### 信号与系统实验报告

名 称： 快速傅里叶变换算法探究及应用

学 院： 计软智学院

专 业： 人工智能

学 号： 58121124

姓 名： 张博彦

日期： 2023 年 05 月 09 日

1. 实验目的
2. 加深对快速傅里叶变换的理解。
3. 熟悉并掌握按时间抽取FFT算法的程序编制。
4. 了解应用FFT进行信号分析中可能出现的问题，如混淆、泄露等，以便在实际应用中正确应用FFT。
5. 实验任务
6. 完成实验内容全部题目，分析解决调试代码过程中出现的问题。
7. 认真完成本次实验小结，思考快速傅里叶变换的原理和算法及其应用。
8. 主要设备、软件平台
9. 硬件：计算机
10. 软件：Matlab
11. 实验内容
12. 参照“按时间抽取法FFT-基2”算法结构，编写相应的FFT程序*myFFT*()。
13. 用所编写的myFFT()分析信号



* + 1. 信号频率，采样点数，采样间隔
    2. 信号频率，采样点数，采样间隔
    3. 信号频率，采样点数，采样间隔
    4. 信号频率，采样点数，采样间隔
    5. 将信号④后补全32个0，完成64点FFT

要求：

记录各种情况下的X(k)值，绘制频谱图并对结果分析讨论，说明参数的变化对信号频谱产生的影响；频谱只需绘制幅度频谱，归一化处理；

程序需提供人机交互模式(控制台/图形窗口均可)；提供是否补零输入选项；提供参数输入功能；

打印myFFT()源程序，标注相关代码注释。

1. 实验小结

代码实现：

根据具体输入，得到输出信号

function original = originalArray(f, N, T)

original = [];

for i = 1:N

original = [original, sin(2 \* pi \* f \* i \* T)];

end

n = input("是否需要进行补0操作？输入0表示不补0，输入大于0的数表示补多少位:");

original = [original, zeros(1, n)];

end

快速傅里叶计算实现

function X = myFFT(x)

M = log2(length(x));

N = length(x);

% 求数列的倒序（此处使用雷德算法实现）

change = bin2dec(fliplr(dec2bin([1:N] - 1))) + 1;

X = x(change);

% 进行蝶形运算

for s = 1:M

Nk = 2^s;

% 计算旋转因子

n = 1; % 初始为WN^0,所以取值为1

WN = exp(-(1i) \* 2 \* pi / Nk);

for j = 1:(Nk / 2)

for k = j:Nk:N

kN = k + Nk / 2;

temp = X(kN) \* n;

X(kN) = X(k) - temp;

X(k) = X(k) + temp;

end

n = n \* WN;

end

end

end

脚本输出：

f = input("请输入频率：");

N = input("请输入采样点数：");

T = input("请输入采样间隔：");

temp = originalArray(f, N, T);

X = myFFT(temp);

disp(X);

% 验证和matlab自带的fft算法结果是否相同

% disp(fft(temp));

k = 0:(length(temp) - 1);

array = zeros(1,length(X));

max = 0;% 用于归一化

for i = 1:length(X)

array(i) = abs(X(i));

if max < array(i)

max = array(i);

end

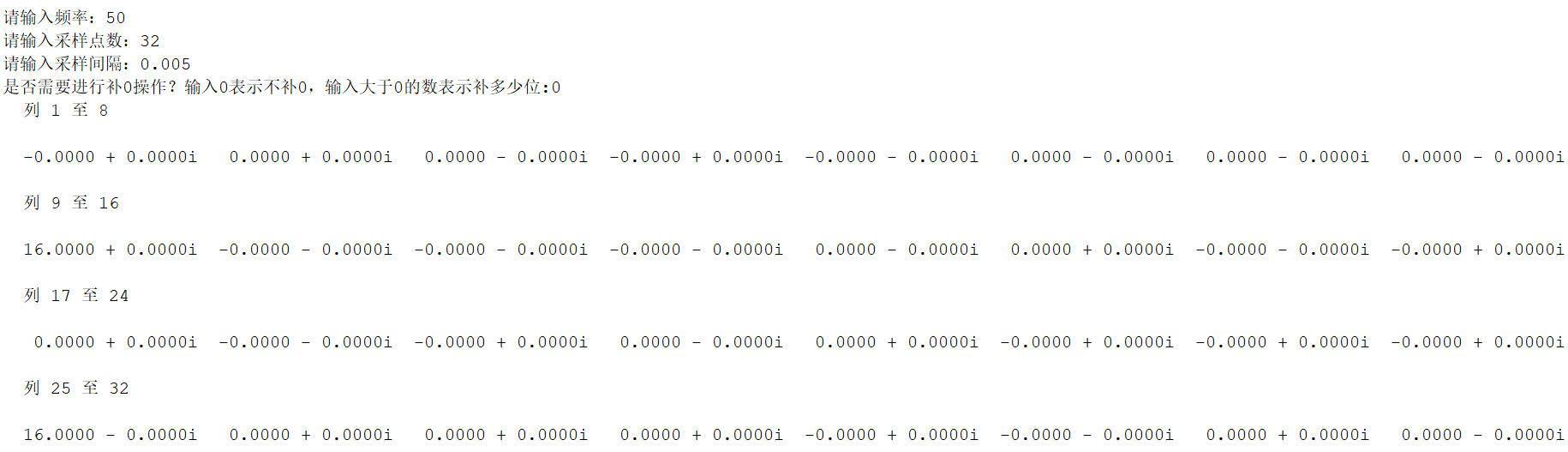
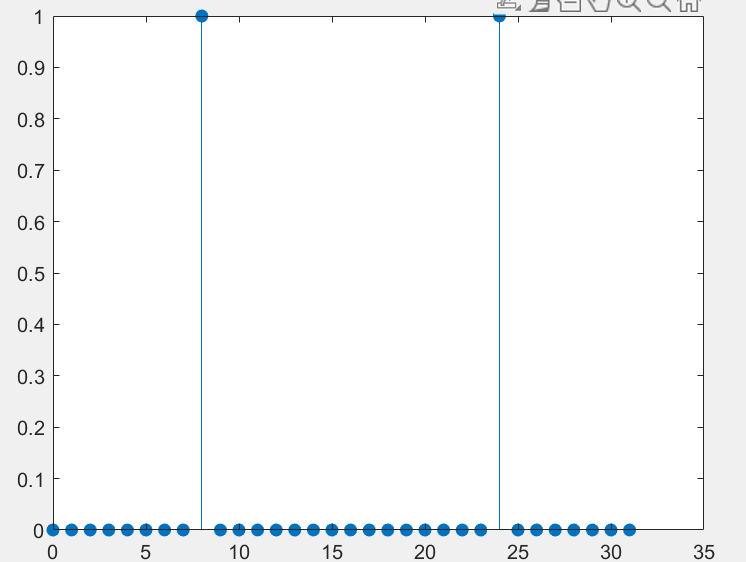
end

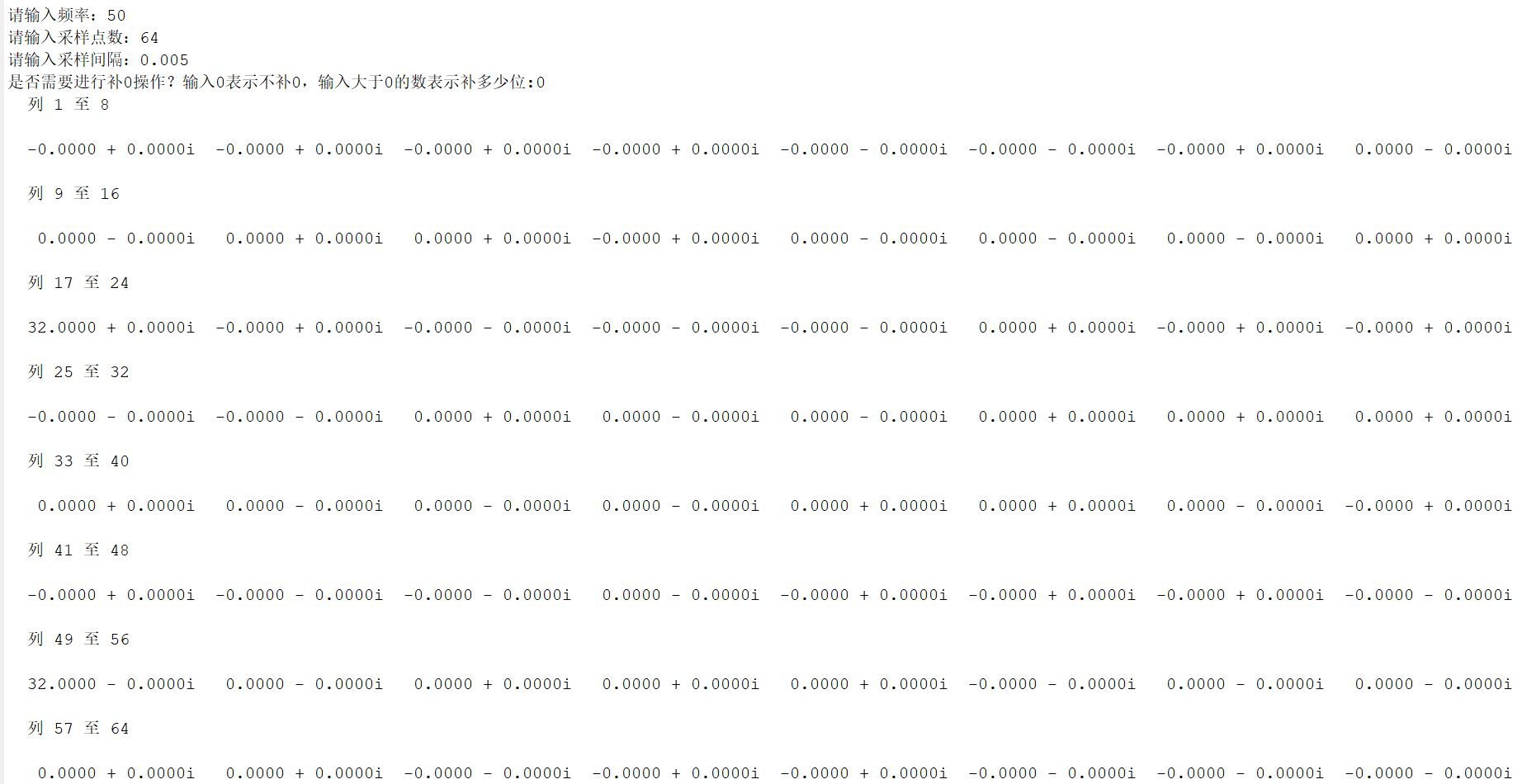
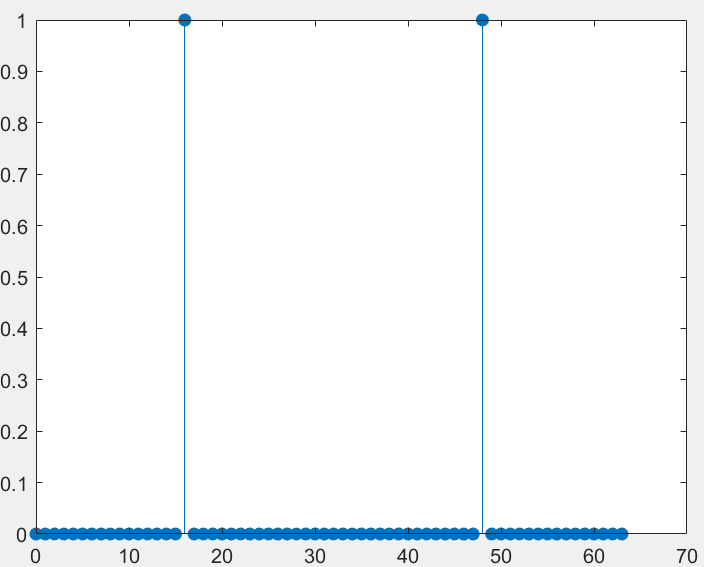
for i = 1:length(temp)

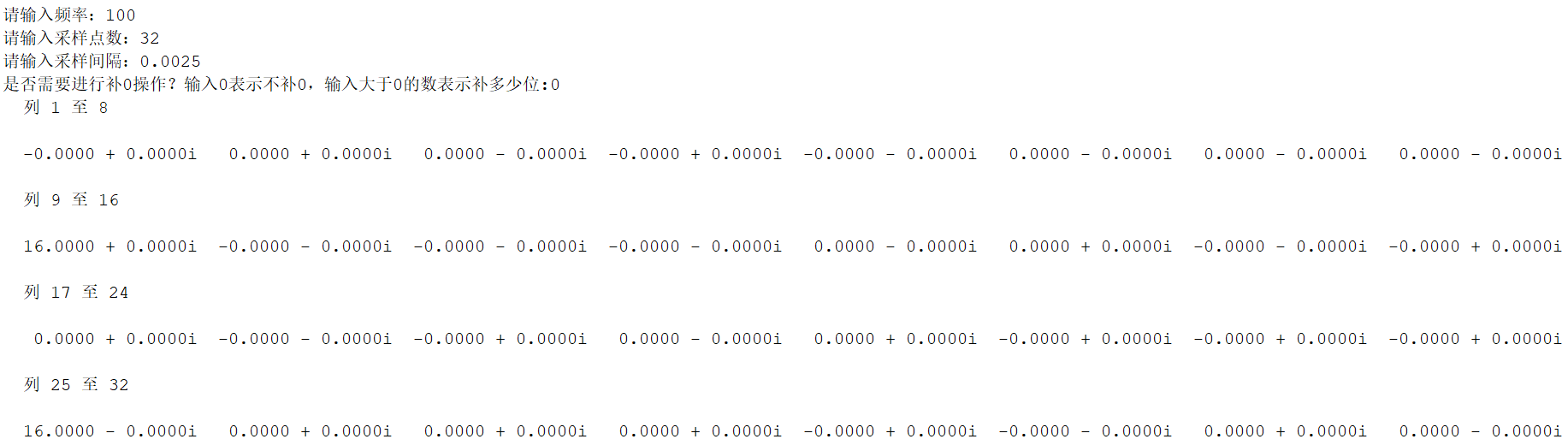
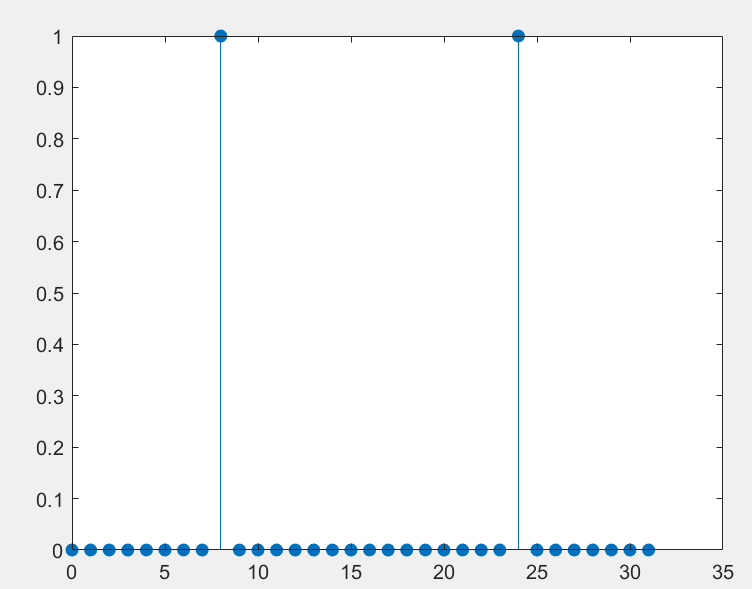
array(i) = array(i) / max;

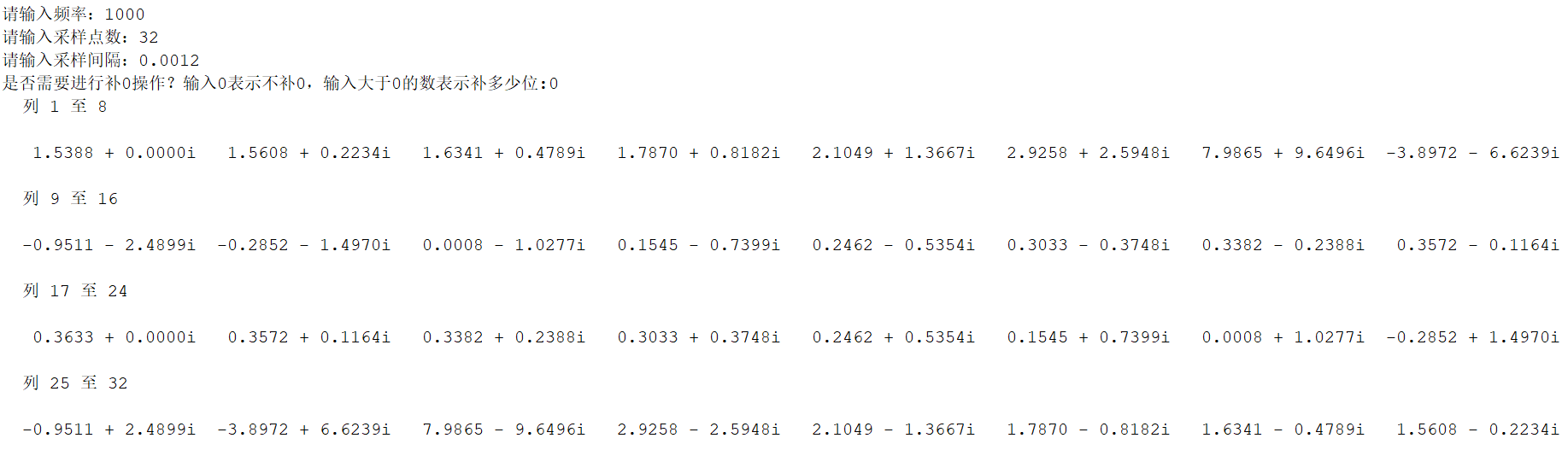
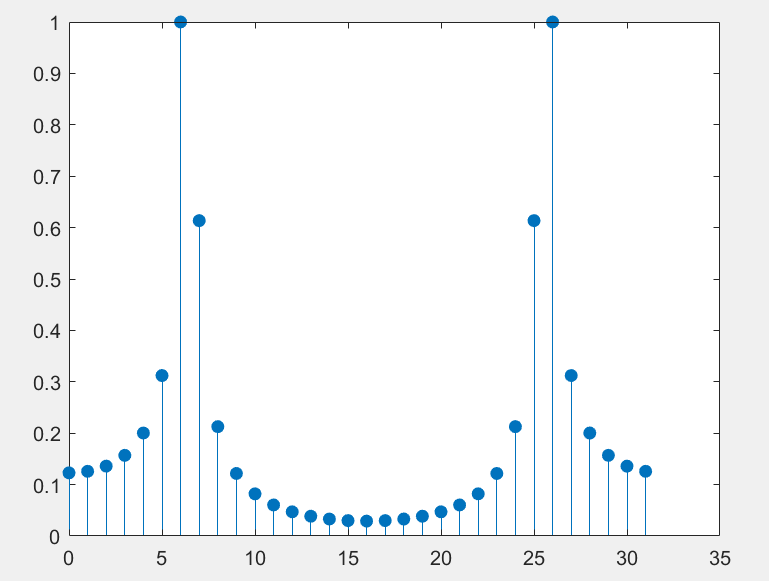
end

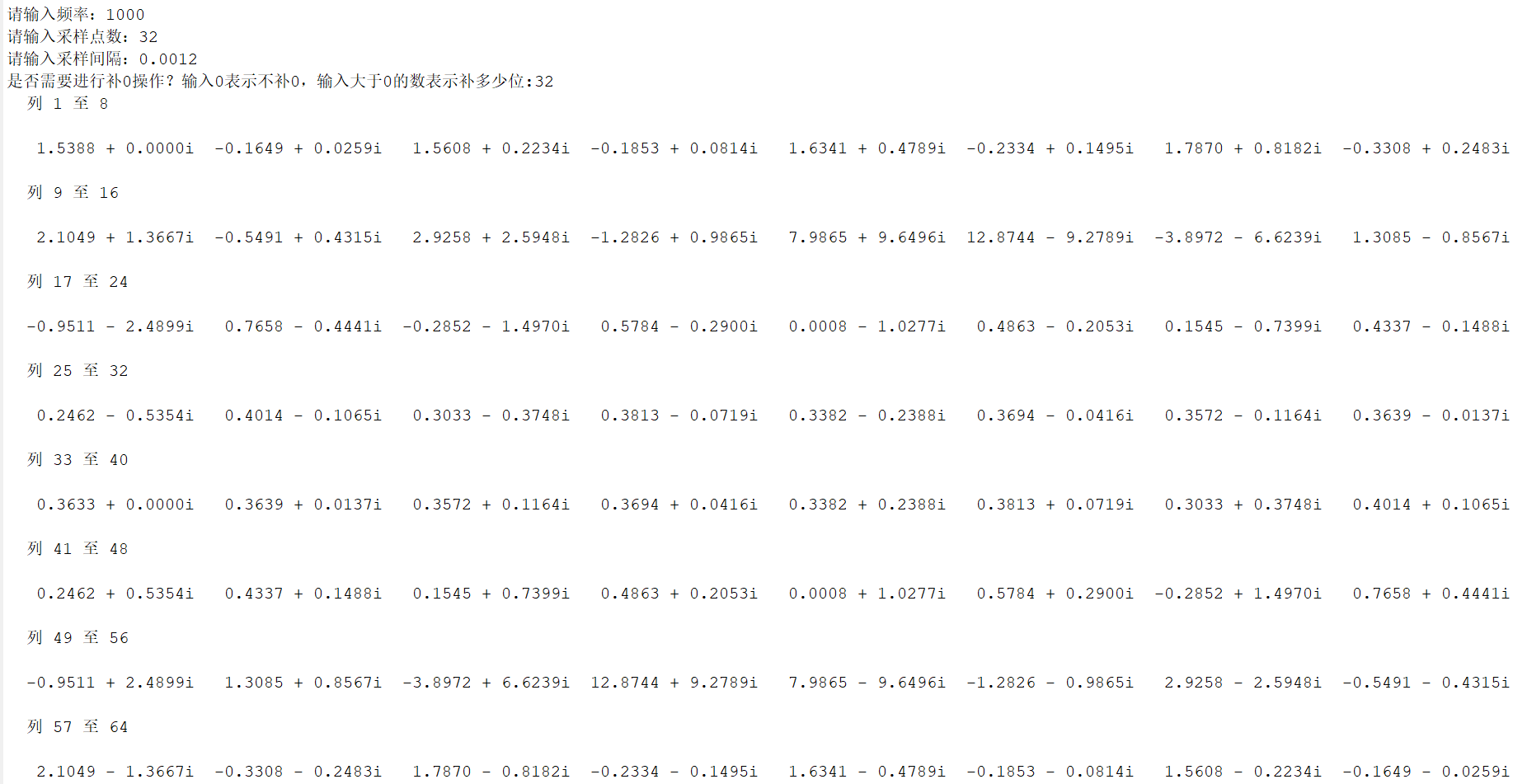
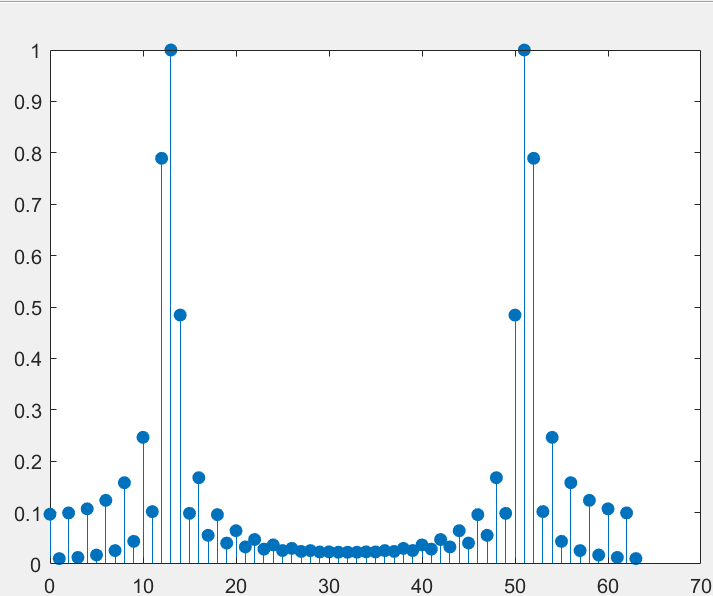
stem(k, array, "filled")

实验一：

实验二：

实验三：

实验四：

实验五：

结果分析：

在实验一、二、三中，由于采样点仅采到了sin函数的极值点和零点，导致该函数的特征并未被完全体现，使得进行快速傅里叶变换得到的频谱图出现了频谱泄露现象。

而在实验四和五中，采样频率小于信号最大频率的两倍，便产生了频谱混叠的现象，使得频谱图仍旧失真。

实验小结：

1. 深度了解快速傅里叶变换，并使用matlab进行了代码实现
2. 根据五个实验结果，了解到了频谱泄露和混叠产生的原因以及可能导致频谱出现的问题
3. 此后使用FFT算法时，应注意采样频率不应过小，同时选取采样点时应在能够保证反映原函数特征的情况下，进行采样的最优化。从而尽可能减少频谱混叠和泄露现象。同时根据实验五发现，可以选择在采样点后补0从而减少频谱泄露，也能够降低栅栏效应。