叠加原理

DFT（离散傅里叶变换）矩阵是连接离散时域与频域的核心工具，核心性质支撑其在信号处理中的关键应用。它具备**线性变换特性**，可依线性叠加原理将时域信号映射至频域，为信号分析奠基；同时是**满秩矩阵**，任意行、列无法由其他行、列线性表出，能确保信号转换无信息丢失，保障逆变换可行性。

其**范德蒙矩阵特殊结构**让 FFT（快速傅里叶变换）得以通过分治策略优化计算，将 DFT 原本  $O(N^2)$  的复杂度降至  $O(N\log N)$ ，实现大规模信号高效处理。此外，DFT 矩阵的**正交性**不仅简化逆变换求解（逆矩阵可通过转置或共轭转置获取），还能助力信号相位校正，修正传输或处理中的相位偏差，有效提升信噪比 SNR。

实际应用中，连续时间信号（t）需经采样变为离散序列（ $n\Delta T$ ， $\Delta T$  为采样间隔）。

$$w_0 = 2\pi f_0$$

$$A\cos(2\pi f_0 t + \varphi_0)$$

↓

$$n\Delta t \rightarrow \frac{n}{f_s}$$

$$x[n] = A_0 \cos(2\pi f_0 \cdot \frac{n}{f_s})$$

Exp:

$$f_0 = 125\text{Hz} \rightarrow \text{周期 } T = 8\text{ms}$$

$$f_s = 1\text{kHz} \rightarrow \text{周期 } T_s = 1\text{ms}$$

持续时间 8ms，采样一个周期 8 个点

### 三、实验内容与主要步骤

实数信号， $N=8$ ， $F_{re}=1$ ，手算第 1，2，3 点，这里 1、2、3 点指  $X[0], X[1], X[2]$ ，下面的题以此类推，第一个点是  $X[0]$ ，画复数平面图。

首先，信号实数时，可表示为  $x[n] = \cos\left(\frac{2\pi}{n} \cdot Fre \cdot n\right)$

代入此题已知条件

$$\begin{aligned}x[n] &= \cos\left(\frac{2\pi}{N} \cdot Fre \cdot n\right) \\&= \cos\left(\frac{2\pi}{N} \cdot 1 \cdot n\right) = \cos\frac{\pi}{4}n \\&= \frac{1}{2}(e^{j\frac{\pi}{4}n} + e^{-j\frac{\pi}{4}n}) \\&= [1, \frac{\sqrt{2}}{2}, 0, -\frac{\sqrt{2}}{2}, -1, -\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}]\end{aligned}$$

根据离散傅里叶变换公式：

$$\begin{aligned}X[k] &= \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N} \cdot k \cdot n} \\X[k] &= \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad , W_N = e^{-j\frac{2\pi}{N}}\end{aligned}$$

代入此题已知条件

$$X[k] = \sum_{n=0}^7 x[n] e^{-j\frac{\pi}{4} \cdot k \cdot n}$$

★ 如果把  $e$  也写成向量形式，也就是

$$e[n] = [e^0, e^{-j\frac{\pi}{4} \cdot k \cdot 1}, e^{-j\frac{\pi}{4} \cdot k \cdot 2}, \dots, e^{-j\frac{\pi}{4} \cdot k \cdot n}]$$

那么这个求和又可看作  $x[n]$  向量与  $e$  向量的点乘，即

$$X(k) = [x[n]] \cdot [e[n]]$$

手算  $X[0,1,2]$

$$\text{实数时 } x[n] = \cos\left(\frac{2\pi}{N} \cdot Fre \cdot n\right)$$

**1.  $N = 8$  ,  $Fre = 1$  , 求  $X[0, 1, 2]$**

$$\begin{aligned}x[n] &= \cos\left(\frac{2\pi}{N} \cdot Fre \cdot n\right) \\&= \cos\left(\frac{2\pi}{8} \cdot 1 \cdot n\right) = \cos\frac{\pi}{4}n\end{aligned}$$

$$= \frac{1}{2}(e^{j\frac{\pi}{4}n} + e^{-j\frac{\pi}{4}n})$$

$$= [1, \frac{\sqrt{2}}{2}, 0, -\frac{\sqrt{2}}{2}, -1, -\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}]$$

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

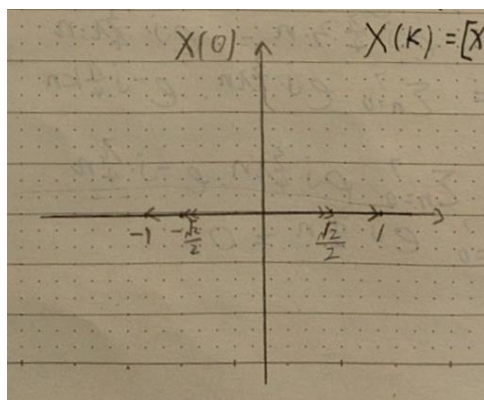
$$= \sum_{n=0}^7 X[n] e^{-j\frac{\pi}{4}kn}$$

这个求和又可看作  $x[n]$  向量与  $e$  向量的点乘，即

$$X(k) = [x[n]] \cdot [e[n]]$$

$$X(0) = \sum_{n=0}^7 x[n] \cdot 1$$

$$= 0$$



图像中标记的-1、 $-\frac{\sqrt{2}}{2}$ 、 $\frac{\sqrt{2}}{2}$ 、1 对应离散余弦序列  $x[n] = \cos\frac{\pi}{4}n$  的时域幅值，它们体现了该序列在不同离散点的大小分布，而这样的时域幅值分布使得序列无直流分量（如  $X(0)=0$ ），同时也决定了其频域会在与余弦频率对应的点（如  $k=1$ 、 $k=7$ ）呈现能量峰值，这些幅值是频域频谱特性与时域序列关联的直观体现。

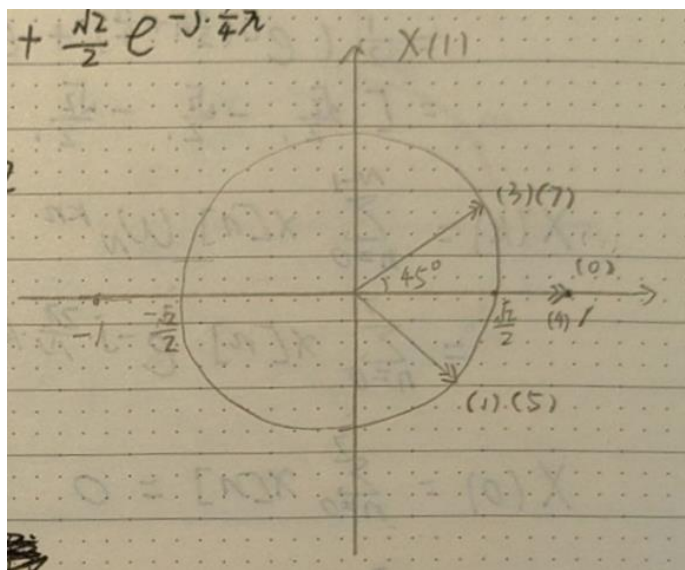
$$X(1) = \sum_{n=0}^7 x[n] e^{-j\frac{\pi}{4}n}$$

$$x[n] = [1, \frac{\sqrt{2}}{2}, 0, -\frac{\sqrt{2}}{2}, -1, -\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}]$$



$$e[n] = [1, e^{-j\frac{\pi}{4}}, e^{-j\frac{\pi}{2}}, e^{-j\frac{3\pi}{4}}, e^{-j\pi}, e^{-j\frac{5\pi}{4}}, e^{-j\frac{3\pi}{2}}, e^{-j\frac{7\pi}{4}}]$$

$$\begin{aligned} X(1) &= 1 + \frac{\sqrt{2}}{2}e^{-j\frac{\pi}{4}} + 0 - \frac{\sqrt{2}}{2}e^{-j\frac{3\pi}{4}} - e^{-j\pi} - \frac{\sqrt{2}}{2}e^{-j\frac{5\pi}{4}} + 0 + \frac{\sqrt{2}}{2}e^{-j\frac{7\pi}{4}} \\ &= \frac{\sqrt{2}}{2} \times \cos\frac{\pi}{4} \cdot 2 \times 2 + 1 \times 2 \\ &= 2 + 2 \\ &= 4 \end{aligned}$$



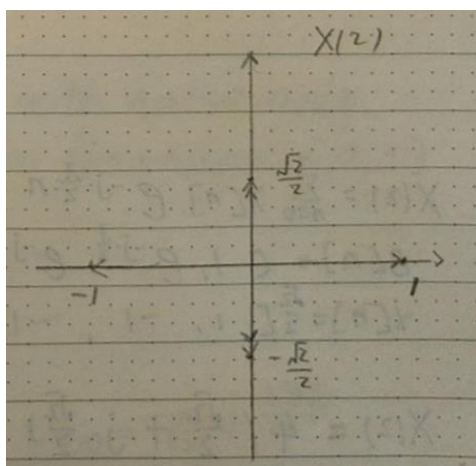
复指数 $e^{-j\frac{\pi}{4}n}$ 的相位变化对于 8 点序列( $n = 0, 1 \dots, 7$ )，复指数 $e^{-j\frac{\pi}{4}n}$ 的相位随  $n$  变化: $n = 0$ 时 $e^0 = 1$ (实轴正方向，对应图中 $(0)$ 附近的实轴点， $n = 4$ 时 $e^{-j\frac{\pi}{4} \times 4} = e^{-j\pi} = -1$ ，实轴负方向)。  $n = 1$ 时 $e^{-j\frac{\pi}{4}} = \cos\frac{\pi}{4} - j\sin\frac{\pi}{4} = \frac{\sqrt{2}}{2} - j\frac{\sqrt{2}}{2}$ (第四象限， $45^\circ$ 角，对应图中 $(1)$ 附近的点； $n = 5$ 时 $e^{-j\frac{5\pi}{4}} = -\frac{\sqrt{2}}{2} + j\frac{\sqrt{2}}{2}$ ，第二象限，与 $n = 1$ 关于原点对称)。  $n = 3$ 时 $e^{-j\frac{3\pi}{4}} = -\frac{\sqrt{2}}{2} - j\frac{\sqrt{2}}{2}$ (第三象限，对应图中 $(3)$ 附近的点； $n = 7$ 时 $e^{-j\frac{7\pi}{4}} = \frac{\sqrt{2}}{2} + j\frac{\sqrt{2}}{2}$ ，第一象限，与 $n = 3$ 关于原点对称)。3.与 $X(1)$ 计算的关联  $X(1) = \sum_{n=0}^7 x[n]e^{-j\frac{\pi}{4}n}$ 是 $x[n]$ 与复指数 $e^{-j\frac{\pi}{4}n}$ 的“点乘”，图中单位圆上的点直观展示了 $e^{-j\frac{\pi}{4}n}$ 的相位分布，相互叠加之后只剩下了实数轴上的 4。

$$X(2) = \sum_{n=0}^7 x[n] e^{-j\frac{\pi}{2}n}$$

$$e[n] = [1, e^{-j\frac{\pi}{2}}, e^{-j\pi}, e^{-j\frac{3\pi}{2}}, e^{-j2\pi}, e^{-j\frac{5\pi}{2}}, e^{-j3\pi}, e^{-j\frac{7\pi}{2}}]$$

$$x[n] = [1, \frac{\sqrt{2}}{2}, 0, -\frac{\sqrt{2}}{2}, -1, -\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}]$$

$$X(2) = 0$$



当计算 $X(2) = \sum_{n=0}^7 x[n] e^{-j\frac{\pi}{2}n}$ 时，基函数是 $e^{-j\frac{\pi}{2}n}$ （因 DFT 基函数为 $e^{-j\frac{2\pi}{N}kn}$ ，这里 $N = 8, k = 2$ ，故指数为 $-j\frac{\pi}{2}n$ ）。基函数 $e^{-j\frac{\pi}{2}n}$ 随 $n$ 的取值在复平面上沿单位圆旋转（每次旋转 $90^\circ$ ！

$n = 0 \ e^0 = 1$ (实轴正方向，对应图中实轴“1”);

$n = 1 \ e^{-j\frac{\pi}{2}} = -j$ (虚轴负方向);

$n = 2 \ e^{-j\pi} = -1$ (实轴负方向，对应图中实轴“-1”);

$n = 3 \ e^{-j\frac{3\pi}{2}} = j$ (虚轴正方向);

$n = 4 \ e^{-j2\pi} = 1$ (周期为 4，回到实轴正方向)。

序列 $x[n] = [1, \frac{\sqrt{2}}{2}, 0, -\frac{\sqrt{2}}{2}, -1, -\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}]$ 与基函数 $e^{-j\frac{\pi}{2}n}$ 逐点相乘后求和，结果 $X(2) = 0$ 。从图中看，虚轴标记的 $\pm \frac{\sqrt{2}}{2}$ 对应 $x[n]$ 中的分量，实轴 $\pm 1$ 对应基函数的实值点。最终 $X(2) = 0$ 说明：基函数与 $x[n]$ 相乘后，实部和虚部的正、负贡献完全抵消，在复平面上表现为“原点”（值为 0，无偏离原点的分量）。

2.  $N = 8$  ,  $Fre = 2$ , 初相位  $45^\circ$ , 求  $X[1, 2, 3]$

$$\begin{aligned}
 x[n] &= \cos\left(\frac{2\pi}{N} \cdot Fre \cdot n + \varphi_0\right) \\
 &= \cos\left(\frac{2\pi}{8} \cdot 2 \cdot n + \frac{\pi}{4}\right) = \cos\left(\frac{\pi}{2}n + \frac{\pi}{4}\right) \\
 &= \frac{1}{2}(e^{j(\frac{\pi}{2}n + \frac{\pi}{4})} + e^{-j(\frac{\pi}{2}n + \frac{\pi}{4})}) \\
 &= \left[\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right]
 \end{aligned}$$

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{N-1} x[n] W_N^{kn}, W_N = e^{-j\frac{2\pi}{N}} \\
 &= \sum_{n=0}^7 x[n] e^{-j\frac{2\pi}{N}kn}
 \end{aligned}$$

这个求和又可看作  $x[n]$  向量与  $e$  向量的点乘, 即

$$X(k) = [x[n]] \cdot [e[n]]$$

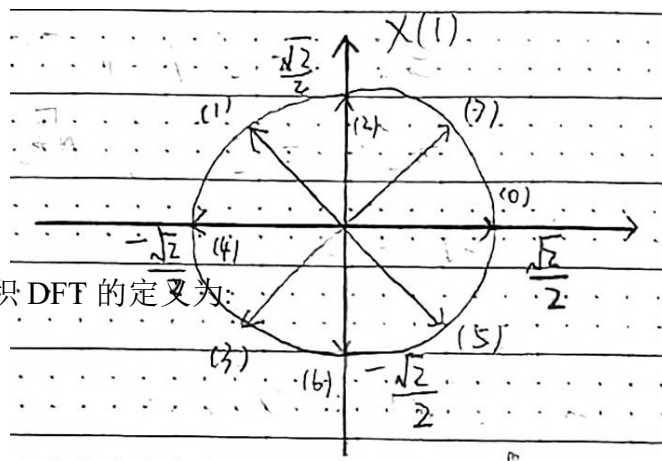
$$\begin{aligned}
 X(0) &= \sum_{n=0}^7 x[n] \\
 &= 0
 \end{aligned}$$

$$X(1) = \sum_{n=0}^7 x[n] e^{-j\frac{\pi}{4}n}$$

$$e[n] = [1, e^{-j\frac{\pi}{4}}, e^{-j\frac{\pi}{2}}, e^{-j\frac{3\pi}{4}}, e^{-j\pi}, e^{-j\frac{5\pi}{4}}, e^{-j\frac{3\pi}{2}}, e^{-j\frac{7\pi}{4}}]$$

$$X(1) = 0$$

复向量的内积 DFT 的定义为:



$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{kn}, \quad W_N = e^{-j\frac{2\pi}{N}}$$

其中  $W_N^{kn} = e^{-j\frac{2\pi}{N}kn}$  是旋转因子,几何意义是复平面上角度为  $-\frac{2\pi}{N}kn$  的单位向量。

因此,  $X[k]$  可理解为:序列  $x[n]$  与”旋转因子向量”  $\{W_N^{kn}\}_{n=0}^{N-1}$  的复内积(点乘)。单频信号的频谱稀疏性已知信号为单频余弦信号:

$$x[n] = \cos\left(\frac{\pi}{2}n + \frac{\pi}{4}\right)$$

利用欧拉公式展开为两个复指数的和:

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

复指数  $e^{j\omega n}$  的频率由  $\omega$  决定。此处  $\omega = \frac{\pi}{2} = \frac{2\pi}{8} \cdot 2$ , 对应 DFT 的频率索引  $k = \text{Fre} = 2$  (因为  $\omega = \frac{2\pi}{N}k$ )。根据 DFT 的正交性:只有当旋转因子的频率  $k^*$  与信号复指数的频率匹配时,内积(求和)才会非零;否则,因正交性(向量夹角为  $90^\circ$ ),内积为 0。3.具体到  $X[1]$ : 频率不匹配导致正交对于  $X[1]$ , 旋转因子的频率是  $k = 1$ , 对应的旋转因子为  $W_8^{1 \cdot n} = e^{-j\frac{\pi}{4}n}$  (角度为  $-\frac{\pi}{4}n$  的单位向量)。而信号  $x[n]$  的复指数频率是  $\pm 2$  (对应  $k = 2$  和  $k = 8 - 2 = 6$ ), 与  $k = 1$  不匹配。因此, “信号向量  $x[n]$  与旋转因子向量  $\{e^{-j\frac{\pi}{4}n}\}^n$  “在复平面上正交(内积为 0)。从图形上看:代表  $x[n]$  的复向量与代表  $W_8^{1n}$  的复向量,各分量的相位相互垂直,乘积求和后完全抵消,最终结果为 0。总结单频信号的 DFT 具有频谱稀疏性:仅在与信号频率匹配的索引  $k = \text{Fre}$  和  $k = N - \text{Fre}$  处有非零分量,其余  $k$  因旋转因子与信号复指数正交,导致 DFT 系数为 0。图形中正交向量的相互抵消,直观体现了这一数学本质。

$$X(2) = \sum_{n=0}^7 x[n] e^{-j\frac{\pi}{2}n}$$

$$e[n] = [1, e^{-j\frac{\pi}{2}}, e^{-j\pi}, e^{-j\frac{3\pi}{2}}, e^{-j2\pi}, e^{-j\frac{5\pi}{2}}, e^{-j3\pi}, e^{-j\frac{7\pi}{2}}]$$

$$x[n] = \frac{\sqrt{2}}{2} [1, -1, -1, 1, 1, -1, -1, 1]$$

$$X[2] = 4 \left( \frac{\sqrt{2}}{2} + j \cdot \frac{\sqrt{2}}{2} \right)$$



3. 逐项计算并求和将  $x[n]$  与  $e^{-j\frac{\pi}{2}n}$  逐项相乘,再合并实部和虚部

$n$	$x[n]$	$e^{-j\frac{\pi}{2}n}$	$x[n] \cdot e^{-j\frac{\pi}{2}n}$	实部贡献	虚部贡献
0	$\frac{\sqrt{2}}{2}$	1	$\frac{\sqrt{2}}{2} \cdot 1 = \frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	0
1	$-\frac{\sqrt{2}}{2}$	$-j$	$-\frac{\sqrt{2}}{2} \cdot (-j) = \frac{\sqrt{2}}{2}j$	0	$\frac{\sqrt{2}}{2}j$
2	$-\frac{\sqrt{2}}{2}$	-1	$-\frac{\sqrt{2}}{2} \cdot (-1) = \frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	0
3	$\frac{\sqrt{2}}{2}$	$j$	$\frac{\sqrt{2}}{2} \cdot j = \frac{\sqrt{2}}{2}j$	0	$\frac{\sqrt{2}}{2}j$
4	$\frac{\sqrt{2}}{2}$	1	$\frac{\sqrt{2}}{2} \cdot 1 = \frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	0
5	$-\frac{\sqrt{2}}{2}$	$-j$	$-\frac{\sqrt{2}}{2} \cdot (-j) = \frac{\sqrt{2}}{2}j$	0	$\frac{\sqrt{2}}{2}j$
6	$-\frac{\sqrt{2}}{2}$	-1	$-\frac{\sqrt{2}}{2} \cdot (-1) = \frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	0
7	$\frac{\sqrt{2}}{2}$	$j$	$\frac{\sqrt{2}}{2} \cdot j = \frac{\sqrt{2}}{2}j$	0	$\frac{\sqrt{2}}{2}j$

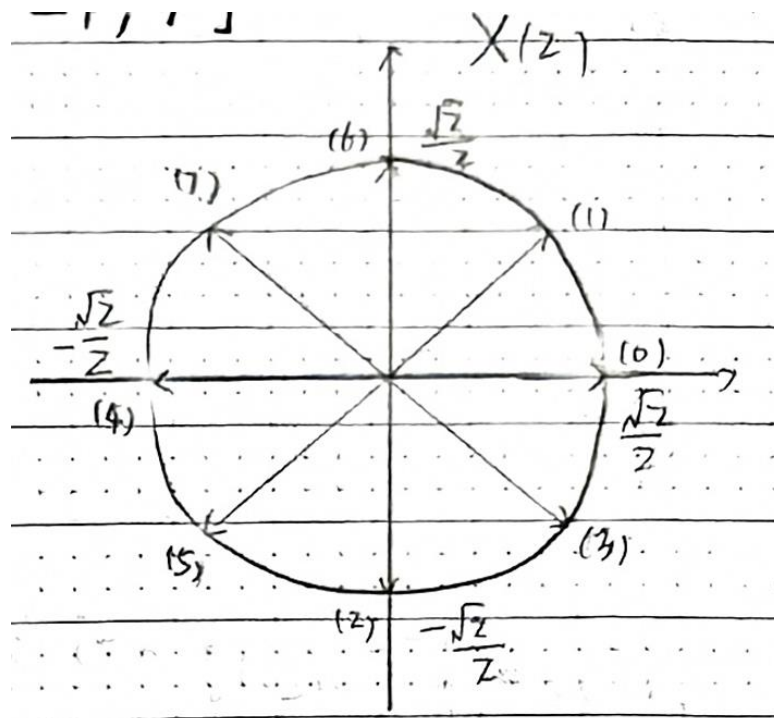
合并实部与虚部实部总和: $n = 0, 2, 4, 6$ 这 4 项的实部均为 $\frac{\sqrt{2}}{2}$ , 因此:实部和= $4 \times \frac{\sqrt{2}}{2} = 2\sqrt{2}$ 虚部总和: $n = 1, 3, 5, 7$ 这 4 项的虚部均为 $\frac{\sqrt{2}}{2}j$ , 因此:虚部和= $4 \times \frac{\sqrt{2}}{2}j = 2\sqrt{2}j$ 最终,  $X[2] = \text{实部和} + \text{虚部和} = 2\sqrt{2} + 2\sqrt{2}j$ , 与题目结果一致。从图的角度,可观察到实部和虚部分别存在 4 次重复的正贡献,因此通过重复项数 $\times$ 单项贡献即可快速得到总和。

$$X(3) = \sum_{n=0}^7 x[n] e^{-j\frac{3}{4}\pi n}$$

$$e[n] = \left[ 1, e^{-j\frac{3}{4}\pi}, e^{-j\frac{3}{2}\pi}, e^{-j\frac{9}{4}\pi}, e^{-j3\pi}, e^{-j\frac{15}{4}\pi}, e^{-j\frac{9}{2}\pi}, e^{-j\frac{21}{4}\pi} \right]$$

$$x[n] = \frac{\sqrt{2}}{2} [1, -1, -1, 1, 1, -1, -1, 1]$$

$$X(3) = 0$$



对于长度为  $N = 8$  的序列  $x[n]$ ，其  $k = 3$  点 DFT 为：

$$X[3] = \sum_{n=0}^7 x[n] \cdot e^{-j\frac{2\pi \cdot 3}{8}n} = \sum_{n=0}^7 x[n] \cdot e^{-j\frac{3\pi}{4}n}$$

• 实部的抵消

$$n = 0: \frac{\sqrt{2}}{2} \times 1 = \frac{\sqrt{2}}{2}$$

$$n = 4: \frac{\sqrt{2}}{2} \times (-1) = -\frac{\sqrt{2}}{2}$$

→  $n = 0$  与  $n = 4$  的实部正负抵消 (和为 0)。

$$n = 1: -\frac{\sqrt{2}}{2} \times \left(-\frac{\sqrt{2}}{2}\right) = \frac{1}{2}$$

$$n = 3: \frac{\sqrt{2}}{2} \times \frac{\sqrt{2}}{2} = \frac{1}{2}$$

$$n = 5: -\frac{\sqrt{2}}{2} \times \frac{\sqrt{2}}{2} = -\frac{1}{2}$$

$$n = 7: \frac{\sqrt{2}}{2} \times \left(-\frac{\sqrt{2}}{2}\right) = -\frac{1}{2}$$

→  $n = 1, 3$  的正实部与  $n = 5, 7$  的负实部两两抵消 (和为 0)。

• 虚部的抵消

$$n = 2: -\frac{\sqrt{2}}{2} \times j = -j\frac{\sqrt{2}}{2}$$

$$n = 6: -\frac{\sqrt{2}}{2} \times (-j) = j\frac{\sqrt{2}}{2}$$

→  $n = 2$  与  $n = 6$  的虚部**正负抵消** (和为 0)。

$$n = 1: -\frac{\sqrt{2}}{2} \times \left(-j\frac{\sqrt{2}}{2}\right) = j\frac{1}{2}$$

$$n = 3: \frac{\sqrt{2}}{2} \times \left(-j\frac{\sqrt{2}}{2}\right) = -j\frac{1}{2}$$

$$n = 5: -\frac{\sqrt{2}}{2} \times j\frac{\sqrt{2}}{2} = -j\frac{1}{2}$$

$$n = 7: \frac{\sqrt{2}}{2} \times j\frac{\sqrt{2}}{2} = j\frac{1}{2}$$

→  $n=1,7$  的正虚部与  $n=3,5$  的负虚部**两两抵消** (和为 0)。

由于**实部总和**与**虚部总和**均因“对称抵消”为 0，因此： $X[3]=$ 实部和+虚部和  
 $=0+0=0$

从图的角度(复平面上的点分布)， $x[n]$ 与 $e^{-j\frac{3\pi}{4}n}$ 的乘积在复平面上关于**原点对称**，  
 正贡献与负贡献完全抵消，最终总和为 0。

### 3. $N = 8$ , $\text{Fre} = 3$ , 求 $X[1, 2, 3]$

$$X[k] = \sum_{n=0}^{N-1} X[n] W_N^{kn} \quad W_N = e^{-j\frac{2\pi}{N}}$$

复数信号表示为

$$x[n] = e^{j\frac{2\pi}{N}\text{Fre}\cdot n} \cdot e^{j\cdot\varphi_0}$$

$$x[n] = e^{j\frac{2\pi}{N} \cdot \text{Fre} \cdot n} \cdot e^{j\cdot 0}$$

$$= e^{j\frac{\pi}{4} \cdot 3 \cdot n}$$

$$X[k] = \sum_{n=0}^7 e^{j\frac{3\pi}{4}n} \cdot e^{-j\frac{\pi}{4}kn}$$

$$X[1] = \sum_{n=0}^7 e^{j\frac{3\pi}{4}n} \cdot e^{-j\frac{\pi}{4}n} = \sum_{n=0}^7 e^{j\frac{2\pi}{4}n} = 0$$



$$\begin{aligned} X[2] &= \sum_{n=0}^7 e^{j\frac{3\pi}{4}n} e^{-j\frac{\pi}{2}n} \\ &= \sum_{n=0}^7 e^{-j\frac{\pi}{4}n} \\ &= 0 \end{aligned}$$

$$\begin{aligned} X[2] &= \sum_{n=0}^7 e^{j\frac{3\pi}{4}n} e^{-j\frac{3\pi}{4}n} \\ &= 8 \end{aligned}$$

$$\therefore X[1] = 0, X[2] = 0, X[3] = 8$$

## 实验四 matlab 实现 DFT

### 一、 实验目的

- (1) 用 m 语言写循环实现 DFT
- (2) 用矩阵实现 DFT，以及 IDFT，验证两个线性变换的矩阵的乘积为单位矩阵；
- (3) 用程序验证 1-3 手算的结果

本次实验旨在通过 MATLAB 软件开展与离散傅里叶变换（DFT）相关的实践操作，具体包括用 M 语言编写循环语句实现 DFT，采用矩阵形式实现 DFT 与离散傅里叶逆变换（IDFT）并验证这两个线性变换对应的矩阵乘积为单位矩阵，同时利用编写的程序对 1-3 项内容的手算结果进行验证，以此全面掌握 DFT 的不同实现方法及相关性质验证过程。

### 二、 实验代码

```
clear; clc;
```

```

close all;

% ===== 定义是实数信号还是复数信号 =====
COMPLEX_SIGNAL = 1;
REAL_SIGNAL = 0;

% ===== 参数设置 =====
N = 8; % DFT 点数
Fre = 1.2; % 信号频率
phai = pi/4; % 初相位
highlight = [2,3,4]; % 要标红的点索引 (MATLAB 下标从 1 开始)

% ===== 构造信号 =====
Signal_Type = COMPLEX_SIGNAL; % REAL_SIGNAL -> cos, COMPLEX_SIGNAL -> exp
n = 0:N-1;
if Signal_Type == COMPLEX_SIGNAL
    x = exp(1j*2*pi*Fre*n/N + phai); % 复数信号
else
    x = cos(2*pi*Fre*n/N + phai); % 实信号
end

% ===== DFT 循环实现 =====
X_loop = zeros(1,N);
for k = 0:N-1
    for nn = 0:N-1
        X_loop(k+1) = X_loop(k+1) + x(nn+1) * exp(-1j*2*pi/N*k*nn);
    end
end

% ===== 矩阵实现 =====
W = exp(-1j*2*pi/N*(0:N-1)'*(0:N-1)); % DFT 矩阵
X_matrix = (W * x.').'; % DFT 结果

% ===== IDFT 验证 =====
W_inv = exp(1j*2*pi*(0:N-1)'*(0:N-1)/N)/N;
I_check = W * W_inv; % 应该接近单位阵
I_check = round(I_check, 2); % 保留两位小数
disp('验证 W * W_inv 是否为单位矩阵 (保留两位小数):');
disp(I_check);

% ===== 打印 DFT 循环和矩阵结果 =====
disp('对比 DFT 循环 vs 矩阵结果:');

```

```

for k = 1:N
    if ismember( k , highlight )
        fprintf( '--X[%d] 循环= %.2f%+.2fj , 矩阵= %.2f%+.2fj\n', ...
            k-1, real(X_loop(k)), imag(X_loop(k)), real(X_matrix(k)),
            imag(X_matrix(k)));
    else
        fprintf( 'X[%d] 循环= %.2f%+.2fj , 矩阵= %.2f%+.2fj\n', ...
            k-1, real(X_loop(k)), imag(X_loop(k)), real(X_matrix(k)),
            imag(X_matrix(k)));
    end
end

% ===== 修正：两张图合并为同一窗口（左右布局） =====
figure('Position', [100, 100, 1000, 500]); % 总窗口大小（x,y,宽,高）

% ----- 左图：极坐标复平面图 -----
% 直接创建极坐标 axes，不使用 subplot
pax = polaraxes;
pax.Position = [0.05, 0.1, 0.4, 0.8]; % [左, 下, 宽, 高] 占比
hold(pax, 'on');

% 圆半径取最大幅值的 1.1 倍
R = max(abs([X_loop, X_matrix])) * 1.1;

% 绘制所有 DFT 点
offset_angle = pi/30; % 文字交替偏移，防止重叠
for k = 1:N
    r = abs(X_loop(k)); % 模值（极径）
    theta = angle(X_loop(k)); % 相位（极角）

    % 区分普通点和重点点
    if ismember(k, highlight)
        polarscatter(pax, theta, r, 80, 'r', 'filled'); % 红色大圆点
    else
        polarscatter(pax, theta, r, 60, 'b', 'filled'); % 蓝色小圆点
    end

    % 点的标注
    if ismember(k, highlight) && ~(abs(real(X_loop(k))) < 1e-10 &&
        abs(imag(X_loop(k))) < 1e-10)

```

```

        txt = sprintf('X[%d]: %.2f%+.2fj', k-1, real(X_loop(k)),
            imag(X_loop(k)));
    else
        txt = sprintf('X[%d]', k-1);
    end

    % 添加标注文字
    text(theta + offset_angle, r, txt, 'FontSize', 9, ...
        'HorizontalAlignment', 'left', 'Parent', pax);
    offset_angle = -offset_angle; % 交替正负偏移
end

title(pax, 'DFT 复平面（极坐标）', 'FontSize', 12);

% ----- 右图: DFT 模值离散图 -----
% 创建普通坐标 axes
ax = axes;
ax.Position = [0.55, 0.1, 0.4, 0.8]; % [左, 下, 宽, 高] 占比
hold(ax, 'on');

X_mag = abs(X_loop); % 计算 DFT 模值

% 绘制离散信号图
stem(ax, 0:N-1, X_mag, 'filled', 'MarkerSize', 6, 'Color', [0.2, 0.6, 0.8]);
grid on;
xlabel(ax, 'k (频率点索引)', 'FontSize', 11);
ylabel(ax, '|X[k]| (DFT 模值)', 'FontSize', 11);
title(ax, 'DFT 模值随 k 的变化', 'FontSize', 12);
xlim(ax, [-0.5, N-0.5]);
ylim(ax, [0, max(X_mag)*1.2]);

% ----- 总标题 -----
Sgtitl 'e(sprintf('DFT 结果可视化 (N=%d, 频率 Fre=%d, 初相 phi=%.2f rad)', N, Fre,
    phai), ...
    'FontSize', 14, 'FontWeight', 'bold');

```

代码可分为 6 个核心部分：环境清理、参数定义、信号构造、DFT 计算（两种方法）、IDFT 验证、结果输出与可视化。

## 1. 环境初始化

```
clear; clc; close all;
```

- **clear**: 清除工作区所有变量，避免历史变量干扰。
- **clc**: 清空命令窗口，保持输出整洁。
- **close all**: 关闭所有已打开的图形窗口，确保绘图环境干净。

## 2. 信号类型与参数定义

% 定义信号类型（实数/复数）

```
COMPLEX_SIGNAL = 1;
```

```
REAL_SIGNAL = 0;
```

% 参数设置

```
N = 8; % DFT 点数（决定频域分辨率）
```

```
Fre = 1.2; % 信号频率（归一化频率，单位：周期/采样点）
```

```
phai = pi/4; % 信号初相位（弧度）
```

```
highlight = [2,3,4]; % 需标红的 DFT 结果索引（MATLAB 下标从 1 开始）
```

- 信号类型定义：用常量区分后续构造的是复数信号（**COMPLEX\_SIGNAL**）还是实数信号（**REAL\_SIGNAL**）。
- 核心参数：
  - **N**: DFT 的点数，决定频域的离散采样点数（频率分辨率为  $1/N$ ）。
  - **Fre**: 信号的归一化频率（实际频率 =  $\text{Fre} \times \text{采样频率} / N$ ）。
  - **phai**: 信号的初始相位，影响 DFT 结果的相位特性。
  - **highlight**: 用于后续可视化时重点标注的频域点索引。

## 3. 构造原始信号

```
Signal_Type = COMPLEX_SIGNAL; % 选择复数信号
```

```
n = 0:N-1; % 时域采样索引（0 到 N-1）
```

```
if Signal_Type == COMPLEX_SIGNAL
```

```
    x = exp(1j*2*pi*Fre*n/N + phai); % 复数指数信号（欧拉公式形式）
```

```
else
```

```
    x = cos(2*pi*Fre*n/N + phai); % 实数余弦信号
```

end

- 时域索引  $n$ : 生成 0 到  $N-1$  的整数, 对应  $N$  个采样点。
- 信号构造:
  - 复数信号: 用  $\exp(1j*2\pi*Fre*n/N + \text{phai})$  表示, 包含幅度、频率、相位信息, 是 DFT 分析的典型输入。
  - 实数信号: 用余弦函数表示, 本质是两个共轭复数指数信号的叠加 (欧拉公式展开)。

#### 4. DFT 计算 (两种方法)

DFT 的数学定义为:  $X[k] = \sum_{n=0}^{N-1} x[n]e^{-\frac{j2\pi kn}{N}}, (k = 0, 1, \dots, N-1)$ , 代码实现了两种计算方式:

##### (1) 循环迭代法

```
X_loop = zeros(1,N); % 初始化 DFT 结果
for k = 0:N-1          % 遍历每个频域点 k
    for nn = 0:N-1      % 遍历每个时域点 n
        % 累加: 时域信号×旋转因子 (负指数项)
        X_loop(k+1) = X_loop(k+1) + x(nn+1) * exp(-1j*2*pi/N*k*nn);
    end
end
```

end

- 直接按 DFT 定义用双重循环计算: 外层循环遍历频域索引  $k$ , 内层循环遍历时域索引  $n$ , 累加每个  $x[n]$  与旋转因子  $e^{-\frac{j2\pi kn}{N}}$  的乘积。
- 缺点: 时间复杂度为  $(O(N^2))$ , 效率较低 (适合理解原理, 不适合大  $N$ )。

##### (2) 矩阵法

```
% 构造 N×N 的 DFT 矩阵 W (元素为旋转因子)
W = exp(-1j*2*pi/N*(0:N-1)*(0:N-1));
% 矩阵乘法实现 DFT: X = x · W (行向量×矩阵)
X_matrix = (W * x.').';
```

- DFT 矩阵  $W$ : 利用矩阵外积 $(0:N-1)*(0:N-1)$ 生成所有  $k*n$  组合, 再乘以旋转因子系数, 得到  $N \times N$  的 DFT 矩阵 (第  $k$  行  $n$  列元素为  $e^{-\frac{j2\pi kn}{N}}$ 。
- 矩阵乘法: 通过  $x$  与  $W$  的矩阵乘法直接得到 DFT 结果, 本质与循环法一致, 但代码更简洁, 适合数学推导。

## 5. IDFT 验证 (DFT 与 IDFT 的互逆性)

% 构造 IDFT 矩阵  $W_{inv}$  (逆变换矩阵)

```
W_inv = exp(1j*2*pi*(0:N-1)*(0:N-1)/N)/N;
```

% 验证  $W \times W_{inv}$  是否为单位矩阵 (理论上应成立)

```
I_check = W * W_inv;
```

```
I_check = round(I_check, 2); % 保留两位小数
```

```
disp('验证 W * W_inv 是否为单位矩阵 (保留两位小数):');
```

```
disp(I_check);
```

- IDFT 矩阵定义:  $W_{inv}[k, n] = \frac{1}{N} e^{j2\pi kn/N}$  (与 DFT 矩阵相比, 指数符号相反且除以  $N$ )。
- 互逆性验证: 理论上, DFT 矩阵  $W$  与 IDFT 矩阵  $W_{inv}$  的乘积应为单位矩阵 (对角线为 1, 其余为 0)。代码通过计算  $W \times W_{inv}$  并显示结果, 验证这一数学性质。

## 6. 结果输出与可视化

### (1) 结果对比打印

```
disp('对比 DFT 循环 vs 矩阵结果:');
```

```
for k = 1:N
```

```
    if ismember(k, highlight) % 标红的点用--前缀标识
```

```
        fprintf('--X[%d] 循环= %.2f%+.2fj, 矩阵= %.2f%+.2fjn', ...
```

```
            k-1,      real(X_loop(k)),      imag(X_loop(k)),      real(X_matrix(k)),
```

```
            imag(X_matrix(k)));
```

```
    else
```

```
        fprintf('X[%d] 循环= %.2f%+.2fj, 矩阵= %.2f%+.2fjn', ...
```

```
            k-1,      real(X_loop(k)),      imag(X_loop(k)),      real(X_matrix(k)),
```

```
imag(X_matrix(k));
```

```
end
```

```
end
```

- 打印每个频域点  $k$  (0 到  $N-1$ ) 的 DFT 结果, 对比循环法 ( $X_{\text{loop}}$ ) 和矩阵法 ( $X_{\text{matrix}}$ ) 的实部与虚部, 验证两种方法的一致性。
- 对 **highlight** 索引的点添加--前缀, 方便重点观察。

## (2) 可视化展示 (同一窗口分左右两图)

- **左图: 极坐标复平面图** 用极坐标展示 DFT 结果的复数特性 (模值 = 极径, 相位 = 极角):
- **右图: DFT 模值离散图** 用离散杆图展示 DFT 模值 (信号在各频率点的能量) 随  $k$  的变化:

同时展示 DFT 结果的**幅度 (极径)** 和**相位 (极角)**: 幅度反映信号在对应频率点的能量强度, 相位反映信号在该频率的相位偏移, 两者共同构成完整的频域信息 (工程中相位对信号重建至关重要, 如通信中的相位调制)。

**重点标注功能 (highlight 参数)**: 通过红色圆点突出指定频率点 (如[2,3,4]), 并显示其具体复数值 (实部 + 虚部), 方便观察特定频率的细节 (例如当信号频率  $F_{\text{re}}=1.2$  时, 可重点分析其附近频率点的能量分布)。

**文字偏移处理 (offset\_angle = -offset\_angle)**: 避免标注文字重叠, 确保复杂结果的可读性, 体现了可视化的实用性设计。

## DFT 模值离散图 (右图)

用 **stem** 函数绘制离散杆图, 直观展示**幅度谱**  $X[k]$  随频率索引  $k$  的分布: 杆的高度代表该频率点的能量, 是频谱分析中最常用的可视化形式 (如音频均衡器的频谱显示)。

**坐标轴范围优化 (ylim(ax, [0, max(X\_mag)\*1.2]))**: 确保所有幅度值都能完整显示, 且留有一定余量避免顶部数据被截断, 符合工程绘图规范。



### 三、实验代码程序运行结果及分析

1. 实数信号,  $N=8$ ,  $F_{re}=1$ , 求第 1, 2, 3 点, 这里 1、2、3 点指  $X[0], X[1], X[2]$

```
% ===== 参数设置 =====
N = 8;           % DFT 点数
Fre = 1;         % 信号频率
phai = 0;        % 初相位, 题 1/题 3 用 0, 题 2 用 pi/4
highlight = [1,2,3]; % 要标红的点索引 (MATLAB 下标从 1 开始)

% ===== 构造信号 =====
% 题 1、2: 实信号 cos, 题 3: 复信号 exp(j*....)
Signal_Type = REAL_SIGNAL; % REAL_SIGNAL -> cos, COMPLEX_SIGNAL -> exp
```

验证  $W * W_{inv}$  是否为单位矩阵 (保留两位小数):

1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1

可以看到  $W * W_{inv}$  斜对角元素为 1, 是单位矩阵, 验证了 DFT 与 IDFT 互为逆矩阵。

对比 DFT 循环 vs 矩阵结果:

--X[0] 循环= -0.00+0.00j, 矩阵= -0.00+0.00j

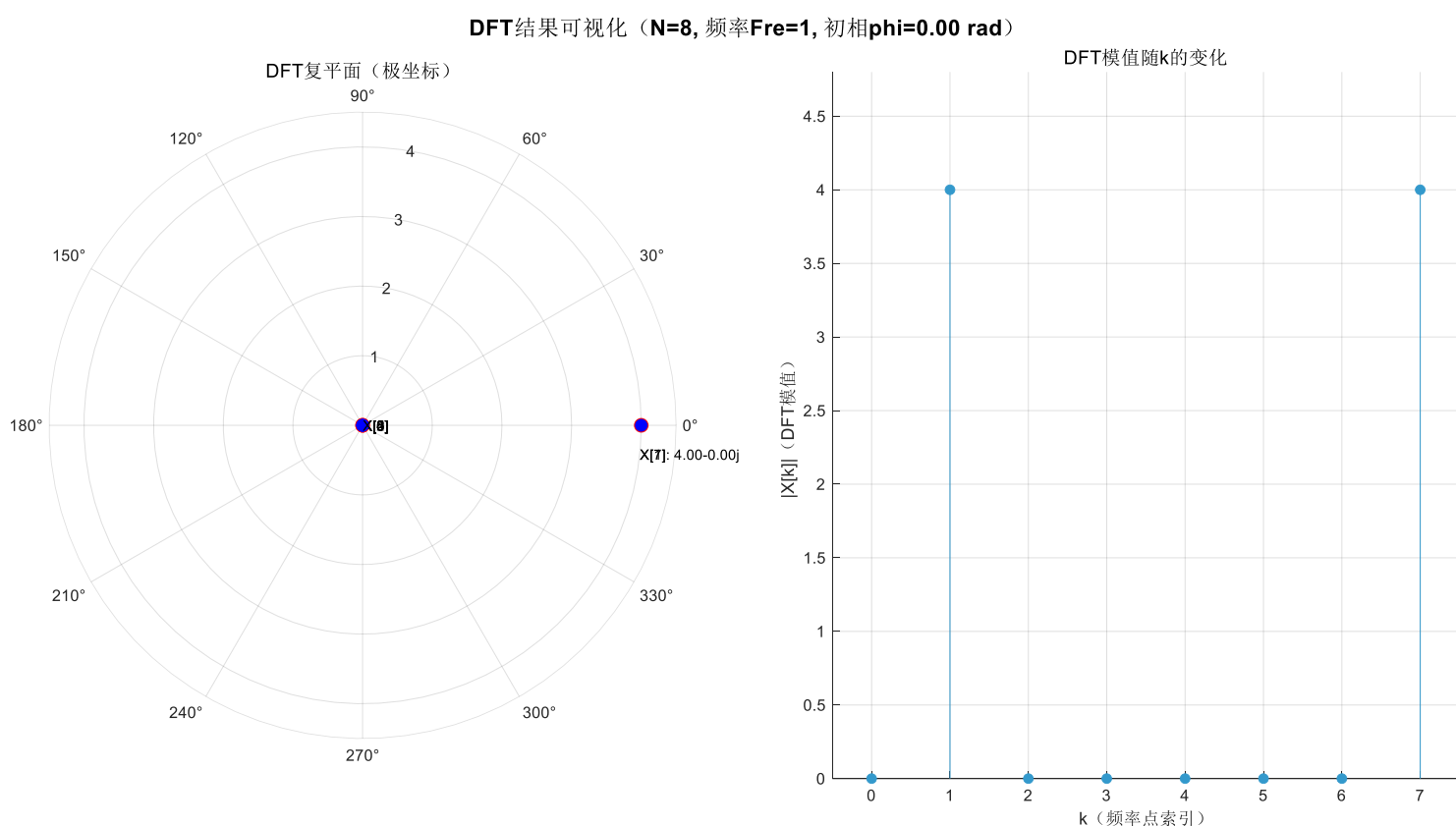
--X[1] 循环= 4.00-0.00j, 矩阵= 4.00-0.00j

--X[2] 循环= -0.00-0.00j, 矩阵= -0.00-0.00j

X[3] 循环= 0.00-0.00j, 矩阵= 0.00-0.00j

X[4] 循环= 0.00-0.00j , 矩阵= 0.00-0.00j  
X[5] 循环= 0.00-0.00j , 矩阵= 0.00-0.00j  
X[6] 循环= 0.00-0.00j , 矩阵= 0.00-0.00j  
X[7] 循环= 4.00+0.00j , 矩阵= 4.00+0.00j

可以看到，DFT 循环法与矩阵法的运算结果一样，可以互相验证程序计算的准确性。其中，要求的三个点 X[0],X[1],X[2]程序计算结果与前面的手算结果一致，可以验证其正确性。



DFT 模值图展示了 DFT 系数的模值 $|X[k]|$ 随频率索引  $k(k=0,1,...,N-1, \text{此处 } N=8)$  的变化, 图中仅在  $k=1$  和  $k=7$  处出现明显峰值(模值约为 4), 其余  $k$  值(如  $k=0,2,3,4,5,6$ ) 的模值几乎为 0, 这与构造的单频实信号  $x(n)=\cos(2\pi \cdot 1 \cdot n/8)$  的频谱特性直接相关。根据 DFT 的定义,  $X[k]$  是信号  $x(n)$  与复指数基函数  $e^{(-j2\pi kn/N)}$  的内积, 反映了信号在该频率分量上的能量强度; 而实余弦信号可通过欧拉公式分解为两个复指数信号的

叠加，即  $x(n)=[e^{j2\pi n/8}+e^{-j2\pi n/8}]/2$ ，其中第一个复指数分量与  $k=1$  对应的基函数  $e^{-j2\pi \cdot 1 \cdot n/8}$  完全匹配，第二个复指数分量  $e^{-j2\pi n/8}$  等价于  $e^{j2\pi \cdot 7 \cdot n/8}$ （因  $-1 \equiv 7 \pmod{8}$ ），与  $k=7$  对应的基函数  $e^{-j2\pi \cdot 7 \cdot n/8}$  完全匹配。根据复指数求和的正交性原理，当基函数与信号中的某个分量匹配时（即指数项为  $0 \pmod{N}$ ），求和结果为  $N$ ，否则相互抵消为  $0$ ，因此  $X[1]$  和  $X[7]$  的模值均为  $N/2=4$ ，而其他  $k$  值因基函数与信号分量不匹配，求和后模值接近  $0$ 。同时，实信号的 DFT 具有共轭对称性，即  $X[N-k]=X^*[k]$ ，这使得  $k=1$  和  $k=7$  处的模值相等，形成对称的峰值分布，整个模值图清晰呈现了单频实信号的能量集中在两个对称频率点上的特性，是 DFT 正交滤波特性的直接体现。

2. 实数信号， $N=8$ ， $Fre=2$ ，加初相  $45^\circ$ ， $X[1], X[2], X[3]$

```
% ===== 参数设置 =====
N = 8; % DFT 点数
Fre = 2; % 信号频率
phai = pi/4; % 初相位
highlight = [2,3,4]; % 要标红的点索引（MATLAB 下标从 1 开始）

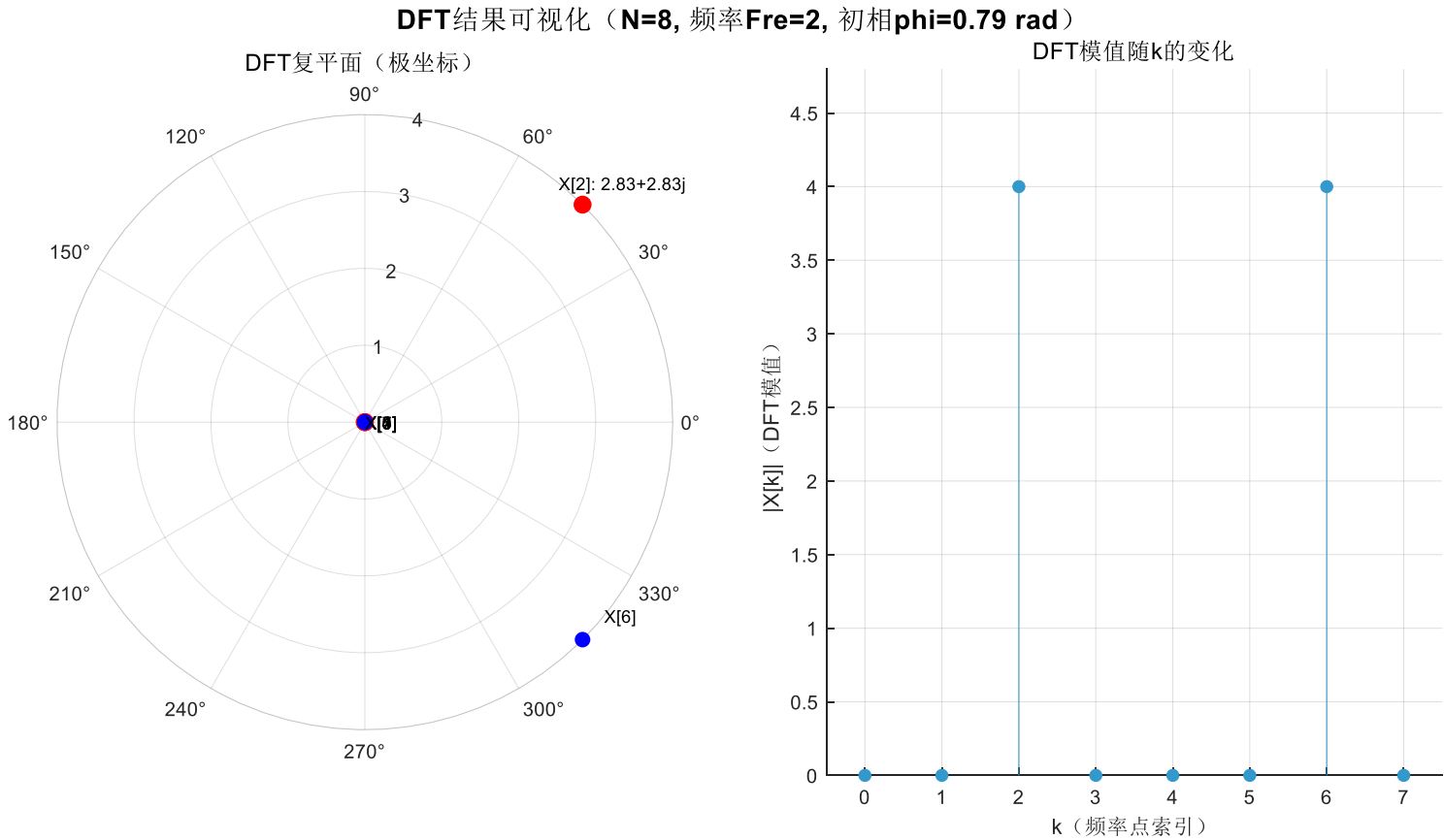
% ===== 构造信号 =====
Signal_Type = REAL_SIGNAL; % REAL_SIGNAL -> cos, COMPLEX_SIGNAL -> exp
```

对比 DFT 循环 vs 矩阵结果:

```
X[0] 循环= 0.00+0.00j , 矩阵= 0.00+0.00j
--X[1] 循环= -0.00+0.00j , 矩阵= -0.00+0.00j
--X[2] 循环= 2.83+2.83j , 矩阵= 2.83+2.83j
--X[3] 循环= 0.00-0.00j , 矩阵= 0.00-0.00j
X[4] 循环= 0.00-0.00j , 矩阵= 0.00-0.00j
X[5] 循环= 0.00+0.00j , 矩阵= 0.00+0.00j
X[6] 循环= 2.83-2.83j , 矩阵= 2.83-2.83j
X[7] 循环= 0.00+0.00j , 矩阵= 0.00+0.00j
```

可以看到，DFT 循环法与矩阵法的运算结果一样，可以互相验证程序计算的正

确性。其中，要求的三个点  $X[1]$ ,  $X[2]$ ,  $X[3]$ , 程序计算结果与前面的手算结果一致，可以验证其正确性。



从 DFT 的定义公式  $X[k] = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn}$  来看，当输入信号为复指数信号  $x(n) = e^{j(\frac{2\pi}{N}n + \text{phai})}$  时，代入 DFT 公式后，根据复指数序列的正交性：只有当基函数的频率（由  $k$  索引，对应频率为  $\frac{2\pi k}{N}$ ）与信号本身的频率  $\frac{2\pi \text{Fre}}{N}$  匹配（即  $k = \text{Fre}$ ）时，求和会得到显著的结果，此时  $X[\text{Fre}]$  的模值会很大；同时，由于 DFT 的频率具有共轭对称性（对于复指数信号，当  $N = 8$  时， $k = N - \text{Fre} = 8 - 2 = 6$  也会与信号频率形成“匹配”的正交分量），所以  $k = 6$  处的模值也会很大。而当  $k$  取其他值时，信号与基函数  $e^{-j\frac{2\pi}{N}kn}$  正交，求和结果的模值会非常小（接近 0）。从模值图来看，确实在  $k = 2$  和  $k = 6$  处出现了明显的峰值，其他  $k$  值（如  $k = 0, 1, 3, 4, 5, 7$ ）对应的模值几乎为 0，这直观体现了 DFT 的频率分析本质——将信号分解到以  $k$  为索引的离散频率分量上，模值的大小反映了对应频率分量的幅度强弱，因此该图清晰

验证了单频复指数信号通过 DFT 后，能量会集中在与自身频率对应的  $k = \text{Fre}$  和  $k = N - \text{Fre}$  这两个频率点上，符合 DFT 的正交分解原理与复指数序列的频率特性。

### 3. 复数信号， $N=8$ ， $\text{Fre}=3$ ， $X[1], X[2], X[3]$

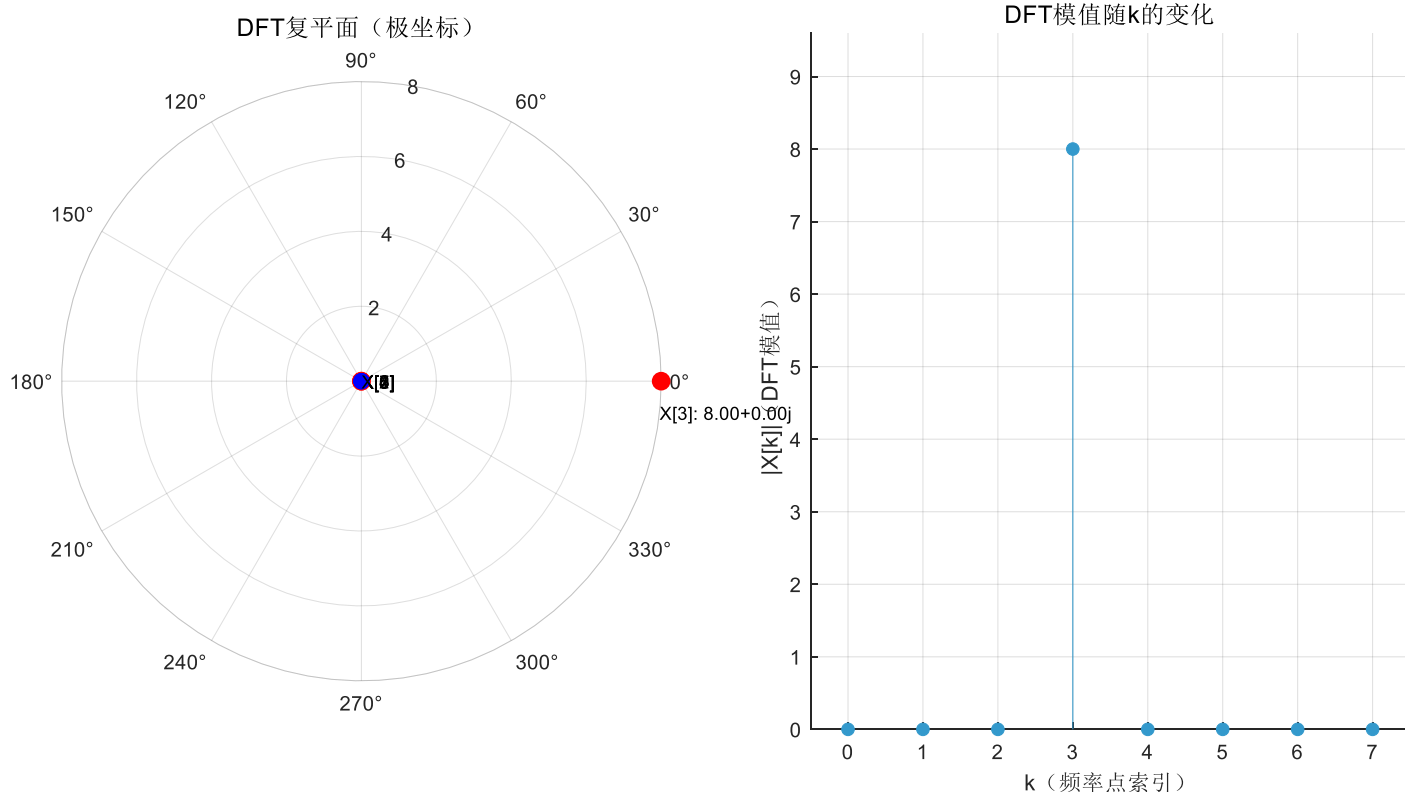
```
% ===== 参数设置 =====  
N = 8; % DFT 点数  
Fre = 3; % 信号频率  
phai = 0; % 初相位  
highlight = [2,3,4]; % 要标红的点索引（MATLAB 下标从 1 开始）  
  
% ===== 构造信号 =====  
Signal_Type = COMPLEX_SIGNAL; % REAL_SIGNAL -> cos, COMPLEX_SIGNAL -> exp
```

对比 DFT 循环 vs 矩阵结果:

```
X[0] 循环= -0.00-0.00j , 矩阵= -0.00-0.00j  
--X[1] 循环= 0.00+0.00j , 矩阵= 0.00+0.00j  
--X[2] 循环= -0.00+0.00j , 矩阵= -0.00+0.00j  
--X[3] 循环= 8.00+0.00j , 矩阵= 8.00+0.00j  
X[4] 循环= -0.00+0.00j , 矩阵= -0.00+0.00j  
X[5] 循环= -0.00-0.00j , 矩阵= -0.00-0.00j  
X[6] 循环= 0.00+0.00j , 矩阵= 0.00+0.00j  
X[7] 循环= 0.00-0.00j , 矩阵= 0.00-0.00j
```

可以看到，DFT 循环法与矩阵法的运输结果一样，可以互相验证程序计算的准确性。其中，要求的三个点  $X[1]$ ,  $X[2]$ ,  $X[3]$ , 程序计算结果与前面的手算结果一致，可以验证其正确性。

DFT结果可视化 (N=8, 频率Fre=3, 初相phi=0.00 rad)



从 DFT 的原理和公式来看，对于构造的复指数信号  $x(n) = \exp\left(j\left(\frac{2\pi\text{Fre}\cdot n}{N} + \text{phai}\right)\right)$  (这里  $N = 8$   $\text{Fre} = 3$   $\text{phai} = 0$ )，其 DFT 的定义为  $X(k) = \sum_{n=0}^{N-1} x(n)\exp\left(-j\frac{2\pi kn}{N}\right)$ 。将  $x(n)$  代入后，可得  $X(k) = \exp(j \cdot 0) \sum_{n=0}^{N-1} \exp\left(j\frac{2\pi(3-k)n}{N}\right)$ 。根据等比数列求和性质，当  $3 - k \equiv 0(\text{mod}N)$  (即  $k = 3$ ) 时，求和式中每一项都是  $\exp(0) = 1$ ，总和为  $N = 8$ ，此时  $|X(k)| = 8$ ；而当  $k \neq 3$  时，该等比数列的公比不为 1 且项数为  $N$ ，求和结果为 0，因此  $|X(k)| = 0$ 。右侧的 DFT 模值图完全验证了这一点：只有频率点索引  $k = 3$  处的模值为 8，其余  $k = 0, 1, 2, 4, 5, 6, 7$  处的模值均为 0，直观体现了 DFT 对单频复指数信号的频率分析能力——即当信号为频率对应  $k = \text{Fre}$  的复指数时，DFT 会在该  $k$  处呈现出等于点数  $N$  的模值，其余频率点因无对应分量而模值为 0。

为什么 DFT 模值图复数信号是一根线，实数信号是共轭的？

1. 复数信号（如单频复指数  $x(n) = \exp\left(j\left(\frac{2\pi\text{Fre}\cdot n}{N} + \text{phai}\right)\right)$ ）

复指数信号是单一频率分量的载体：它只包含正频率(或 DFT 中对应  $k = \text{Fre}$  的频率分量)。根据 DFT 的定义，当信号为  $\exp(j\omega n)$  时，其 DFT 会在  $k$  与  $\omega$  匹配的位置（即  $k = \text{Fre}$ ）出现单个非零值，其余频率点因无对应分量而模值为 0。因此，复数信号的 DFT 模值图仅一根线（仅对应频率点有能量）。

2. 实数信号（如余弦信号  $x(n) = \cos\left(\frac{2\pi\text{Fre}\cdot n}{N} + \text{phai}\right)$ ）

实数信号可通过欧拉公式分解为正、负频率的复指数之和：

$$\cos(\theta) = \frac{\exp(j\theta) + \exp(-j\theta)}{2}$$

其中， $\exp(j\theta)$  对应正频率分量， $\exp(-j\theta)$  对应负频率分量。而 DFT 的频率是循环的（ $-k$  等价于  $N - k$ ），因此负频率分量会映射到  $k = N - \text{Fre}$  处。

根据 DFT 的共轭对称性质：实数序列的 DFT 满足  $X(k) = X^+(N - k)$ （ $X^+$  表示共轭）。这意味着：正频率分量（ $k = \text{Fre}$ ）的 DFT 与负频率分量（ $k = N - \text{Fre}$ ）的 DFT 是共轭关系，且模值相等（ $|X(k)| = |X(N - k)|$ ）。因此，实数信号的 DFT 模值图会在  $k = \text{Fre}$  和  $k = N - \text{Fre}$  处出现两根对称的线，呈现共轭对称的特征。

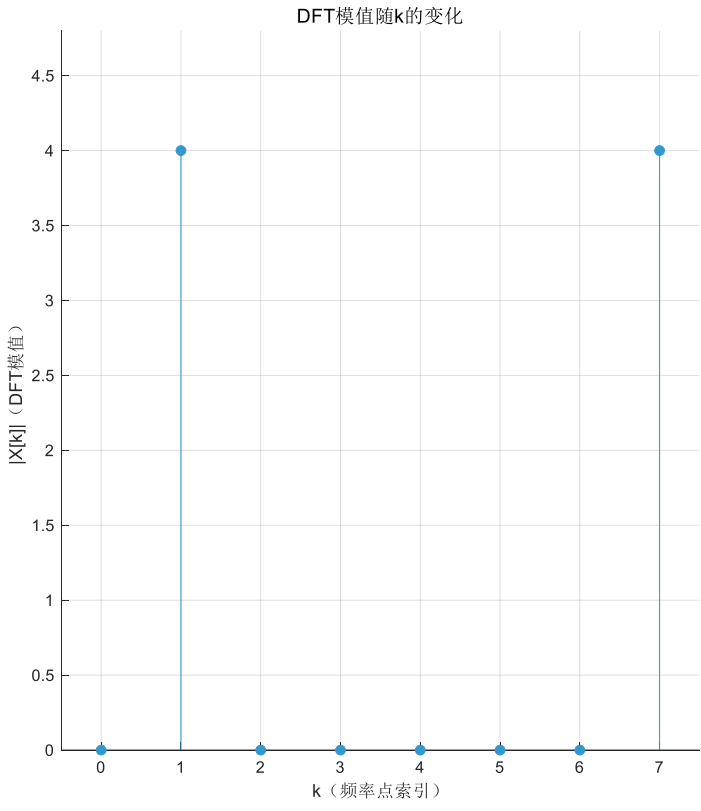
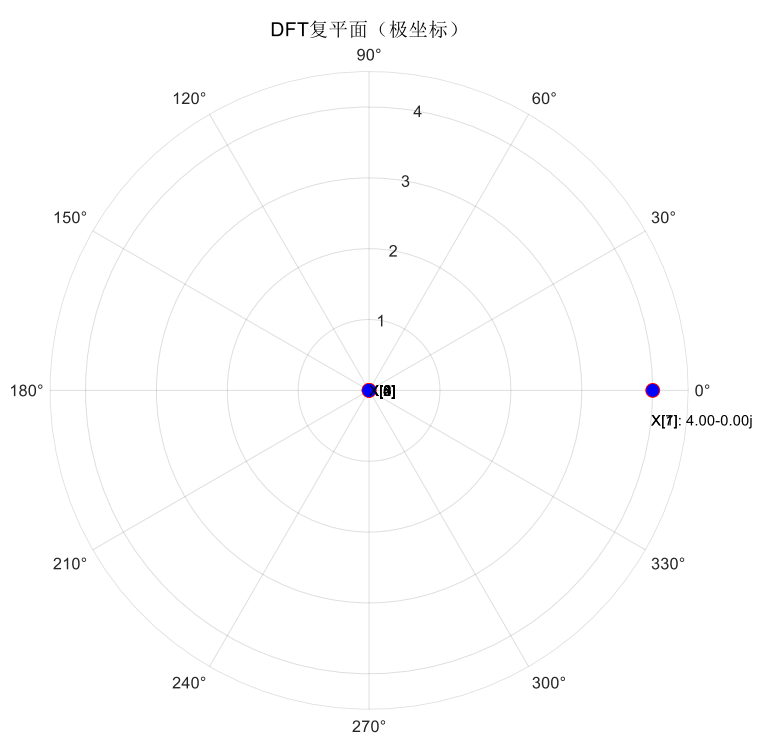
## 实验五 数字 fre=1.2 观察频谱泄露

1. 实数信号， $N=8$ ， $\text{Fre}=1$ ，求第 1, 2, 3 点，这里 1、2、3 点指  $X[0], X[1], X[2]$

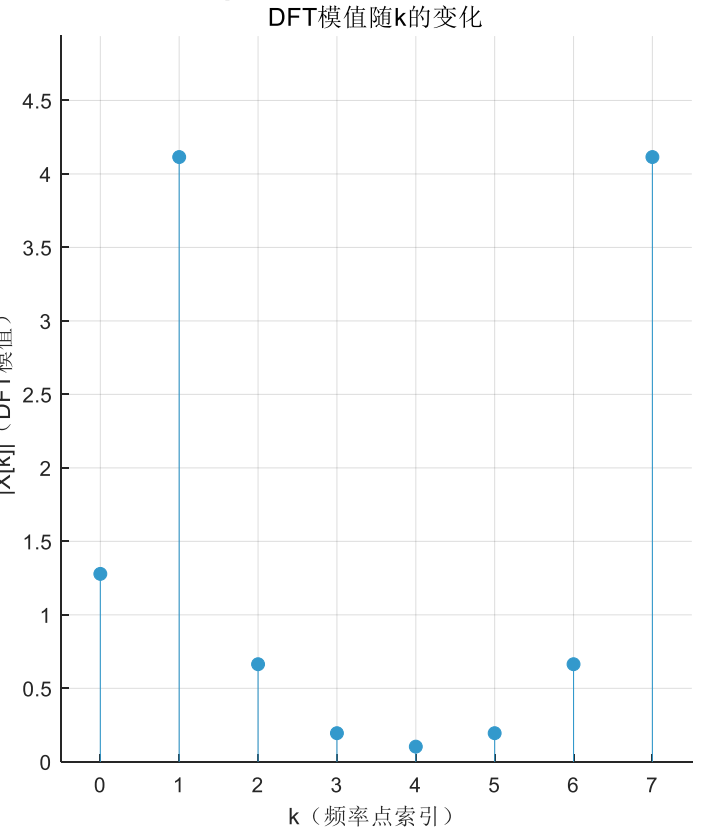
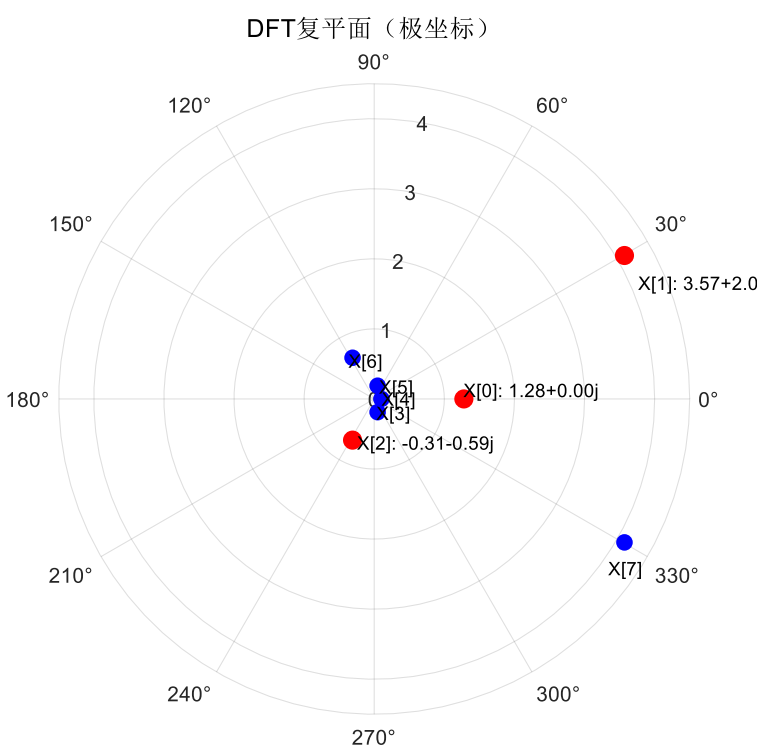
```
% ===== 参数设置 =====
N    = 8;                % DFT 点数
Fre  = 1.2;              % 信号频率
phai = 0;                % 初相位
highlight = [1,2,3];    % 要标红的点索引（MATLAB 下标从 1 开始）

% ===== 构造信号 =====
Signal_Type = REAL_SIGNAL; % REAL_SIGNAL -> cos, COMPLEX_SIGNAL -> exp
```

DFT结果可视化 (N=8, 频率Fre=1, 初相phi=0.00 rad)



DFT结果可视化 (N=8, 频率Fre=1.200000e+00, 初相phi=0.00 rad)





2. 实数信号，N=8，Fre=2，加初相  $45^\circ$ ，X[1],X[2],X[3]

% ===== 参数设置 =====

N = 8; % DFT 点数

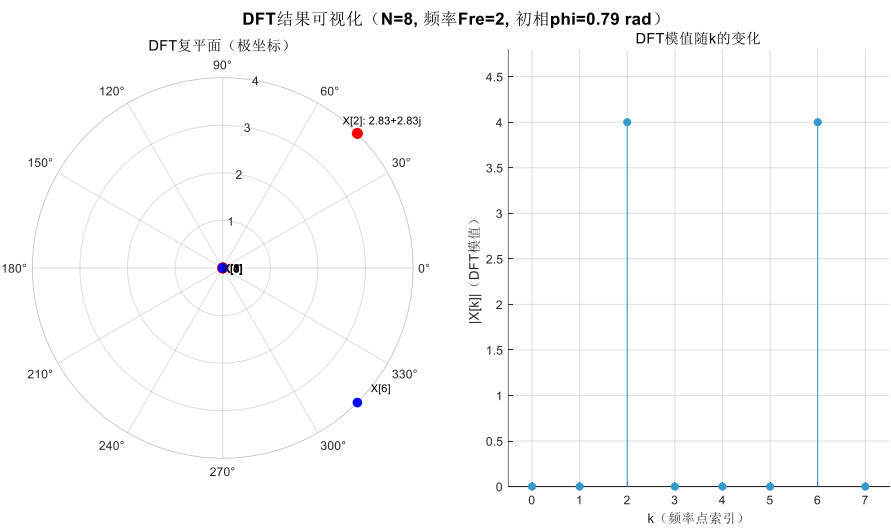
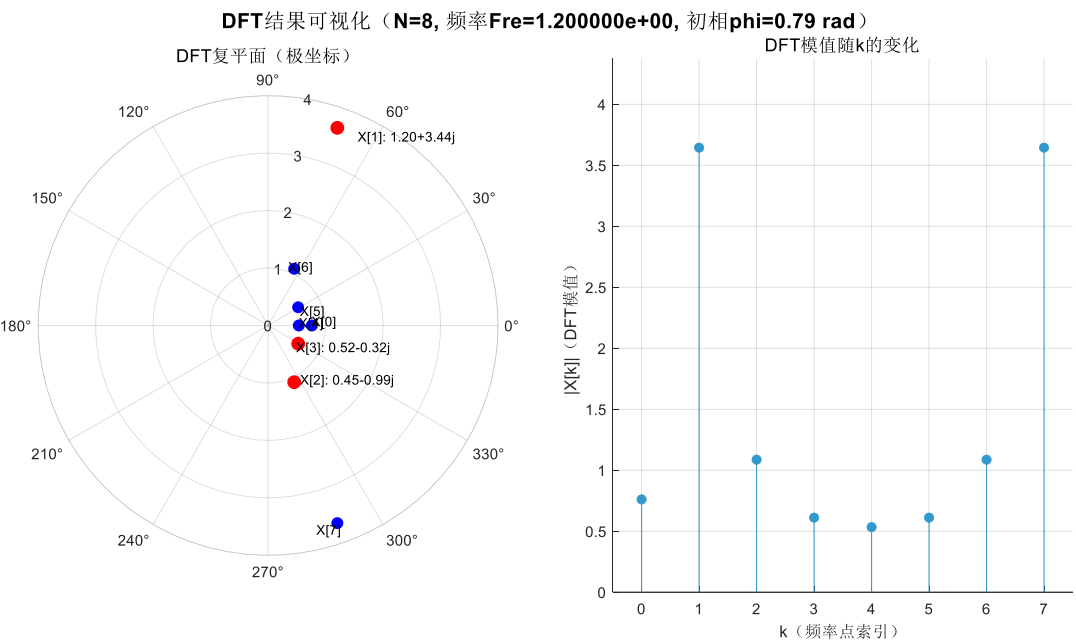
Fre = 1.2; % 信号频率

phai = pi/4; % 初相位

highlight = [2,3,4]; % 要标红的点索引 (MATLAB 下标从 1 开始)

% ===== 构造信号 =====

Signal\_Type = REAL\_SIGNAL; % REAL\_SIGNAL -> cos, COMPLEX\_SIGNAL -> exp

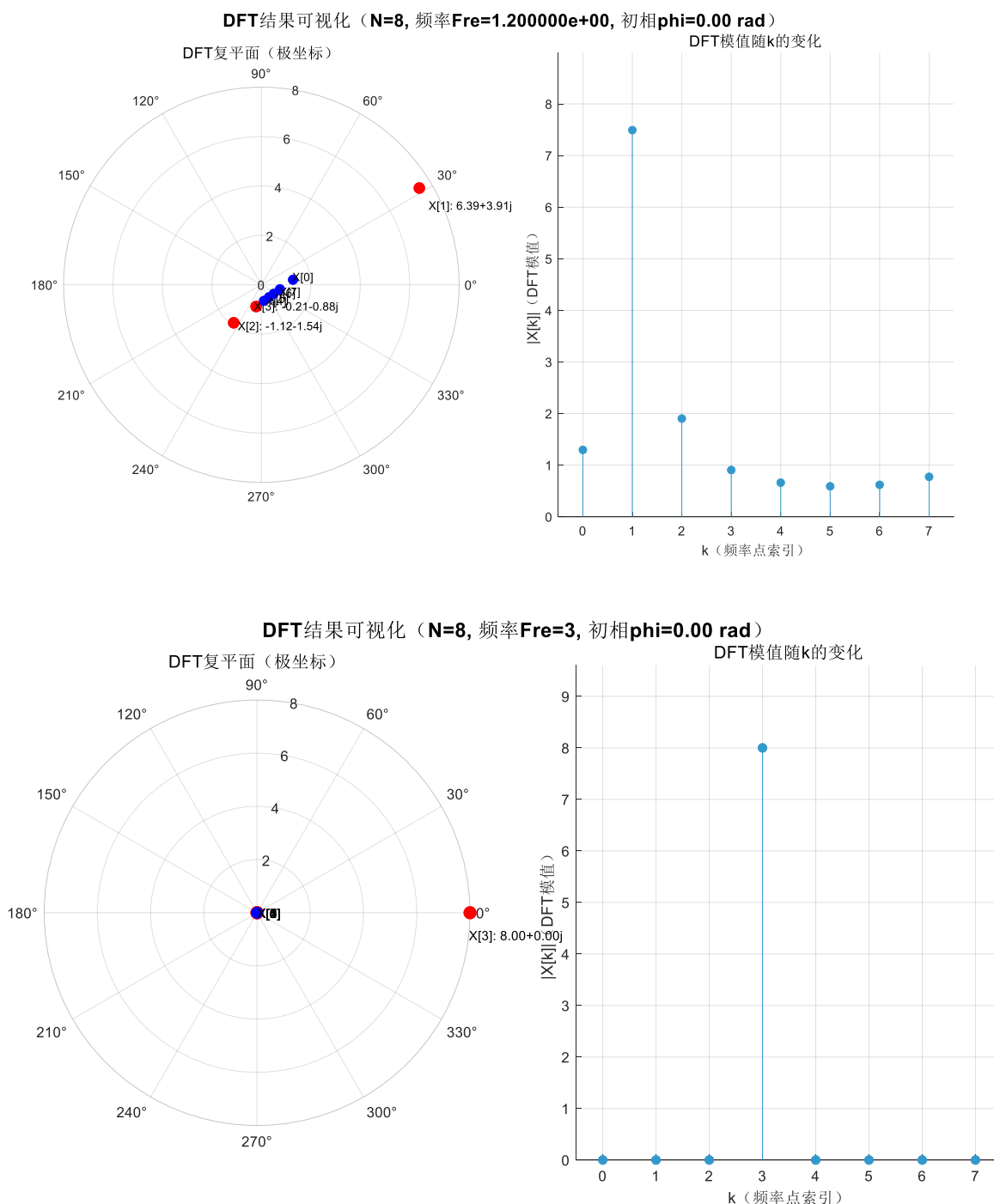


从 DFT 的原理来看，DFT 是对离散信号进行频域采样，其频率分辨率为  $f_s/N$  ( $f_s$  为采样频率，这里可简化为单位采样频率，故分辨率为  $1/N = 0.125$ )。当信号频率  $F_{re}$  是频率分辨率的整数倍时（如第一张图中  $F_{re} = 1$ ，是  $0.125 \times 8$ ，即整数倍），信号的频域能量会集中在对应的频率点索引  $k$  上（此时  $k = 1$  和  $k = 7$ ，因 DFT 具有周期性， $k$  和  $k + N$  对应相同频率）。而当  $F_{re} = 1.2$  时，它不是频率分辨率的整数倍 ( $1.2 \div 0.125 = 9.6$ ，非整数)，此时信号不满足“整周期截断”的条件，会发生频谱泄露。根据 DFT 的定义式  $X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$ ，这里  $x[n] = e^{j2\pi \cdot 1.2n/8}$ ，代入后  $X[k] = \sum_{n=0}^7 e^{j2\pi(1.2-k/8)n}$ ，这是等比数列求和，当  $1.2 - k/8 \neq 0$ （即  $k$  不是使该式为零的整数，而  $k$  只能取 0 到 7 的整数），求和结果的模值不为零，意味着能量会从“理想”的频率点（本应对应非整数的  $k$ ）扩散到周围的整数  $k$  点上。反映在 DFT 模值图中，就不再是仅  $k = 1$  和  $k = 7$  有大的模值，而是  $k = 0$ 、 $k = 1$ 、 $k = 2$ 、 $k = 6$ 、 $k = 7$  等多个点都出现了非零的模值，原本集中的能量被“泄露”到了相邻的频率点，这就是频谱泄露现象的直观体现。

### 3. 复数信号， $N=8$ ， $F_{re}=3$ ， $X[1], X[2], X[3]$

```
% ===== 参数设置 =====
N = 8; % DFT 点数
Fre = 1.2; % 信号频率
phai = 0; % 初相位
highlight = [2,3,4]; % 要标红的点索引（MATLAB 下标从 1 开始）

% ===== 构造信号 =====
Signal_Type = COMPLEX_SIGNAL; % REAL_SIGNAL -> cos, COMPLEX_SIGNAL -> exp
```



从 DFT 的原理来看, DFT 的频率点是离散的, 由公式  $X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$  (其中  $k = 0, 1, \dots, N-1$  为频率点索引) 决定, 只有当信号的归一化频率 Fre 恰好等于某个整数  $k$  (即信号频率与 DFT 的离散频率点对齐) 时, 能量才会集中在对应的  $k$  点上。当  $\text{Fre} = 1.2$  (非整数) 时, 信号  $x[n] = e^{j2\pi \cdot 1.2 \cdot n/N + \phi}$  的频率不与 DFT 的离散频率点 ( $k = 0, 1, \dots, 7$ , 对应归一化频率  $0, 1/8, 2/8, \dots, 7/8$ ) 对齐, 此时 DFT 的正交性不

再能让能量完全集中在单个  $k$  点。从模值图可见，能量不再仅出现在某一个  $k$  点，而是分散到了多个  $k$  点（如  $k = 0, 1, 2, 3$  等都有非零模值），其中  $k = 1$  处模值最大，但其他频率点也存在明显的幅值，这就是频谱泄露的体现——原本集中在单一频率的信号能量，因频率与 DFT 离散栅格不匹配，被“泄露”到相邻的多个频率点上。而若信号频率与 DFT 离散频率点对齐（如  $F_{re} = 3$  时， $k = 3$  为整数），模值会集中在单个  $k$  点，其他点模值接近 0，无频谱泄露； $F_{re} = 1.2$  的模值图则直观展现了这种因频率不匹配导致的能量分散现象。

## 实验六 噪声 中心极限定理

```
close all;
clear;clc;

% -----
% 用户设置区域
% -----
N = 800000;           % 数据量
Ms = [2, 4, 8, 16];   % 要测试的 M 值
dist_type = 'uniform'; % 分布类型: 'uniform', 'gaussian', 'rayleigh',
'exponential', 'laplace'
% -----

num_M = length(Ms);

% 存储结果的矩阵
results = zeros(num_M, 5); % 均值, 标准差, sigma1%, sigma2%, sigma3%

% 创建图形
figure('Name', [dist_type, '分布的随机数和分布'], 'Position', [10, 10, 1000,
600]);

% 循环处理每个 M 值
for i = 1:num_M
    M = Ms(i);
```

```

% 根据选择的分布生成随机数
switch lower(dist_type)
    case 'uniform'
        % 均匀分布 [0,1)
        xx = rand(M, N);
    case 'gaussian'
        % 高斯分布 (均值 0, 方差 1)
        xx = randn(M, N);
    case 'rayleigh'
        % 瑞利分布 (参数 1)
        xx = raylrnd(1, M, N);
    case 'exponential'
        % 指数分布 (参数 1)
        xx = exprnd(1, M, N);
    case 'laplace'
        % 拉普拉斯分布 (位置 0, 尺度 1)
        % 生成方法: 利用指数分布的特性
        u = rand(M, N) - 0.5;
        xx = -sign(u) .* log(1 - 2*abs(u));
    otherwise
        error('不支持的分布类型, 请选择: uniform, gaussian, rayleigh, exponential, laplace');
end

% 求和
xx_sum = sum(xx, 1);

% 计算统计量
mean_val = mean(xx_sum);
std_val = std(xx_sum);

% 计算各 sigma 范围外的百分比
sigma1_percent = (sum(xx_sum < (mean_val - std_val)) + sum(xx_sum > (mean_val + std_val))) / N * 100;
sigma2_percent = (sum(xx_sum < (mean_val - 2*std_val)) + sum(xx_sum > (mean_val + 2*std_val))) / N * 100;
sigma3_percent = (sum(xx_sum < (mean_val - 3*std_val)) + sum(xx_sum > (mean_val + 3*std_val))) / N * 100;

% 存储结果
results(i, :) = [mean_val, std_val, sigma1_percent, sigma2_percent, sigma3_percent];

```

```

% 绘制子图
subplot(2, 2, i);
hist(xx_sum, 300);
title(['M = ', num2str(M)], 'FontSize', 10);
xlabel('求和结果', 'FontSize', 8);
ylabel('频数', 'FontSize', 8);
set(gca, 'FontSize', 7);
grid on;
end

% 添加总标题
sgtitle([upper(dist_type), '分布下随机数和的分布 (M = ', num2str(Ms), ')']);

% 调整布局
set(gcf, 'PaperPositionMode', 'auto');

% 显示统计结果对比
disp(['统计结果对比 (分布类型: ', dist_type, ', M 值从左到右分别为', num2str(Ms),
')']);
disp('-----');
disp('M 值 | 均值 | 标准差 | 1σ 外(%) | 2σ 外(%) | 3σ 外(%)');
disp('-----');
for i = 1:num_M
    fprintf('%d | %.4f | %.4f | %.4f | %.4f | %.6f\n', ...
        Ms(i), results(i,1), results(i,2), results(i,3), results(i,4),
        results(i,5));
end
disp('-----');
disp('正态分布理论参考: 1σ 外≈31.73%, 2σ 外≈4.55%, 3σ 外≈0.27%');

```

这段 MATLAB 代码的核心功能是**验证中心极限定理**，通过对不同分布的随机变量进行求和运算，观察当求和项数（M 值）增加时，求和结果的分布是否逐渐趋近于正态分布。以下是详细解释：

### 整体功能概述

中心极限定理指出：当样本量足够大时，无论原始随机变量服从何种分布，其多个独立随机变量的和（或均值）的分布会趋近于正态分布。

这段代码通过以下步骤验证这一理论：

1. 生成指定分布（均匀、高斯、瑞利等）的随机数；

2. 对不同数量 ( $M$  值) 的随机数求和;
3. 统计求和结果的分布特征 (均值、标准差、各  $\sigma$  范围外的比例);
4. 可视化不同  $M$  值下的求和分布, 并对比理论正态分布的特征。

## 代码结构与详细解释

### 1. 初始化与参数设置

```
close all; % 关闭所有图形窗口
```

```
clear; % 清空工作区变量
```

```
clc; % 清空命令行窗口
```

```
% 用户设置区域
```

```
N = 800000; % 数据量 (每个  $M$  值下生成  $N$  组求和结果)
```

```
Ms = [2, 4, 8, 16]; % 要测试的  $M$  值 (每组求和的随机变量数量)
```

```
dist_type = 'uniform'; % 原始随机变量的分布类型
```

- 初始化命令确保环境干净, 避免干扰;
- 参数  $N$  决定了统计的样本量 (越大结果越稳定);
- $Ms$  是核心变量, 用于测试不同求和项数对分布的影响;
- `dist_type` 指定原始随机变量的分布 (支持 5 种常见分布)。

### 2. 结果存储与图形准备

```
num_M = length(Ms); %  $M$  值的数量
```

```
results = zeros(num_M, 5); % 存储统计结果: 均值、标准差、 $1\sigma$  外(%),  
2 $\sigma$  外(%), 3 $\sigma$  外(%)
```

```
figure('Name', [dist_type, '分布的随机数和分布'], 'Position', [10, 10, 1000, 600]); % 创  
建图形窗口
```

- `results` 矩阵用于记录每个  $M$  值对应的统计指标;
- 创建一个图形窗口, 用于后续绘制不同  $M$  值下的求和分布直方图。

### 3. 核心循环: 生成随机数、求和与统计

对每个  $M$  值 ( $Ms(i)$ ) 执行以下操作:

#### 3.1 生成指定分布的随机数

```

switch lower(dist_type)
    case 'uniform'      % 均匀分布 [0,1)
        xx = rand(M, N);
    case 'gaussian'     % 高斯分布（均值 0，方差 1）
        xx = randn(M, N);
    case 'rayleigh'     % 瑞利分布（参数 1）
        xx = raylrnd(1, M, N);
    case 'exponential' % 指数分布（参数 1）
        xx = exprnd(1, M, N);
    case 'laplace'      % 拉普拉斯分布（位置 0，尺度 1）
        u = rand(M, N) - 0.5;
        xx = -sign(u) .* log(1 - 2*abs(u)); % 通过均匀分布转换生成
    otherwise
        error('不支持的分布类型');
end

```

- 生成  $M \times N$  的随机数矩阵 **xx**：每行代表 1 个随机变量，每列代表 1 组样本（共  $N$  组）；
- 支持 5 种分布，其中拉普拉斯分布通过均匀分布的数学转换生成（无内置函数时的常用方法）。

### 3.2 求和与统计量计算

```

xx_sum = sum(xx, 1); % 按列求和（每组样本的  $M$  个随机数相加，得到  $1 \times N$  的求和结果）

```

% 计算基本统计量

```

mean_val = mean(xx_sum); % 求和结果的均值

```

```

std_val = std(xx_sum); % 求和结果的标准差

```

% 计算落在均值 $\pm 1\sigma$ 、 $\pm 2\sigma$ 、 $\pm 3\sigma$  范围外的数据百分比



```

sigma1_percent = (sum(xx_sum < (mean_val - std_val)) + sum(xx_sum > (mean_val +
std_val))) / N * 100;
sigma2_percent = (sum(xx_sum < (mean_val - 2*std_val)) + sum(xx_sum > (mean_val +
2*std_val))) / N * 100;
sigma3_percent = (sum(xx_sum < (mean_val - 3*std_val)) + sum(xx_sum > (mean_val +
3*std_val))) / N * 100;

```

```

results(i, :) = [mean_val, std_val, sigma1_percent, sigma2_percent, sigma3_percent]; %
存储结果

```

- `xx_sum` 是核心结果：每个元素是  $M$  个随机数的和（共  $N$  个这样的和）；
- 统计量的意义：
  - 均值和标准差描述求和结果的集中趋势和离散程度；
  - `sigma1_percent` 等指标用于衡量偏离均值  $1/2/3$  个标准差的数据比例（正态分布中这些比例是固定的，用于对比）。

### 3.3 可视化求和结果的分布

```

subplot(2, 2, i); % 在 2×2 网格中绘制第 i 个子图
hist(xx_sum, 300); % 绘制求和结果的直方图（300 个 bins，提高分辨率）
title(['M = ', num2str(M)], 'FontSize', 10); % 标题标注当前 M 值
xlabel('求和结果', 'FontSize', 8);
ylabel('频数', 'FontSize', 8);
set(gca, 'FontSize', 7); % 调整坐标轴字体大小
grid on; % 显示网格

```

- 对每个  $M$  值绘制直方图，直观展示求和结果的分布形态；
- 随着  $M$  增大，直方图应逐渐呈现正态分布的“钟形”特征。

## 4. 结果展示与对比

```

sgtitle([upper(dist_type), '分布下随机数和的分布 (M = ', num2str(Ms), ')']); % 总标题
set(gcf, 'PaperPositionMode', 'auto'); % 调整图形布局

```

```
% 命令行输出统计结果
disp(['统计结果对比 (分布类型:', dist_type, ', M 值从左到右分别为', num2str(Ms), '):']);
disp('-----');
disp('M 值 | 均值      | 标准差   | 1 $\sigma$  外(%) | 2 $\sigma$  外(%) | 3 $\sigma$  外(%)');
disp('-----');
for i = 1:num_M
    fprintf('%d      | %.4f | %.4f | %.4f | %.4f | %.6f\n', ...
        Ms(i), results(i,1), results(i,2), results(i,3), results(i,4), results(i,5));
end
disp('-----');
disp('正态分布理论参考: 1 $\sigma$  外 $\approx$ 31.73%, 2 $\sigma$  外 $\approx$ 4.55%, 3 $\sigma$  外 $\approx$ 0.27%');


- 总标题标注当前测试的分布类型和 M 值;
- 命令行输出详细统计表格, 将实际计算的“ $\sigma$  外百分比”与正态分布的理论值对比 (理论值来自正态分布的概率特性)。

```

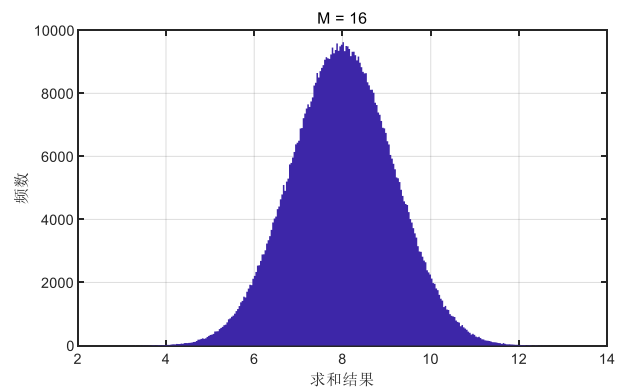
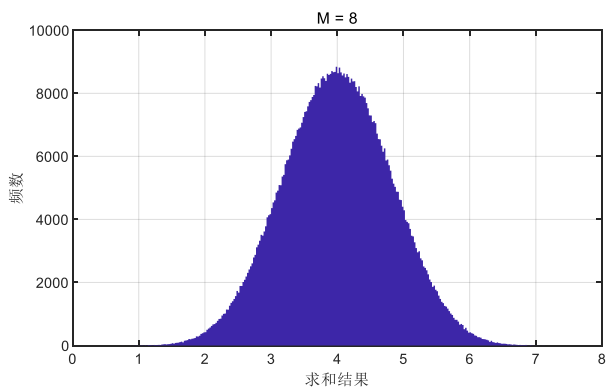
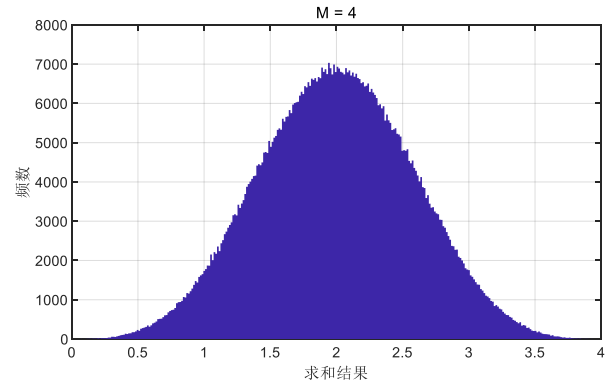
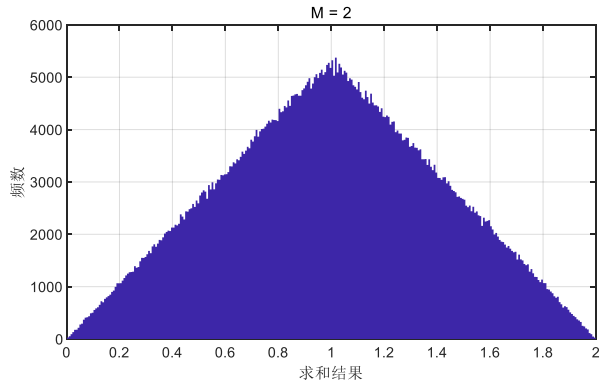
## 核心结论与预期效果

运行代码后, 会观察到以下现象 (验证中心极限定理):

无论原始分布是均匀、瑞利还是指数分布, 当 M 值增大 (如从 2 到 16) 时:

1. 求和结果的直方图逐渐接近“钟形” (正态分布形态);
2. 1 $\sigma$  外、2 $\sigma$  外、3 $\sigma$  外的百分比会逐渐接近正态分布的理论值 (31.73%、4.55%、0.27%), 验证“大量独立随机变量的和趋近于正态分布”的核心结论。

## UNIFORM分布下随机数和的分布 (M = 2 4 8 16)



统计结果对比 (分布类型: uniform, M 值从左到右分别为 2 4 8 16):

M 值 | 均值 | 标准差 | 1  $\sigma$  外(%) | 2  $\sigma$  外(%) | 3  $\sigma$  外(%)

2 | 1.0000 | 0.4077 | 34.9835 | 3.3611 | 0.000000

4 | 1.9990 | 0.5771 | 33.0949 | 4.2481 | 0.044375

8 | 4.0006 | 0.8158 | 32.3218 | 4.4240 | 0.163000

16 | 8.0004 | 1.1544 | 32.0149 | 4.4703 | 0.227500

正态分布理论参考: 1  $\sigma$  外  $\approx$  31.73%, 2  $\sigma$  外  $\approx$  4.55%, 3  $\sigma$  外  $\approx$  0.27%

从公式与原理角度分析: 首先, 设每个随机变量  $X_i \sim U(0,1)$  (均匀分布, 区间

$[0,1)$ 且独立同分布。根据期望与方差的性质，单个 $X_i$ 的均值 $\mu = \frac{0+1}{2} = 0.5$ ，方差 $\sigma^2 = \frac{(1-0)^2}{12} = \frac{1}{12}$ 当计算  $M$  个这样的变量和 $S_M = X_1 + X_2 + \cdots + X_M$ 时，理论均值为 $E[S_M] = M \times 0.5$ ，理论方差为 $D[S_M] = M \times \frac{1}{12}$ ，标准差为 $\sqrt{\frac{M}{12}}$

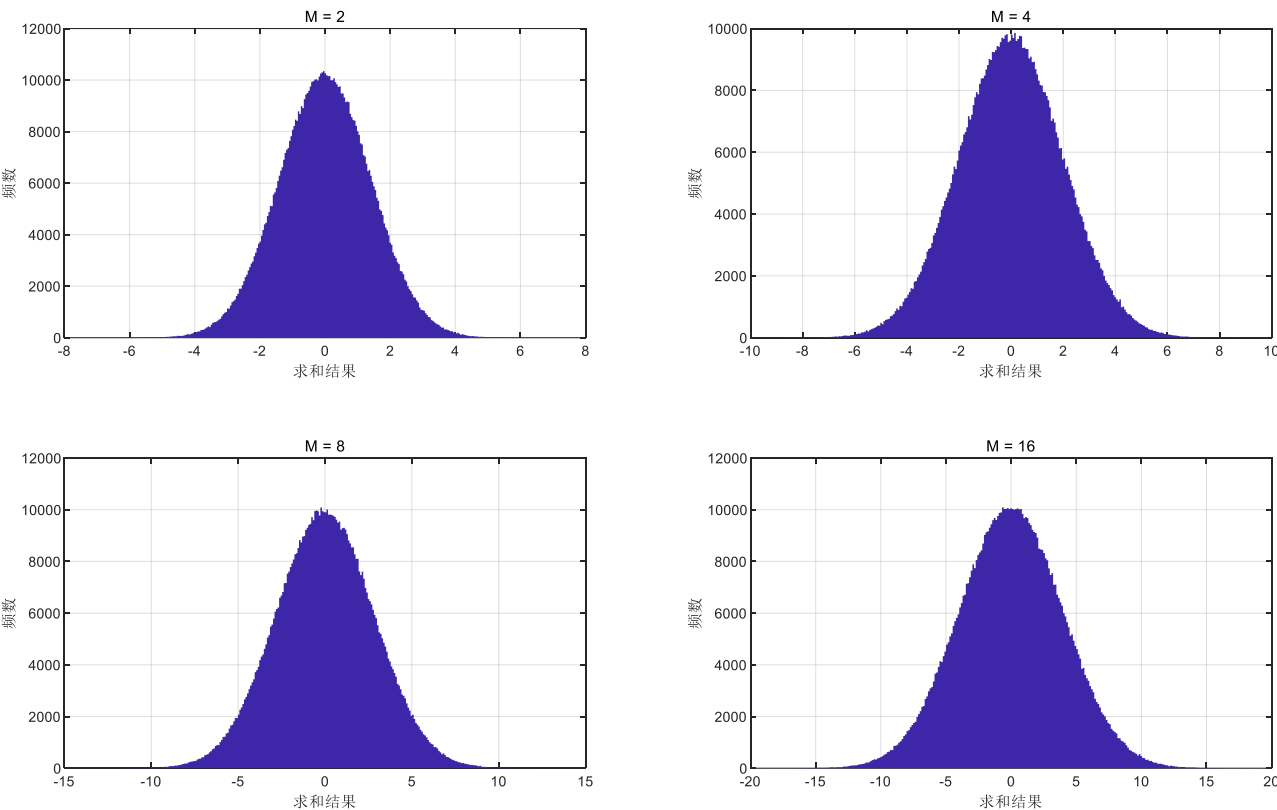
从图形看， $M = 2$ 时，和的分布为三角分布（两个独立均匀分布和的理论分布），此时偏离正态；随着 $M$ 增大（ $M = 4, 8, 16$ ），分布的”钟形”特征逐渐明显，越来越接近正态分布——这是中心极限定理的体现：无论原分布如何，独立同分布随机变量的和，当 $M$ 足够大时，分布趋近于正态分布。同时，横轴范围对应 $M$ 个 $[0,1)$ 变量和的取值范围 $[0, M)$ ，峰值位置与均值 $M/2$ 一致（如 $M = 2$ 时峰值在 1 附近， $M = 16$ 时峰值在 8 附近），验证了均值的理论计算。

从统计结果看，均值列（如 $M = 2$ 时约 1.0000， $M = 16$ 时约 8.0004）与理论值 $M/2$ 高度一致；标准差列（如 $M = 2$ 时约 0.4077，理论值 $\sqrt{2/12} \approx 0.4082$  $M = 16$ 时约 1.1544，理论值 $\sqrt{16/12} \approx 1.1547$ ）也与 $\sqrt{\frac{M}{12}}$ 几乎重合，说明方差的计算符合独立变量和的性质。

再看“ $\sigma$ 外百分比”：正态分布理论上， $1\sigma$ 外约31.73%、 $2\sigma$ 外约4.55%、 $3\sigma$ 外约0.27%。当 $M = 2$ 时， $1\sigma$ 外约34.98%（与理论差距大，因 $M$ 小、分布偏离正态）；随着 $M$ 增大（ $M = 4 \rightarrow 8 \rightarrow 16$ ）， $1\sigma$ 外百分比逐渐逼近31.73%， $2\sigma$ 外逼近4.55%， $3\sigma$ 外逼近0.27%（如 $M = 16$ 时  $3\sigma$ 外约0.2275%）。这进一步验证了中心极限定理： $M$ 越大，和的分布越接近正态，各区间比例越贴合正态分布的理论值。

综上，图形与统计结果共同体现了中心极限定理的核心：独立同分布随机变量的和，随项数 $M$ 增加，从非正态（如三角分布）逐渐收敛于正态分布，且均值、方差、区间比例均向正态分布的理论特征逼近。

GAUSSIAN分布下随机数和的分布 (M = 2 4 8 16)



统计结果对比 (分布类型: gaussian, M 值从左到右分别为 2 4 8 16):

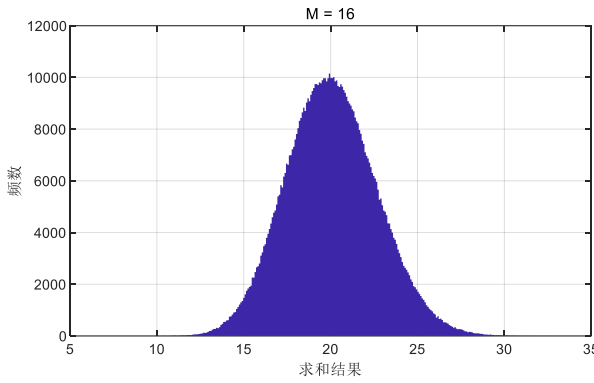
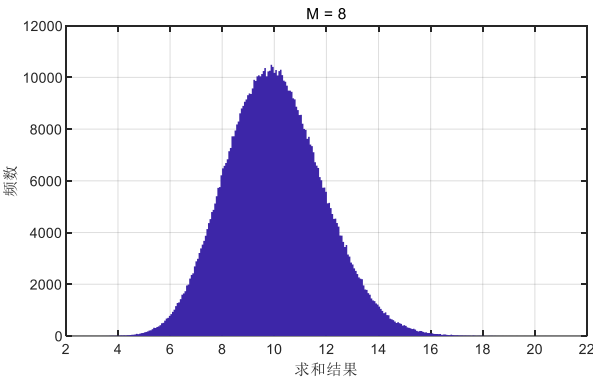
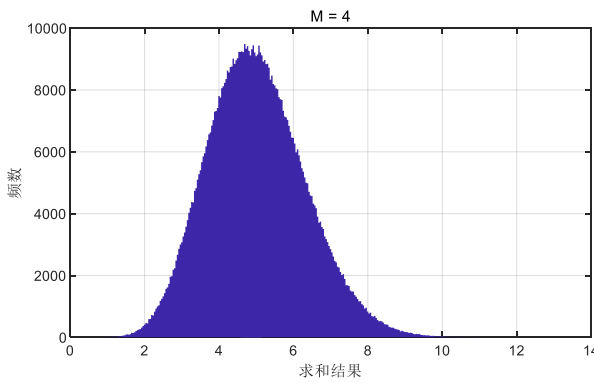
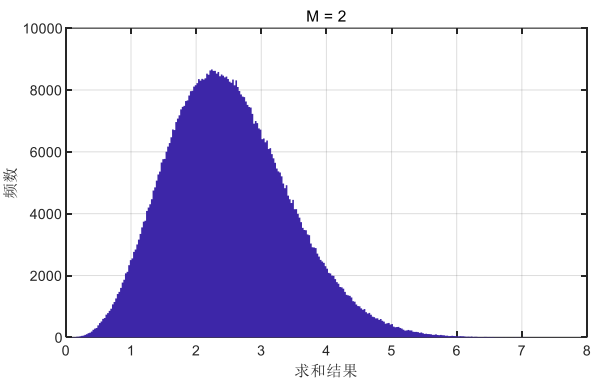
M 值	均值	标准差	1 $\sigma$ 外(%)	2 $\sigma$ 外(%)	3 $\sigma$ 外(%)
2	0.0007	1.4156	31.7165	4.5423	0.272875
4	-0.0029	2.0031	31.7814	4.5492	0.261875
8	0.0010	2.8291	31.7258	4.5409	0.280500
16	-0.0028	3.9966	31.7486	4.5435	0.269375

正态分布理论参考: 1  $\sigma$  外 $\approx$ 31.73%, 2  $\sigma$  外 $\approx$ 4.55%, 3  $\sigma$  外 $\approx$ 0.27%

从公式与原理角度分析, 根据中心极限定理, 独立同分布随机变量的和的分布会趋近于正态分布; 而当原随机变量本身服从高斯(正态)分布时, 其和的分布会精确

服从正态分布。对于代码中 `randn` 生成的均值为 0、方差为 1 的高斯随机变量，若有  $M$  个这样的变量求和，根据正态分布的可加性，和的分布应为均值  $\mu_{\text{sum}} = M \times 0 = 0$ 、方差  $\sigma_{\text{sum}}^2 = M \times 1 = M$  的正态分布，即标准差  $\sigma_{\text{sum}} = \sqrt{M}$ 。从运行结果看，均值列（如  $M = 2$  时为 0.0007， $M = 16$  时为 -0.0028）因随机模拟存在微小波动，但整体接近理论均值 0；标准差列（ $M = 2$  时约 1.4156，接近  $\sqrt{2} \approx 1.4142$ ， $M = 16$  时约 3.9966，接近  $\sqrt{16} = 4$ ）与  $\sqrt{M}$  的理论值高度吻合，验证了和的方差为  $M$  的结论。从图形来看，四个子图分别对应  $M = 2, 4, 8, 16$ ，所有子图的分布均呈现正态分布的钟形曲线，说明  $M$  个高斯随机变量的和仍服从正态分布。随着  $M$  增大，横轴“求和结果”的范围逐渐扩展（因标准差  $\sqrt{M}$  增大，数据离散程度提升），同时峰值附近的频数分布更集中（符合正态分布“方差越大，分布范围越宽但形状仍为钟形”的特点）。此外， $1\sigma$ 、 $2\sigma$ 、 $3\sigma$  外的百分比统计（如  $M = 2$  时  $1\sigma$  外  $\approx 31.7165\%$ ，接近理论值  $31.73\%$ ； $2\sigma$  外  $\approx 4.5423\%$ ，接近  $4.55\%$ ； $3\sigma$  外  $\approx 0.272875\%$ ，接近  $0.27\%$ ），验证了正态分布的“ $3\sigma$  法则”——无论方差大小，变量落在均值  $\pm 1\sigma$ 、 $\pm 2\sigma$ 、 $\pm 3\sigma$  外的概率稳定在理论值附近，进一步说明和的分布严格服从正态分布，且模拟结果与理论高度一致，既体现了高斯分布的可加性，也验证了中心极限定理在原分布为正态时的精确性。

RAYLEIGH分布下随机数和的分布 (M = 2 4 8 16)



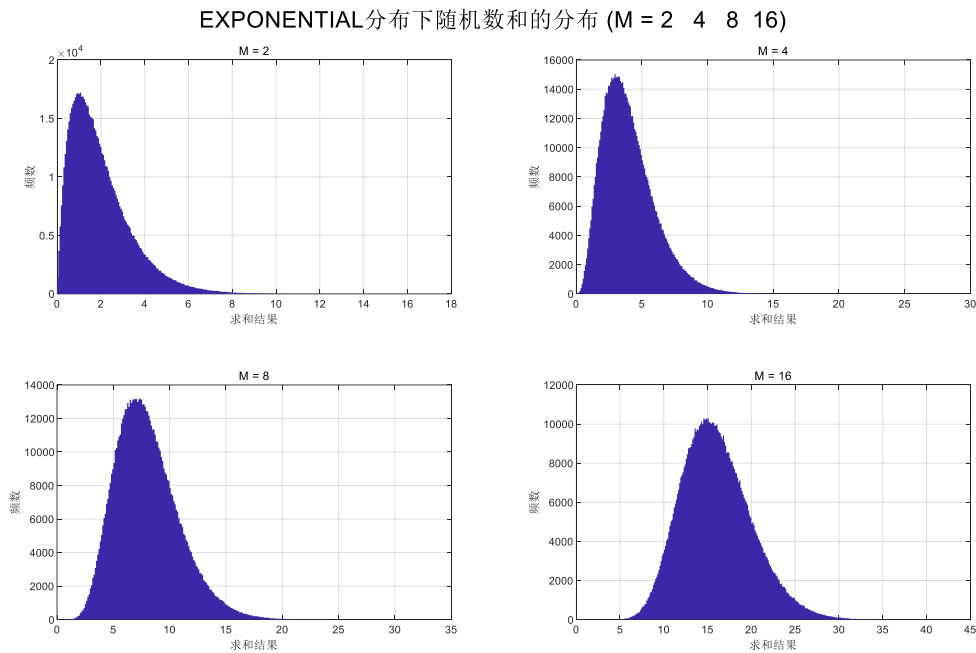
统计结果对比 (分布类型: rayleigh, M 值从左到右分别为 2 4 8 16):

M 值	均值	标准差	1 $\sigma$ 外(%)	2 $\sigma$ 外(%)	3 $\sigma$ 外(%)
2	2.5064	0.9258	32.0606	4.0572	0.430875
4	5.0126	1.3101	31.9206	4.3422	0.348250
8	10.0236	1.8529	31.8845	4.4188	0.299000
16	20.0549	2.6171	31.7462	4.4936	0.295375

正态分布理论参考: 1  $\sigma$  外 $\approx$ 31.73%, 2  $\sigma$  外 $\approx$ 4.55%, 3  $\sigma$  外 $\approx$ 0.27%

从原理和公式角度分析, 瑞利分布的概率密度函数为 $f(x) = \frac{x}{\sigma^2} e^{-x^2/(2\sigma^2)}$ (此处  $\sigma = 1$ ), 其均值为 $\sigma\sqrt{\pi/2} \approx 1.2533$ , 方差为 $\sigma^2(2 - \pi/2) \approx 0.4292$ 。根据中心极限定

理，当对  $M$  个独立同分布（且方差有限）的随机变量求和时，随着  $M$  增大，和的分布会趋近于正态分布。从图形看，当  $M = 2$  时，求和结果的直方图仍带有瑞利分布的正偏特性（左侧上升快、右侧拖尾）；随着  $M$  增加到 4、8、16，直方图逐渐变得对称，钟形曲线特征愈发明显，这是中心极限定理的直观体现。统计结果中，和的均值近似为  $M \times 1.2533$  (如  $M = 2$  时均值约 2.5064， $M = 16$  时约 20.0549)，符合“和的期望等于期望的和”的线性性质；和的标准差近似为  $\sqrt{M} \times 0.4292$  (如  $M = 2$  时约 0.926， $M = 16$  时约 2.617)，符合“独立变量和的方差等于方差的和”的可加性。此外， $1\sigma$ 、 $2\sigma$ 、 $3\sigma$  范围外的百分比随  $M$  增大逐渐逼近正态分布的理论值（ $1\sigma$  外  $\approx 31.73\%$ 、 $2\sigma$  外  $\approx 4.55\%$ 、 $3\sigma$  外  $\approx 0.27\%$ ）： $M = 2$  时  $1\sigma$  外 32.06% 已接近理论值， $M = 16$  时  $1\sigma$  外 31.7462%、 $2\sigma$  外 4.4936%、 $3\sigma$  外 0.295375% 几乎与理论一致。这表明，随着求和的瑞利变量个数  $M$  增加，和的分布愈发符合正态分布，完美验证了中心极限定理——大量独立随机变量的和的分布会收敛于正态分布，无论原分布的偏度、形状如何，求和后分布的对称性与正态特性都会逐渐增强。



统计结果对比 (分布类型: exponential,  $M$  值从左到右分别为 2    4    8    16):

-----



M 值	均值	标准差	1 σ 外(%)	2 σ 外(%)	3 σ 外(%)
-----	----	-----	----------	----------	----------

2	2.0011	1.4153	26.2663	4.6529	1.416875
4	4.0018	2.0035	29.3388	4.2447	1.036250
8	7.9956	2.8225	30.5876	4.1069	0.747000
16	16.0096	4.0108	31.1969	4.2589	0.531000

正态分布理论参考: 1 σ 外≈31.73%, 2 σ 外≈4.55%, 3 σ 外≈0.27%

从公式与原理角度分析: 指数分布(参数 $\lambda = 1$ )的概率密度为 $f(x) = e^{-x}(x \geq 0)$ , 若 $X_1, X_2, \dots, X_M$ 是独立同分布的指数分布随机变量, 则它们的和 $S_M = \sum_{i=1}^M X_i$ 服从爱尔朗(Erlang)分布, 其概率密度为 $f_{S_M}(x) = \frac{x^{M-1}e^{-x}}{(M-1)!}(x \geq 0)$ 。同时, 根据中心极限定理, 当 $M$ 足够大时, 独立同分布随机变量的和会趋近于正态分布(只要原分布有有限均值和方差)。观察图形: 当 $M = 2$ 时, 和的分布呈明显右偏态(因指数分布本身右偏, 两个指数分布之和的偏态仍显著); 随着 $M$ 增大( $M = 4, 8, 16$ ), 分布的偏态逐渐减弱, 峰形越来越对称, 逐渐向“钟形”(正态分布特征)靠拢。这直观体现了中心极限定理的趋势—— $M$ 越大, 和的分布越接近正态分布。结合统计结果: 指数分布的均值 $\mu = 1$ 、方差 $\sigma^2 = 1$ , 因此 $M$ 个指数分布随机变量和的理论均值为 $M \times 1 = M$ , 理论标准差为 $\sqrt{M \times 1} = \sqrt{M}$ 。运行结果中,  $M = 2$ 时均值 $\approx 2.0011$ 、标准差 $\approx 1.4153$ (接近 $\sqrt{2} \approx 1.4142$ );  $M = 4$ 时均值 $\approx 4.0018$ 、标准差 $\approx 2.0035$ (接近 $\sqrt{4} = 2$ );  $M = 8$ 时均值 $\approx 7.9956$ 、标准差 $\approx 2.8225$ (接近 $\sqrt{8} \approx 2.8284$ );  $M = 16$ 时均值 $\approx 16.0096$ 、标准差 $\approx 4.0108$ (接近 $\sqrt{16} = 4$ ), 均与理论值高度一致, 验证了和的均值、方差的可加性。再看“ $\sigma$ 范围外的百分比”: 正态分布理论上“1σ外 $\approx 31.73\%$ 、2σ外 $\approx 4.55\%$ 、3σ外 $\approx 0.27\%$ ”。当 $M = 2$ 时, 1σ外 $\approx 26.27\%$ (与正态分布差距大, 因分布偏态明显); 随着 $M$ 增大到4、8、16, 1σ外的百分比逐渐接近31.73%, 2σ外的百分比接近4.55%, 3σ外的百分比也呈趋近趋势(虽仍大于0.27%, 但逐渐减小)。这进一步说明:  $M$ 越大, 指数分布随机数的和越接近正态分布, 图形的峰形从右偏逐

渐对称、统计量也更贴合正态分布的理论特征，完美呼应了中心极限定理与爱尔朗分布向正态分布的过渡规律。

## 实验七 DFT+Noise

固定噪声

- (1) 手算，仅噪声 DFT
- (2) 信号加噪声 DFT，对比信号的 DFT 和噪声的 DFT
- (3) 噪声增大，再 N 增大--》噪声中提取信号信息的概念

### 1. 手算，仅噪声 DFT

Matlab generated 8 random numbers [0, 1]

0.4360 0.0259 0.5497 0.4353 0.4204 0.3303 0.2046 0.6193

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

$$= \sum_{n=0}^7 x[n]e^{-j\frac{\pi}{4}kn}$$

$$X(0) = \sum_{n=0}^7 x[n] = 3.0215$$

$$X(1) = \sum_{n=0}^7 x[n]e^{-j\frac{\pi}{4}n}$$

$$X(2) = \sum_{n=0}^7 x[n]e^{-j\frac{\pi}{2}n}$$

$$X(3) = \sum_{n=0}^7 x[n]e^{-j\frac{3}{4}\pi n}$$

$$X(4) = \sum_{n=0}^7 x[n]e^{-j\pi n}$$

$$X(5) = \sum_{n=0}^7 x[n]e^{-j\frac{5}{4}\pi n}$$

$$X(6) = \sum_{n=0}^7 x[n]e^{-j\frac{3}{2}\pi n}$$

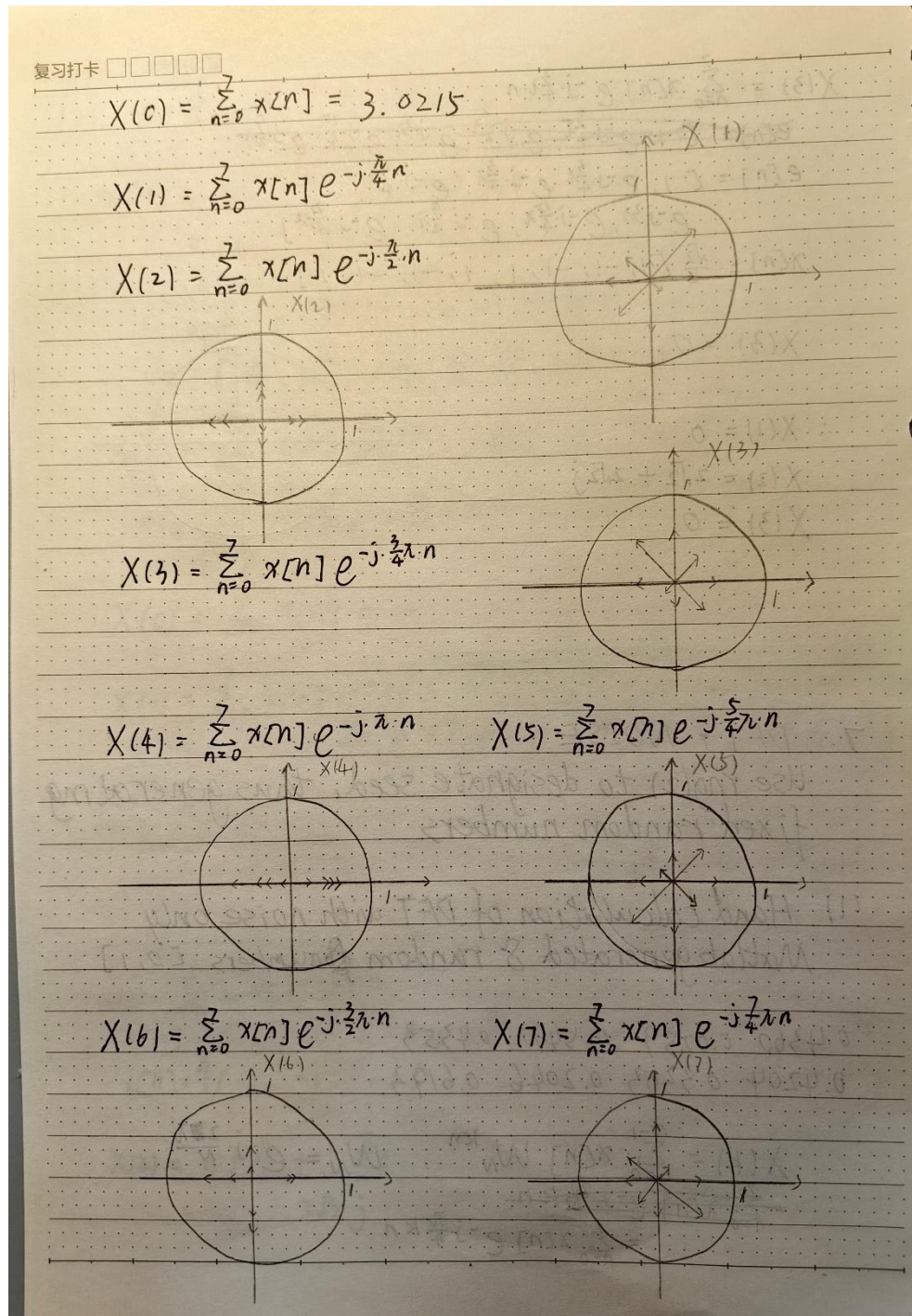
$$X(7) = \sum_{n=0}^7 x[n]e^{-j\frac{7}{4}\pi n}$$

$$X(1)=0.436+0.0183+0-0.308-0.4204-0.233+0+0.438+j \cdot (0-0.018-0.549-0.308+0+0.233+0.2046+0.438)$$

$$\approx -0.0695+0.0002j$$

$$X(2)=0.436-0.5497+0.4204-0.2046+j \cdot (-0.0259+0.4353-0.3303+0.6193)$$

$$=0.1021+0.6984j$$



## 2. 信号加噪声 DFT，对比信号的 DFT 和噪声的 DFT

% 通用 DFT 绘制程序（极坐标版 + 合并显示修正版）

clear; clc;

close all;

% ===== 定义是实数信号还是复数信号 =====

COMPLEX\_SIGNAL = 1;

REAL\_SIGNAL = 0;

% ===== 参数设置 =====

N = 8; % DFT 点数

Fre = 1; % 信号频率

phai = 0; % 初相位

highlight = [2,3,4]; % 要标红的点索引（MATLAB 下标从 1 开始）

Noise\_Amplitude = 1;

Signal\_Amplitude = 1;

% ===== 构造信号 =====

Signal\_Type = REAL\_SIGNAL; % REAL\_SIGNAL -> cos, COMPLEX\_SIGNAL -> exp

n = 0:N-1;

if Signal\_Type == COMPLEX\_SIGNAL

    x = Signal\_Amplitude\*exp(1j\*2\*pi\*Fre\*n/N + phai); % 复数信号

else

    x = Signal\_Amplitude\* cos(2\*pi\*Fre\*n/N + phai); % 实信号

end

%%%

rng('shuffle'); %使用当前时间作为种子值，产生不可预测的随机数序列；不调用 rng 也产生不同不可预测的随机数序列

rng(2); %指定种子，产生固定的随机数

noise = rand(1, N);

x = x + Noise\_Amplitude \* noise;

%%%

% ===== DFT 循环实现 =====

X\_loop = zeros(1,N);

```

for k = 0:N-1
    for nn = 0:N-1
        X_loop(k+1) = X_loop(k+1) + x(nn+1) * exp(-1j*2*pi/N*k*nn);
    end
end

% ===== 矩阵实现 =====
W = exp(-1j*2*pi/N*(0:N-1)'*(0:N-1)); % DFT 矩阵
X_matrix = (W * x.').'; % DFT 结果

% ===== IDFT 验证 =====
W_inv = exp(1j*2*pi*(0:N-1)'*(0:N-1)/N)/N;
I_check = W * W_inv; % 应该接近单位阵
I_check = round(I_check, 2); % 保留两位小数
disp('验证 W * W_inv 是否为单位矩阵 (保留两位小数):');
disp(I_check);

% ===== 打印 DFT 循环和矩阵结果 =====
disp('对比 DFT 循环 vs 矩阵结果:');
for k = 1:N
    if ismember( k , highlight )
        fprintf('--X[%d] 循环= %.2f%.2fj , 矩阵= %.2f%.2fj\n', ...
            k-1, real(X_loop(k)), imag(X_loop(k)), real(X_matrix(k)),
            imag(X_matrix(k)));
    else
        fprintf('X[%d] 循环= %.2f%.2fj , 矩阵= %.2f%.2fj\n', ...
            k-1, real(X_loop(k)), imag(X_loop(k)), real(X_matrix(k)),
            imag(X_matrix(k)));
    end
end

% ===== 修正: 两张图合并为同一窗口 (左右布局) =====
figure('Position', [100, 100, 1000, 500]); % 总窗口大小 (x,y,宽,高)

% ----- 左图: 极坐标复平面图 -----
% 直接创建极坐标 axes, 不使用 subplot
pax = polaraxes;
pax.Position = [0.05, 0.1, 0.4, 0.8]; % [左, 下, 宽, 高] 占比
hold(pax, 'on');

% 圆半径取最大幅值的 1.1 倍

```

```

R = max(abs([X_loop, X_matrix])) * 1.1;

% 绘制所有 DFT 点
offset_angle = pi/30; % 文字交替偏移, 防止重叠
for k = 1:N
    r = abs(X_loop(k)); % 模值 (极径)
    theta = angle(X_loop(k)); % 相位 (极角)

    % 区分普通点和重点
    if ismember(k, highlight)
        polarscatter(pax, theta, r, 80, 'r', 'filled'); % 红色大圆点
    else
        polarscatter(pax, theta, r, 60, 'b', 'filled'); % 蓝色小圆点
    end

    % 点的标注
    if ismember(k, highlight) && ~(abs(real(X_loop(k))) < 1e-10 &&
abs(imag(X_loop(k))) < 1e-10)
        txt = sprintf('X[%d]: %.2f%+.2fj', k-1, real(X_loop(k)),
imag(X_loop(k)));
    else
        txt = sprintf('X[%d]', k-1);
    end

    % 添加标注文字
    text(theta + offset_angle, r, txt, 'FontSize', 9, ...
        'HorizontalAlignment', 'left', 'Parent', pax);
    offset_angle = -offset_angle; % 交替正负偏移
end

title(pax, 'DFT 复平面 (极坐标)', 'FontSize', 12);

% ----- 右图: DFT 模值离散图 -----
% 创建普通坐标 axes
ax = axes;
ax.Position = [0.55, 0.1, 0.4, 0.8]; % [左, 下, 宽, 高] 占比
hold(ax, 'on');

X_mag = abs(X_loop); % 计算 DFT 模值

% 绘制离散信号图

```

```

stem(ax, 0:N-1, X_mag, 'filled', 'MarkerSize', 6, 'Color', [0.2, 0.6, 0.8]);
grid on;
xlabel(ax, 'k (频率点索引)', 'FontSize', 11);
ylabel(ax, '|X[k]| (DFT 模值)', 'FontSize', 11);
title(ax, 'DFT 模值随 k 的变化', 'FontSize', 12);
xlim(ax, [-0.5, N-0.5]);
ylim(ax, [0, max(X_mag)*1.2]);

% ----- 总标题 -----
sgtitle(sprintf('DFT 结果可视化 (N=%d, 频率 Fre=%d, 初相 phi=%.2f rad)', N, Fre,
    phai), ...
    'FontSize', 14, 'FontWeight', 'bold');

```

## 生成基础信号

Signal\_Type = REAL\_SIGNAL; % 选择实信号（可改为 COMPLEX\_SIGNAL 切换复数信号）

n = 0:N-1; % 离散时间索引（0~N-1）

if Signal\_Type == COMPLEX\_SIGNAL

% 复数信号：复指数形式  $x(n) = A \cdot e^{j(2\pi \text{Fre} \cdot n/N + \phi)}$

x = Signal\_Amplitude \* exp(1j \* 2 \* pi \* Fre \* n / N + phai);

else

% 实信号：余弦形式  $x(n) = A \cdot \cos(2\pi \text{Fre} \cdot n/N + \phi)$

x = Signal\_Amplitude \* cos(2 \* pi \* Fre \* n / N + phai);

end

- 信号基于离散时间索引 n 生成，频率 Fre 对应离散域的频率（例如 Fre=1 表示每 N 个点完成 1 个周期）。
- 复数信号更贴近 DFT 的理论模型（DFT 本质处理复数序列），实信号是复数信号的特例（共轭对称）。

## 添加噪声

rng(2); % 固定随机数种子（保证结果可重复）

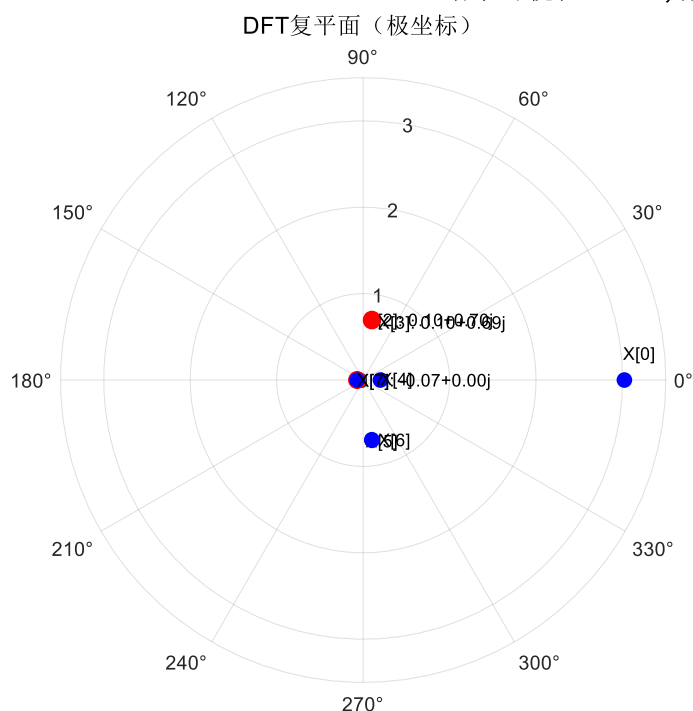
noise = rand(1, N); % 生成[0,1)区间的均匀噪声

x = x + Noise\_Amplitude \* noise; % 信号叠加噪声

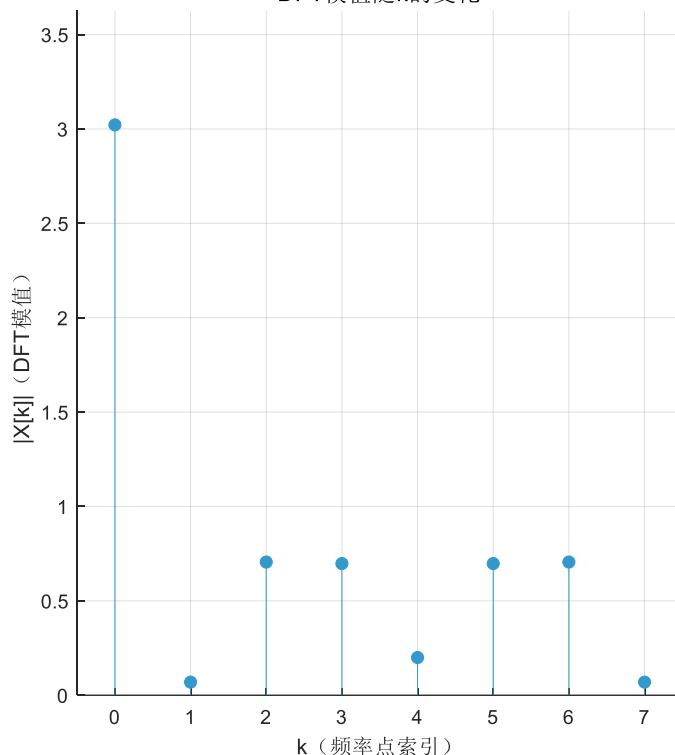
- 噪声用于模拟实际信号中的干扰，`rng(2)`固定种子使每次运行噪声相同，便于对比结果。

### (1) 仅噪声 DFT

**DFT结果可视化 (N=8, 频率Fre=1, 初相phi=0.00 rad)**



**DFT模值随k的变化**



```
Noise_Amplitude = 1;
Signal_Amplitude = 0;
```

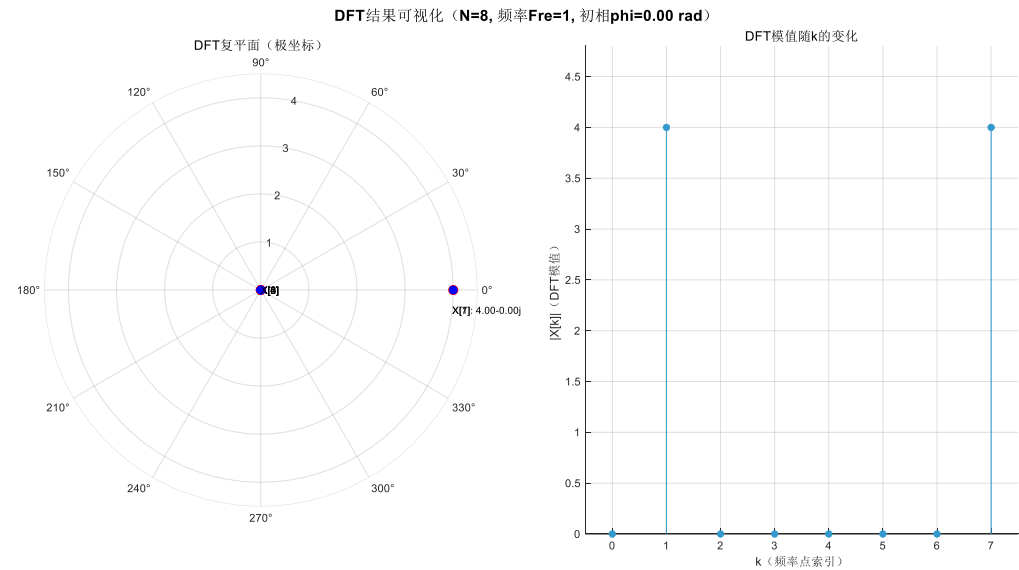
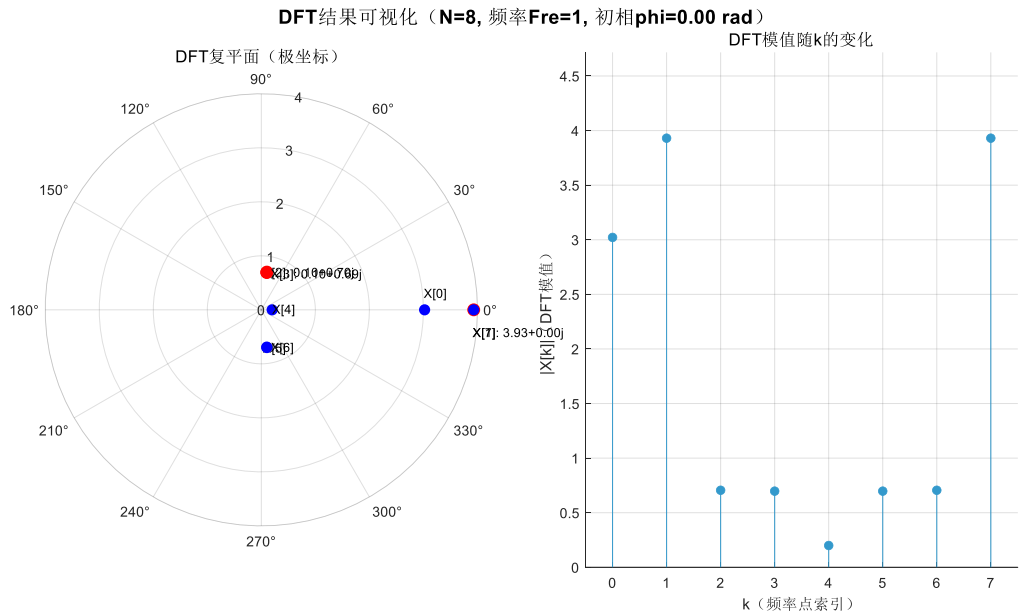
当仅存在噪声时，噪声是通过 `rand` 生成的实值随机序列。根据离散傅里叶变换 (DFT) 的定义  $X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}$ ，这里的  $x[n]$  为随机噪声序列。由于噪声的随机性，每个频率点  $k$  对应的傅里叶分量  $X[k]$  是随机的复数值：在极坐标复平面图中，各  $X[k]$  的复向量会随机分布在复平面上，没有因确定性单频信号存在而产生的“集中指向某一相位、聚集在特定模值半径”的情况；从白噪声的功率谱特性（平坦功率谱，即各频率分量的能量近似均匀分布）来看，其 DFT 模值  $|X[k]|$  在各个频率点  $k$  上应呈现随机且相对均匀的分布，不会出现某几个  $k$  对应的模值显著高于其他点的“峰值”情况。反映在图形中，极坐标图里的复点随机散布，无明显聚集趋势；DFT 模



值随  $k$  变化的图中，各  $k$  对应的模值分散且无主导性的高峰，这与随机噪声的频谱特性（各频率分量随机且能量均匀分布）在离散傅里叶变换下的表现相契合。

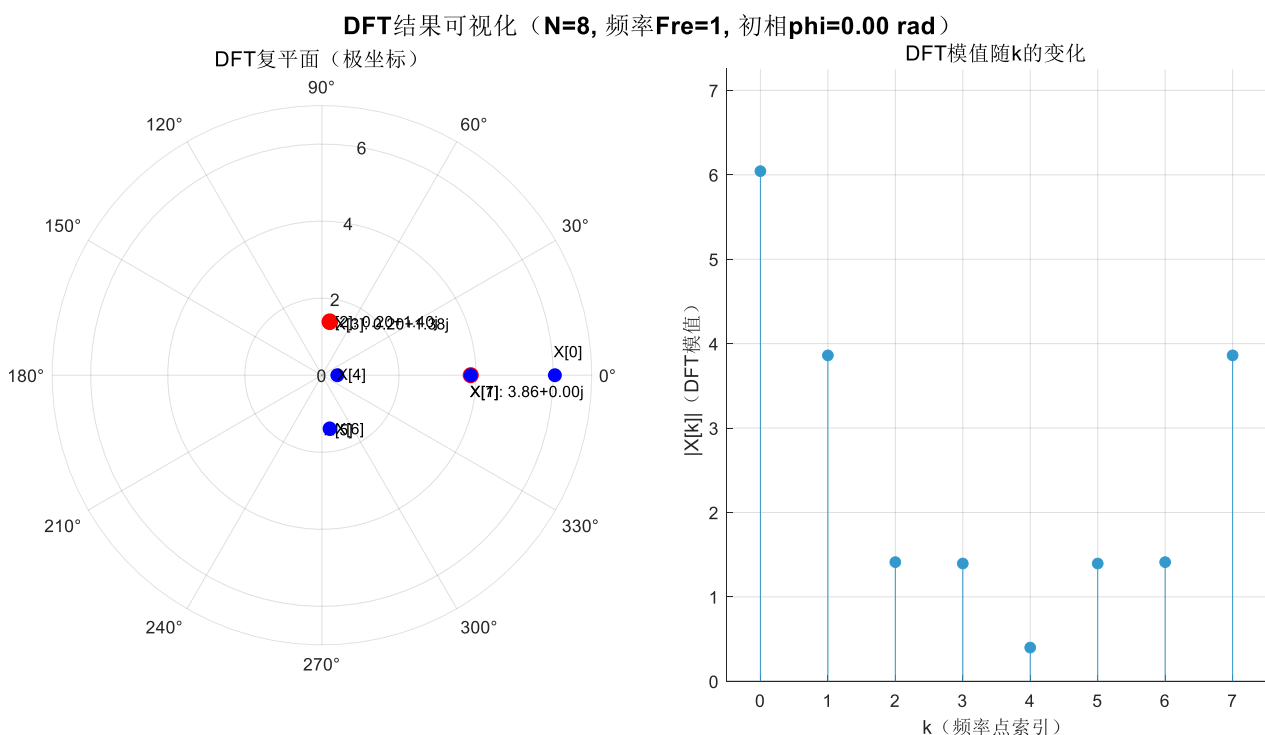
(2) 信号+噪声 DFT

```
% ===== 参数设置 =====
N = 8; % DFT 点数
Fre = 1; % 信号频率
phai = 0; % 初相位
highlight = [2,3,4]; % 要标红的点索引（MATLAB 下标从 1 开始）
Noise_Amplitude = 1;
Signal_Amplitude = 1;
```

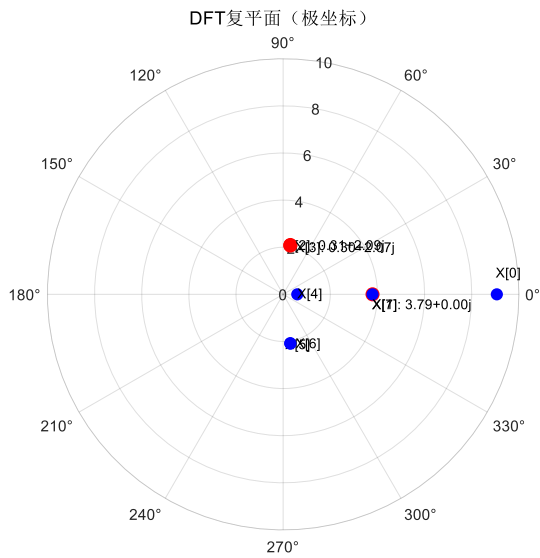


从 DFT 的定义  $X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$  ( $k = 0, 1, \dots, N-1$ ) 来看, 当信号为纯实余弦信号  $x[n] = A\cos(2\pi Fn/N + \phi)$  时, 由于余弦信号可分解为  $\frac{A}{2}e^{j(2\pi Fn/N + \phi)} + \frac{A}{2}e^{-j(2\pi Fn/N + \phi)}$  根据 DFT 对复指数信号的频率采样特性, 这两个复指数会分别在  $k = F$  和  $k = N - F$  (因  $N = 8$ 、 $F = 1$ , 故  $k = 1$  和  $k = 7$ ) 处产生幅度为  $\frac{AN}{2}$  的谱线, 其他频率点  $k$  因无信号分量, DFT 模值应接近 0。因此仅信号的图形中, 极坐标下  $k = 1$  和  $k = 7$  的点模值大且相位对称, 其他点接近原点; 离散模值图中仅  $k = 1$  和  $k = 7$  有高柱形, 其余几乎为 0, 体现了纯信号频谱的稀疏性与频率集中性。而当加入噪声后, 噪声是随机序列 (此处为均匀分布随机数), 其 DFT 会因随机性在所有频率点  $k$  产生无规律的频谱分量 (因为噪声是宽带的, 能量分散在各频率)。此时信号+噪声的 DFT 中,  $k = 1$  和  $k = 7$  的主谱线仍存在 (信号能量主导这些点), 但其他频率点 (如  $k = 2, 3, 4, 5, 6$ ) 的模值不再接近 0: 极坐标下这些点远离原点, 离散模值图中对应柱形出现高度, 这是噪声将能量分散到各频率点的结果, 体现了噪声对信号频谱的干扰—使非目标频率点的频谱分量上升, 频谱从“干净稀疏”变为“带有杂散分量”, 但主频率分量仍可通过高模值谱线识别。 $k = 0$ : 直流分量 (对应频率  $f = 0$ )。该点的幅度由信号的直流成分 (余弦信号的直流分量为 0, 但噪声有直流叠加) 和噪声的直流叠加共同决定, 因此也会有一定突出。

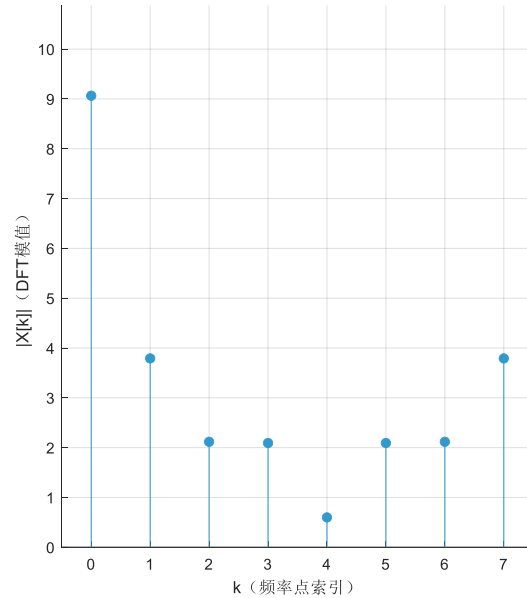
### Noise\_Amplitude = 2, 3, 4 时图像变化



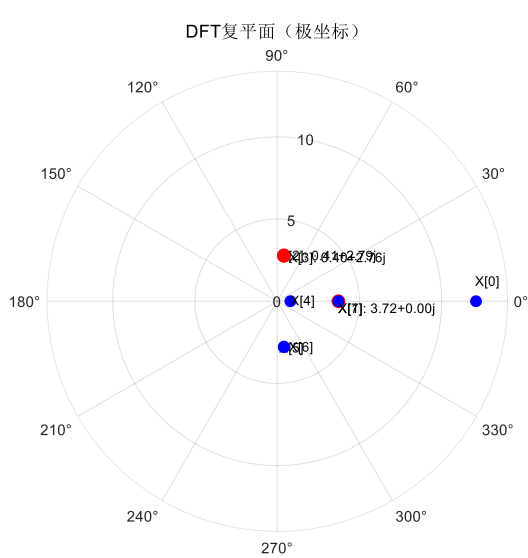
DFT结果可视化 (N=8, 频率Fre=1, 初相phi=0.00 rad)



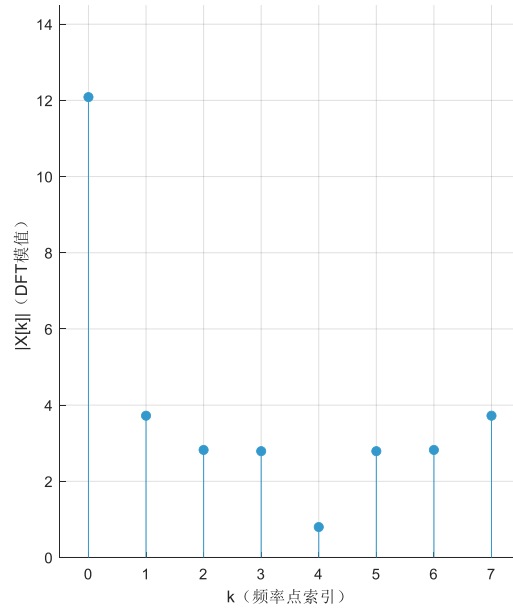
DFT模值随k的变化



DFT结果可视化 (N=8, 频率Fre=1, 初相phi=0.00 rad)



DFT模值随k的变化



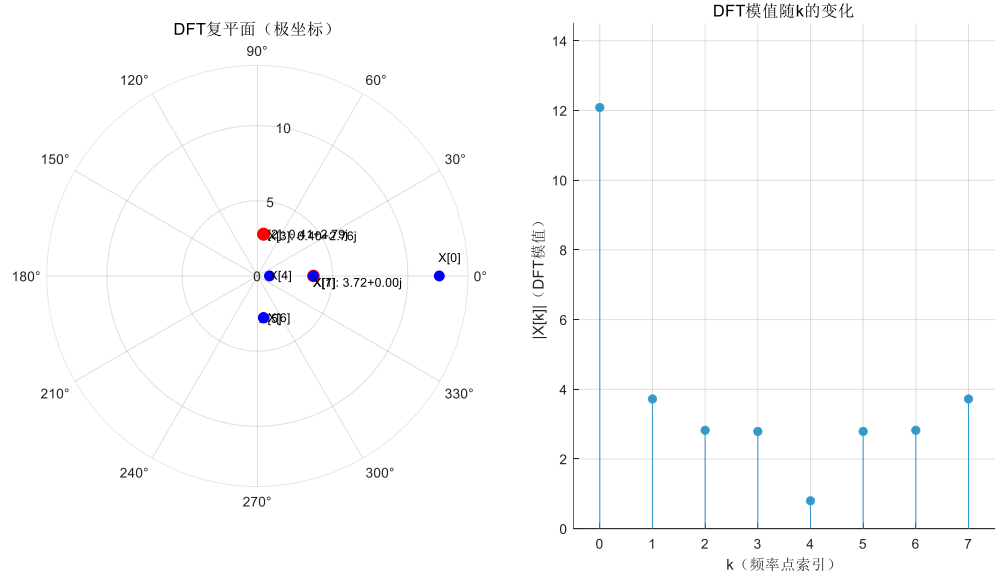
可以看到，随着噪声幅值的增加，经过傅里叶变换后，信号的频谱越来越难以辨认原本应该突出的  $k=1$  和  $k=7$  的谱线，随着噪声影响的加大，其他  $k$  值对应的谱线逐渐突出。

### 3. 噪声增大，再 N 增大→噪声中提取信号信息的概念

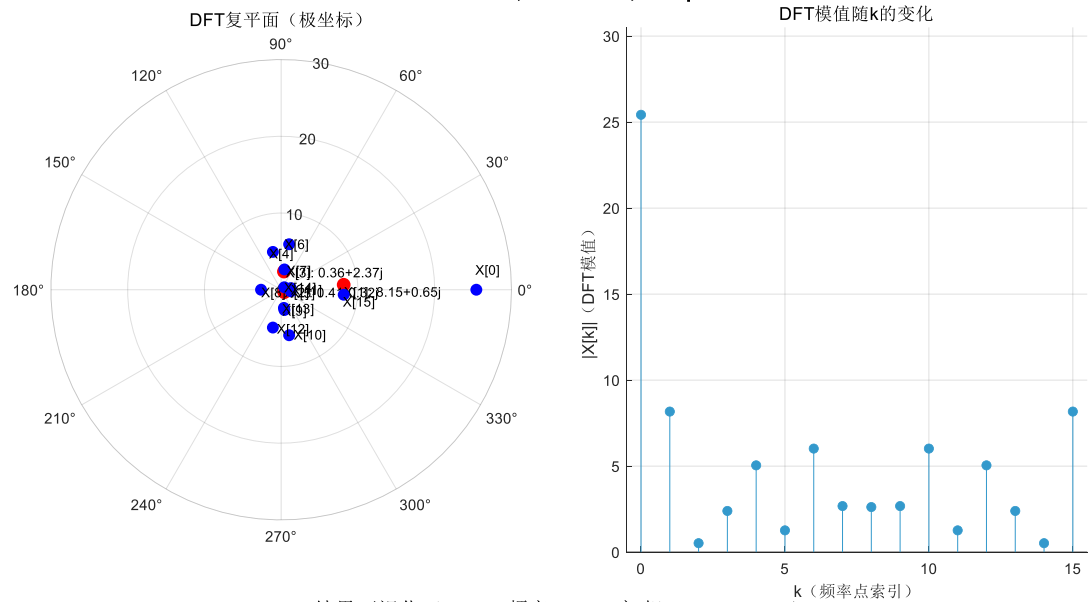
固定 Noise\_Amplitude = 4, 增大 N 值分别为 8, 16, 32, 64, 128, 观察谱线变

化。

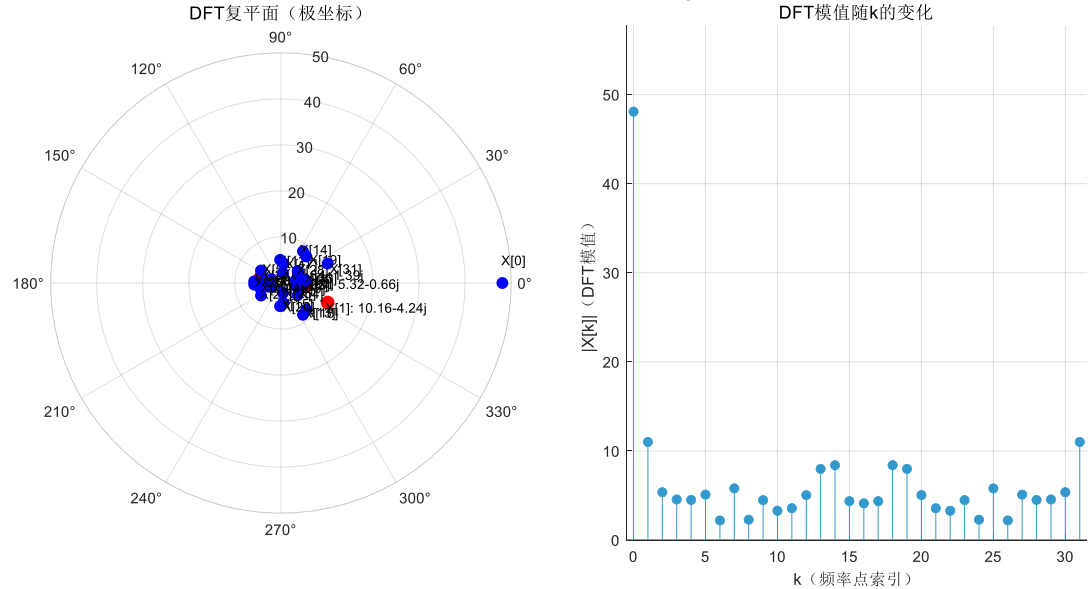
DFT结果可视化（N=8, 频率Fre=1, 初相phi=0.00 rad）

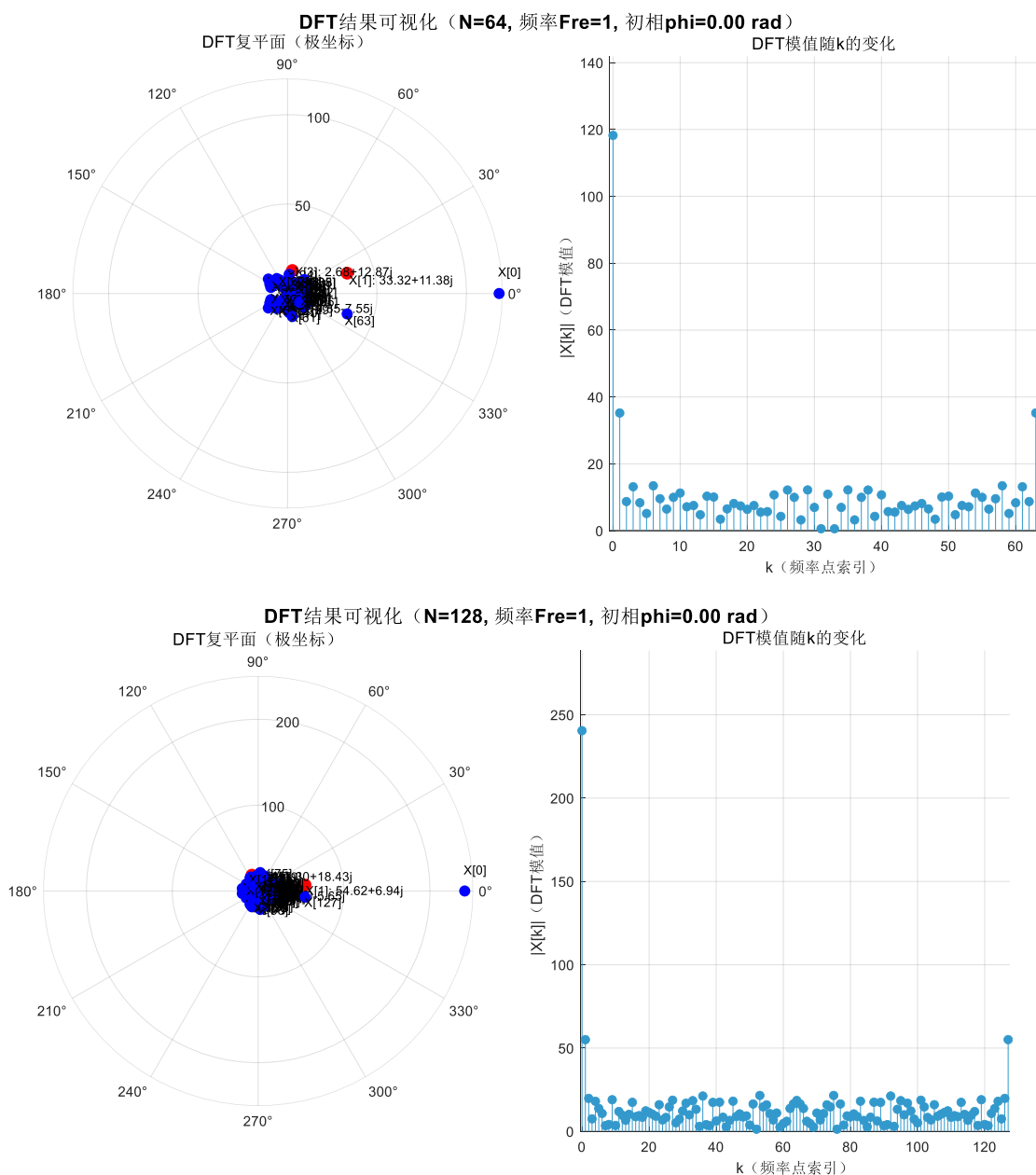


DFT结果可视化（N=16, 频率Fre=1, 初相phi=0.00 rad）



DFT结果可视化（N=32, 频率Fre=1, 初相phi=0.00 rad）





从 DFT 的公式 $X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$ 分析, 这里的 $x[n]$ 由单频信号( $\text{Fre} = 1$ , 对应频域索引 $k = 1$ )和固定幅度的随机噪声组成。对于单频信号部分 (若为复指数信号 $x_{\text{sig}}[n] = Ae^{j2\pi \cdot 1 \cdot n/N + \phi}$ ), 当计算 $k = 1$ 处的 DFT 时, 各 $n$ 对应的复指数 $e^{-j2\pi \cdot 1 \cdot n/N}$ 与信号本身的复指数 $e^{j2\pi \cdot 1 \cdot n/N}$ 相乘后, 相位项抵消, 仅剩常数  $A$ , 因此 $X[1] = \sum_{n=0}^{N-1} A e^{j\phi} = ANe^{j\phi}$ , 即信号的 DFT 模值随  $N$  线性增长 (幅度为  $AN$ )。而噪声部分 $x_{\text{noise}}[n]$ 是随机的, 各 $n$ 对应的噪声样本独立, 其 DFT 是“不相干累加”——每

个噪声样本乘以复指数后的相位随机，叠加时幅度遵循“随机游走”规律，模值增长仅与 $\sqrt{N}$ 同阶。当固定噪声幅度 `Noise_Amplitude= 4`并增大  $N$ （从 8 到 128）时，图形呈现明显规律：信号对应的 $k = 1$ 处，DFT 模值随  $N$  线性升（如 $N = 128$ 时，模值接近 $1 \times 128 = 128$ 的理论值，因噪声存在略有波动，但整体趋势显著）；而噪声引起的其他  $k$  处 DFT 模值，因“不相干累加”仅随 $\sqrt{N}$ 缓慢增长，与 $k = 1$ 处的信号模值差距越来越大。

- **信号：**能量高度**集中**在特定频率点（如 $k=2$ 、 $k=126$ ）—— 因为余弦信号是“单频”的，其 DFT 仅在对应频率点有显著峰值。
- **噪声：**能量**均匀分散**在所有  $N$  个频率点（代码中噪声是随机序列 `rand(1,N)`，近似白噪声）—— 每个频率点的噪声幅度随机且平均能量低。

反映在极坐标图中， $k = 1$ 附近的标注点（红色）从 $N = 8$ 时的“相对突出”变为 $N = 128$ 时的“极其突出”，周围噪声点的模值相对可忽略；模值图中 $k = 1$ 的谱线也从 $N = 8$ 时与噪声谱线“差距较小”，变为 $N = 128$ 时“远远高于所有噪声谱线”。这一现象的本质是：DFT 对单频信号的“相干累加”使其频域幅度随  $N$  线性增强，而噪声的“不相干累加”使其频域幅度增长远慢于信号，因此随着  $N$  增大，信号在频域的“信噪比”急剧提升，越来越容易从噪声中被识别提取，体现了“增加数据长度可利用 DFT 的相干特性强化信号、抑制随机噪声”的原理。

## 谱线位置的解释

在这里，采样频率 $f_s$ 是隐含设定的，默认与 DFT 点数  $N$  相等（即 $f_s = N = 128$ ）。以下是关键推导：1.信号频率的表示形式代码中构造实信号的公式为：

$$x(n) = A \cos\left(\frac{2\pi \cdot \text{Frec}n}{N} + \phi\right)$$

其中， $n$  是采样点索引( $0 \leq n < N$ )

我们通常研究“归一化频率”（即把采样频率 $f_s$ 视为“1 个单位”）。此时：采样频率 $f_s$ 对应“1 个单位频率”，角频率的“归一化形式”则把 $2\pi f_s$ 视为“ $2\pi$  弧度（1 个单位角频率）”。结合 DFT 频率索引的推导已知 DFT 中频率索引  $k$  与“归一化频率”的关系为：归一化频率 $= \frac{k}{N}$ （因为 DFT 有  $N$  个点，索引  $k$  从 0 到  $N-1$ ，覆盖“0 到

1 个单位频率”)。而“归一化角频率”是“归一化频率”的 $2\pi$ 倍(对应 $\omega = 2\pi f$ 的归一化形式), 因此: 归一化角频率 $= 2\pi \cdot \frac{k}{N} = \frac{2\pi k}{N}$

从 DFT 频率与索引的关系 $f = \frac{k \cdot f_s}{N}$ ( $k$  为频率索引) 来看, 信号的角频率可写为 $\omega = \frac{2\pi k}{N}$ 。对比信号公式中的角频率 $\omega = \frac{2\pi \cdot \text{Freq}}{N}$ , 可得:

$k = \text{Fre}$

谱线位置的解释: 代码中 $\text{Fre} = 2$ , 因此正频率分量的索引 $k_c = 2$ 。由于信号是实信号, 其 DFT 满足共轭对称性:  $X[k] = X^*[N - k]$  (\*表示共轭)。因此, 负频率分量会在 $k = N - k_c = 128 - 2 = 126$ 处体现, 最终谱线会突出在 $k = 2$ 和 $k = 126$ 处。总结采样频率 $f_s$ : 隐含为 $N = 128$  (默认 $f_s = N$ , 属于“归一化频率”的简化表示)。信号频率  $f$ : 若按物理频率理解,  $f = \frac{\text{Freq} \cdot f_s}{N} = \frac{2 \times 128}{128} = 2\text{Hz}$  (与  $\text{Fre}$  数值一致, 因 $f_s = N$ )。频率索引 $k_c$ : 由  $\text{Fre}$  直接得到 $k_c = 2$ , 再由实信号共轭对称性推出 $k = 126$ 处的对称谱线。

## 六、实验收获和体会

通过本次高级数字信号处理 DFT 实验, 我实现了从理论公式到工程实践的深度跨越, 对离散傅里叶变换的本质、信号时域与频域的对应关系, 以及实际场景中的信号分析方法有了远超课本的直观理解。最初接触 DFT 时, 我仅将其视为“时域转频域的数学工具”, 对公式中旋转因子的物理意义、不同信号类型对 DFT 结果的影响缺乏具象认知, 但在手动计算三类信号(无初相实数、含  $45^\circ$  初相实数、复数)的 DFT 过程中, 这种模糊认知被彻底打破。以实数信号  $N=8$ 、 $\text{Fre}=1$  的计算为例, 我逐点推导  $n$  从 0 到 7 时  $x[n]$  与旋转因子  $e^{-j\pi kn/4}$  的乘积, 发现旋转因子在复平面上沿单位圆的旋转规律—— $n$  每增加 1, 相位便累积  $-\pi/4$ ,  $n=1$  时指向第四象限,  $n=3$  时指向第三象限, 而  $n=5$  与  $n=1$ 、 $n=7$  与  $n=3$  的复数值恰好关于原点对称。这种几何直观让我突然理解为何求和时部分项会相互抵消: 比如计算  $X[2]$  时,  $x[n]$  与  $e^{-j\pi n/2}$  逐项相乘后, 实部与虚部的正负贡献完全抵消, 最终结果为 0, 这也印证了“单频实信号仅在  $k=\text{Fre}$  和  $k=N-\text{Fre}$  处有能量集中”的理论, 让抽象的“正交性”

不再是课本上的名词，而是可通过计算验证的具体规律。

在 MATLAB 实现 DFT 的过程中，我进一步体会到“理论推导”与“工程实现”的互补性。实验中采用循环迭代和矩阵运算两种方法计算 DFT，循环法虽时间复杂度为  $O(N^2)$ ，但逐次累加的过程完全贴合 DFT 的定义，让我清晰看到每个频域点  $k$  是如何通过时域序列  $x[n]$  与对应旋转因子向量的点乘得到；而矩阵法则通过构造  $N \times N$  的 DFT 矩阵  $W$ ，将离散傅里叶变换转化为矩阵乘法，不仅代码更简洁，还让我直观感受到 DFT 矩阵的正交性——当验证  $W$  与 IDFT 矩阵  $W_{\text{inv}}$  的乘积为单位矩阵时，屏幕上显示的对角元素为 1、其余元素接近 0 的矩阵，彻底证实了“DFT 与 IDFT 互为逆变换”的理论，也让我明白为何可通过 IDFT 从频域准确重建时域信号。更重要的是，通过程序验证手算结果时，当看到循环法与矩阵法得到的  $X[0]$ 、 $X[1]$ 、 $X[2]$  与手动计算值完全一致（如无初相实数信号  $X[1]=4.00+0.00j$ ），我不仅确认了手算的正确性，更对 DFT 的数学严谨性有了更强的信任感，这种“理论推导-手动验证-软件复现”的闭环，让我掌握了信号处理实验的核心逻辑。

频谱泄露实验让我意识到了理想情况与实际场景的差异。当信号频率  $\text{Fre}=1$ （为频率分辨率  $1/N$  的整数倍）时，DFT 模值图仅在  $k=1$  和  $k=7$  处出现尖锐峰值，其他频率点模值接近 0，符合“单频信号频谱稀疏”的理想模型；但当  $\text{Fre}=1.2$ （非整数倍）时，原本集中的能量扩散到多个频率点， $k=0、1、2、6、7$  处均出现非零模值，这种“频谱泄露”现象起初让我困惑，直到结合 DFT 频率分辨率的定义（ $f_s/N$ ）才明白：当信号不满足“整周期截断”时，其频域能量会因频率与离散栅格不匹配而扩散。这一发现让我意识到，实际信号采样中参数选择的重要性——若要避免频谱泄露，需确保采样点数  $N$  与信号频率  $\text{Fre}$  满足“ $\text{Fre}$  为  $1/N$  整数倍”，或通过加窗函数抑制泄露，这种从实验现象反推工程解决方案的思维，是课本理论无法直接给予的。

中心极限定理实验则为我理解噪声的统计特性提供了关键支撑。通过对均匀、高斯、瑞利等不同分布的随机数求和，我观察到：无论原始分布是否对称，随着求和项数  $M$  的增加，求和结果的直方图都会逐渐趋近于钟形的正态分布——比如均匀分布在  $M=2$  时呈三角分布， $M=16$  时已接近标准正态分布；指数分布从  $M=2$  时的右偏态，逐渐变为  $M=16$  时的对称分布。同时，统计结果中“ $1\sigma$  外百分比”从  $M=2$  时的 26.27%（指数分布）逐渐逼近正态分布理论值 31.73%，这让我不仅巩固了概率统计知识，



更理解了“噪声为何常被近似为正态分布”——实际场景中的噪声多为多种随机干扰的叠加，符合中心极限定理的适用条件。这种认知为后续“噪声下的 DFT 分析”打下基础，当我在信号中加入随机噪声后，能快速意识到噪声的频域分布是随机且均匀的，而信号的能量集中特性可通过 DFT 的“相干累加”与噪声的“不相干累加”区分。

噪声下的 DFT 实验进一步深化了我对信号提取的理解。当仅存在噪声时，DFT 模值在各频率点随机分布，无明显峰值；而当信号叠加噪声后，尽管非目标频率点因噪声出现杂散分量，但信号对应的  $k=Fre$  处仍能观察到相对峰值。更关键的是，当固定噪声幅度、增大 DFT 点数  $N$  时，信号对应的频域模值随  $N$  线性增长（因相干累加），而噪声模值仅随  $\sqrt{N}$  缓慢增长（因不相干累加），两者差距逐渐扩大—— $N=128$  时，信号峰值已远高于噪声分量，清晰可辨。这一现象让我掌握了“从噪声中提取微弱信号”的核心方法：通过增加数据长度利用 DFT 的相干特性强化信号，同时抑制随机噪声，这种方法在雷达、通信等实际系统中具有重要应用，也让我明白“理论参数调整对工程性能的影响”并非抽象概念，而是可通过实验量化验证的规律。

整个实验过程中，我也深刻体会到细节严谨的重要性。比如 MATLAB 中数组索引从 1 开始，而 DFT 理论中  $k$  和  $n$  从 0 开始，初期因混淆两者导致程序结果与手算偏差，直到调整索引对应关系（ $k+1$ 、 $nn+1$ ）才解决问题；手动计算旋转因子相位时，因忽略复指数的周期性（如  $e^{-j5\pi/2}=e^{-j\pi/2}$ ）导致求和错误，反复检查后才发现周期性对简化计算的关键作用。这些细节让我明白，信号处理实验不仅需要理论支撑，更需要细致的操作和严谨的验证，任何微小的疏忽都可能导致结果偏差。

总的来说，本次 DFT 实验不仅让我熟练掌握了 DFT 的计算方法、实现技巧及相关性质，更培养了我“从理论到实践、从现象到本质”的工程思维。无论是时域与频域的对应关系、频谱泄露的抑制、噪声的统计特性，还是信号提取的方法，都为后续学习 FFT、滤波器设计、调制解调等内容奠定了坚实基础。更重要的是，通过亲手完成“推导-计算-编程-验证-分析”的完整流程，我真正理解了数字信号处理的核心逻辑——用数学工具解决实际问题，用实验验证理论猜想，这种能力将成为我后续学习和工作中的重要财富。