专业：自动化（控制）
姓名：朱少廷
学号：3200104845
日期：2022.4
地点：线上

浙江大学实验报告

课程名称：信号分析与处理　指导老师：张建良　实验类型：验证型

实验名称：离散傅立叶变换　成绩：　签名：

一、实验目的_

1、掌握 DFT 变换
2、掌握 DFT 性质
3、掌握利用 DFT 计算线性卷积
4、掌握快速傅立叶变换（FFT)

二、实验设备

1、PC 机。
2、Matlab2021a。

三、实验原理

离散傅里叶变换

首先要明白，离散傅里叶变换（Discrete Fourier Transform）和傅里叶变换做的事是一样，也是从时域到频域，然后它做的事是把一组复数 $\{x_n\} = x_0, x_1, \cdots, x_{N-1}$ 变换到另一组复数 $\{X_N\} = X_0, X_1, \cdots, X_{N-1}$ ：

$$
\begin{aligned}
X_k &= \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{j2\pi kn}{N}} \\
&= \sum_{n=0}^{N-1} x_n \cdot \left[\cos\left(\frac{2\pi kn}{N}\right) - j\sin\left(\frac{2\pi kn}{N}\right)\right]
\end{aligned}
$$

相应的，也有逆变换公式：

$$
x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j\frac{2\pi kn}{N}}
$$

装

订

线

四、预习要求（选做）

暂无

五、实验内容

1、实验操作方法和步骤

（1）求以下有限长离散时间信号的离散时间傅立叶变换 $X(e^{j\Omega})$

- 已知 $x(n)=\left(0.9e^{j\frac{\pi}{3}}\right)^n \quad 0\le n\le 10$
- 已知 $x(n)=2^n \quad -10\le n\le 10$

（2）已知序列 x(n)=cos(0.82πn) +2sin(πn)，0≤n≤50，绘制 x(n)及其离散傅立叶变换 X (k)的幅度、相位图。

（3）设 x(n)=sin(0.2πn)+randn(n)，0≤n≤N-1，其中，randn(n)为高斯白噪声。求出 N=4m，m=2,3,4 的 matlab 采用不同算法的执行时间。

（4）研究高密度频谱和高分辨率频谱。设有连续信号：

$$x(t)=\cos(2\pi\times 6.5\times 10^3 t)+\cos(2\pi\times 7\times 10^3 t)+\cos(2\pi\times 9\times 10^3 t)$$

以采样频率 fs=32kHz 对信号 x(t)采样，分析下列三种情况的幅频特性。

・采集数据长度 N=16 点，做 N=16 点的 DFT，并画出幅频特性。

・采集数据长度 N=16 点，补零到 256 点，做 N=256 点的 DFT，并画出幅频特性。

・采集数据长度 N=256 点，做 N=256 点的 DFT，并画出幅频特性。

观察三种不同频率特性图，分析和比较它们的特点以及形成的原因。

2、实验数据记录和处理

（1）Matlab 代码：

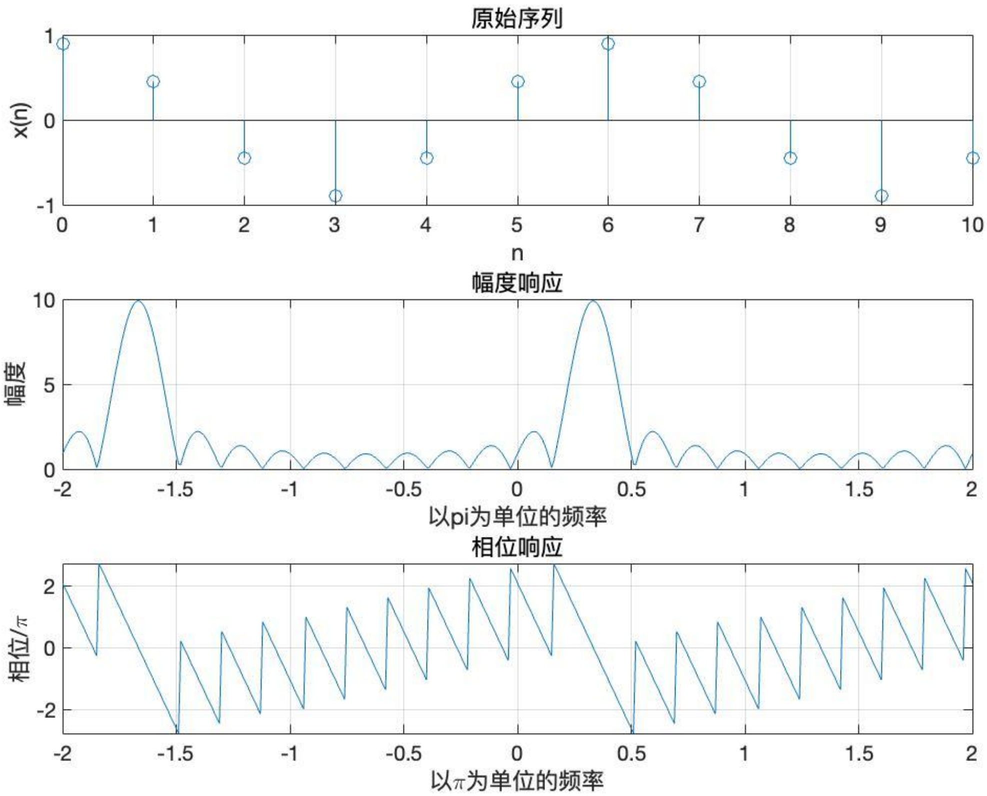
```
clc,clear
%% (1)
n=0:10;
x=0.9*exp(1j*pi/3.*n);
w=(-200:1:200)*pi/100;
X=x*(exp(-1i)).^(n'*w);
magX=abs(X);angX=angle(X);
subplot(3,1,1);stem(n,x);title('原始序列');grid;
ylabel('x(n)');xlabel('n');
subplot(3,1,2);plot(w/pi,magX);title('幅度响应');grid;
ylabel('幅度');xlabel('以 pi 为单位的频率');
subplot(3,1,3);plot(w/pi,angX);title('相位响应');grid;
ylabel('相位/\pi');xlabel('以\pi 为单位的频率');
```

实验名称： 离散傅立叶变换 姓名： 朱少廷 学号： 3200104845

```
%% (2)
n=-10:10;
x=(2).^n;
w=(-200:1:200)*pi/100;
X=x*(exp(-1i)).^(n'*w);
magX=abs(X);angX=angle(X);
subplot(3,1,1);stem(n,x);title('原始序列');grid;
ylabel('x(n)');xlabel('n');
subplot(3,1,2);plot(w/pi,magX);title('幅度响应');grid;
ylabel('幅度');xlabel('以 pi 为单位的频率');
subplot(3,1,3);plot(w/pi,angX);title('相位响应');grid;
ylabel('相位/\pi');xlabel('以\pi 为单位的频率');
```

图像结果：

第一个函数：

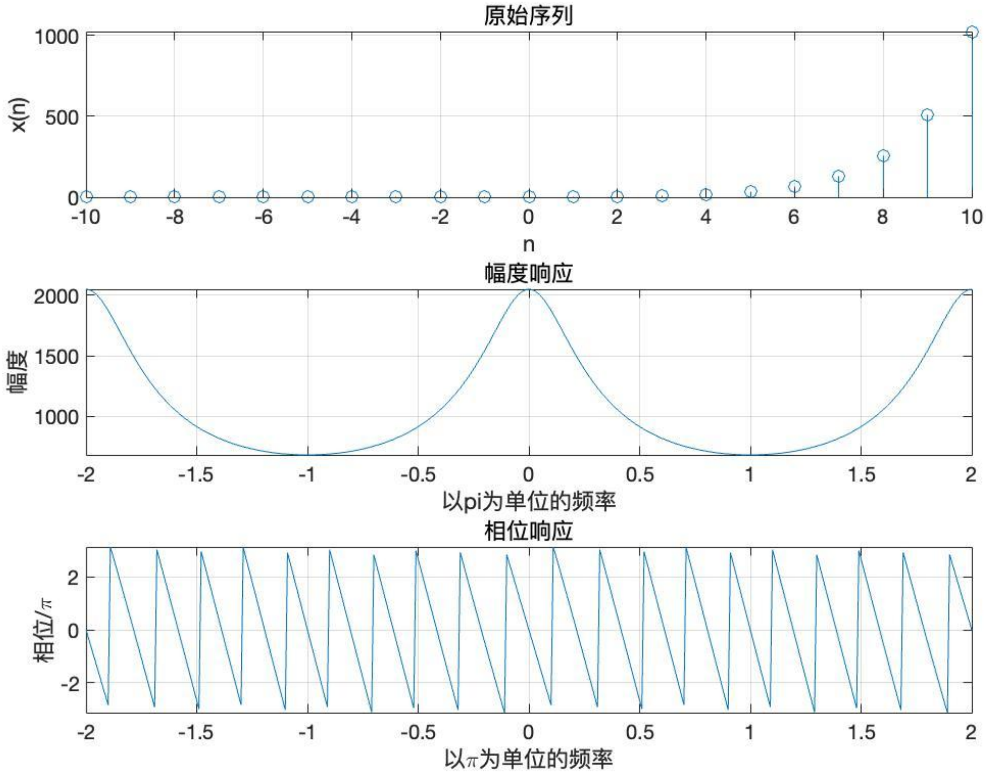


装

订

线

第二个函数：



（2）Matlab 代码：

```
clf,clc
N=50;
n=0:N-1;
xn=cos(0.82*pi*n)+2*sin(pi*n);
Xk=dft(xn,N);
magXk=abs(Xk);
angleXk=angle(Xk);
figure(1);
plot(xn);
xlabel('n');
ylabel('x(n)');
title('x(n)N=50');
figure(2);
k=0:length(magXk)-1;
plot(k,magXk);
xlabel('k');
ylabel('|X(k)|');
title('|X(k)|N=50');
```

装

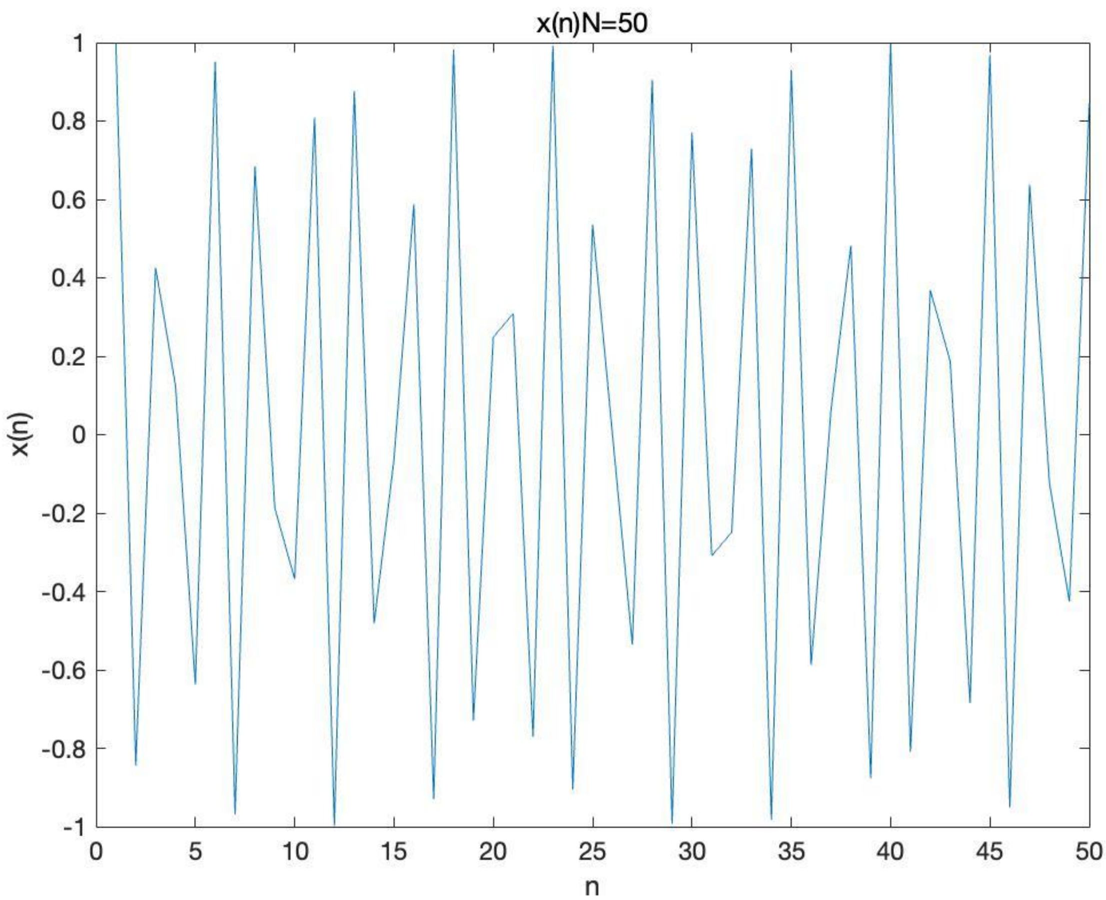
订

线

```
figure(3);
plot(k,angleXk);
xlabel('k');
ylabel('angle(X(k))');
title('Angle(X(k))N=50');

function Xk=dft(xn,N)
n=0:N-1;
k=n;
Wn=exp(-1j*2*pi/N);
nk=n'*k;
Wnnk=Wn.^nk;
Xk=xn*Wnnk;
end
```

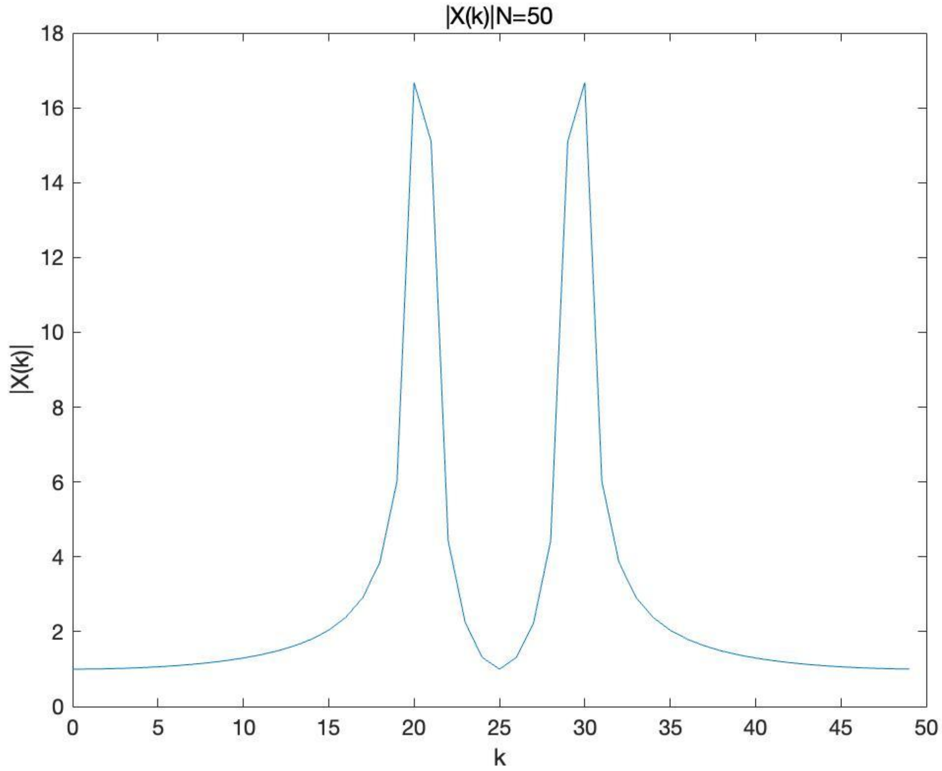
图片结果：

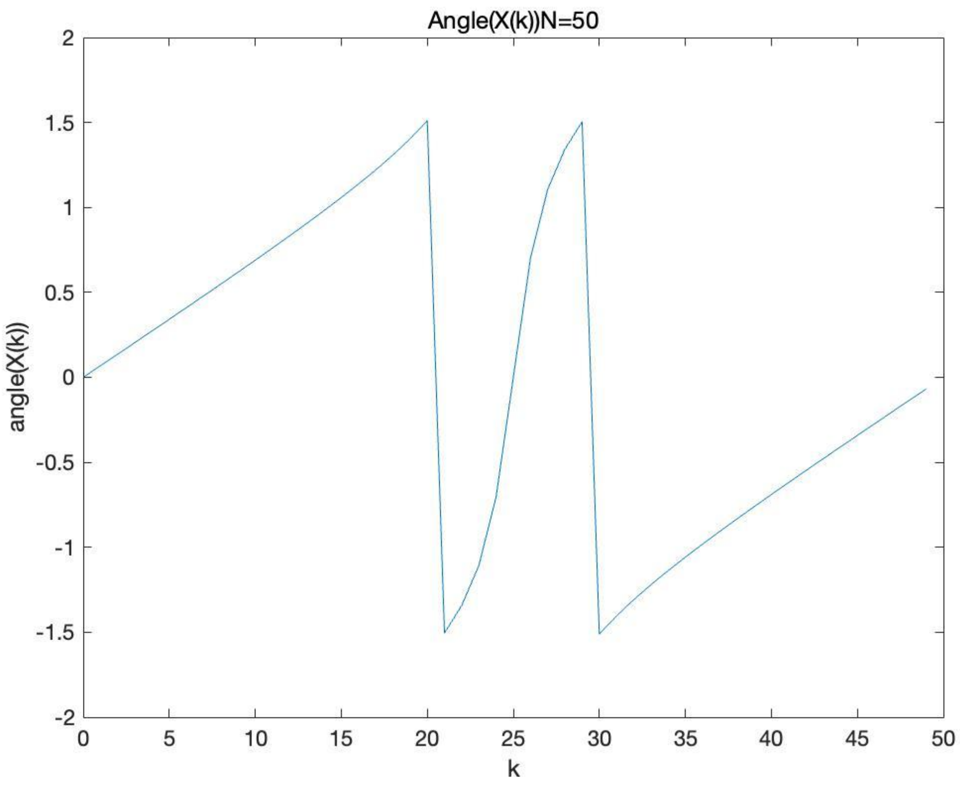


装

订

线


|X(k)|N=50
|X(k)|
k
0
2
4
6
8
10
12
14
16
18
0
5
10
15
20
25
30
35
40
45
50


Angle(X(k))N=50
angle(X(k))
k
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
0
5
10
15
20
25
30
35
40
45
50

（3）Matlab 代码：

```
clf,clc
N=4^3;
n=0:N-1;
xn=sin(0.2*pi*n)+randn(1,N);
disp('N=4^3【上面为 dft 时间，中间为 fft 时间，下面为 dtft 时间】')
tic
Xk=dft(xn,N);
toc
tic
Xf=fft(xn);
toc
tic
k=0:1000;
w=(pi/1000)*k;
X=xn*(exp(-1i*pi/1000)).^(n'*k);
toc

function Xk=dft(xn,N)
n=0:N-1;
k=n;
Wn=exp(-1j*2*pi/N);
nk=n'*k;
Wnnk=Wn.^nk;
Xk=xn*Wnnk;
End
```

运行结果：

```
N=4^2【上面为dft时间，中间为fft时间，下面为dtft时间】
历时 0.000359 秒。
历时 0.000069 秒。
历时 0.001591 秒。

N=4^3【上面为dft时间，中间为fft时间，下面为dtft时间】
历时 0.000483 秒。
历时 0.000019 秒。
历时 0.005370 秒。
```

装

订

线

N=4^4【上面为dft时间，中间为fft时间，下面为dtft时间】
历时 0.007720 秒。
历时 0.000736 秒。
历时 0.026565 秒。

可以看出，dft 的时间远大于 fft 的时间，并且 dtft 的时间在三者之中最长。因此，利用 fft 简化 dft 的计算非常有效。

（4）Matlab 代码：

```
clf;
% fs=100;
fs=32000;
Ndata=16;
N=16;
n=0:Ndata-1;
t=n/fs;
x=cos(2*pi*6.5*1000*t)+cos(2*pi*7*1000*t)+cos(2*pi*9*1000*t);
y=fft(x,N);
mag=abs(y);
f=(0:N-1)*fs/N;
subplot(2,2,1);plot(f(1:N/2),mag(1:N/2)*2/N);
xlabel('频率/Hz');
ylabel('振幅');
title('真实长度16，fft长度30');
grid;

% fs=100;
Ndata=16;
n=0:Ndata-1;
N=256;
t=n/fs;
x=cos(2*pi*6.5*1000*t)+cos(2*pi*7*1000*t)+cos(2*pi*9*1000*t);
y=fft(x,N);
mag=abs(y);
f=(0:N-1)*fs/N;
subplot(2,2,2);plot(f(1:N/2),mag(1:N/2)*2/N);
xlabel('频率/Hz');
ylabel('振幅');
title('真实长度16，fft长度256');
grid;
```

```
% fs=100;
```

装

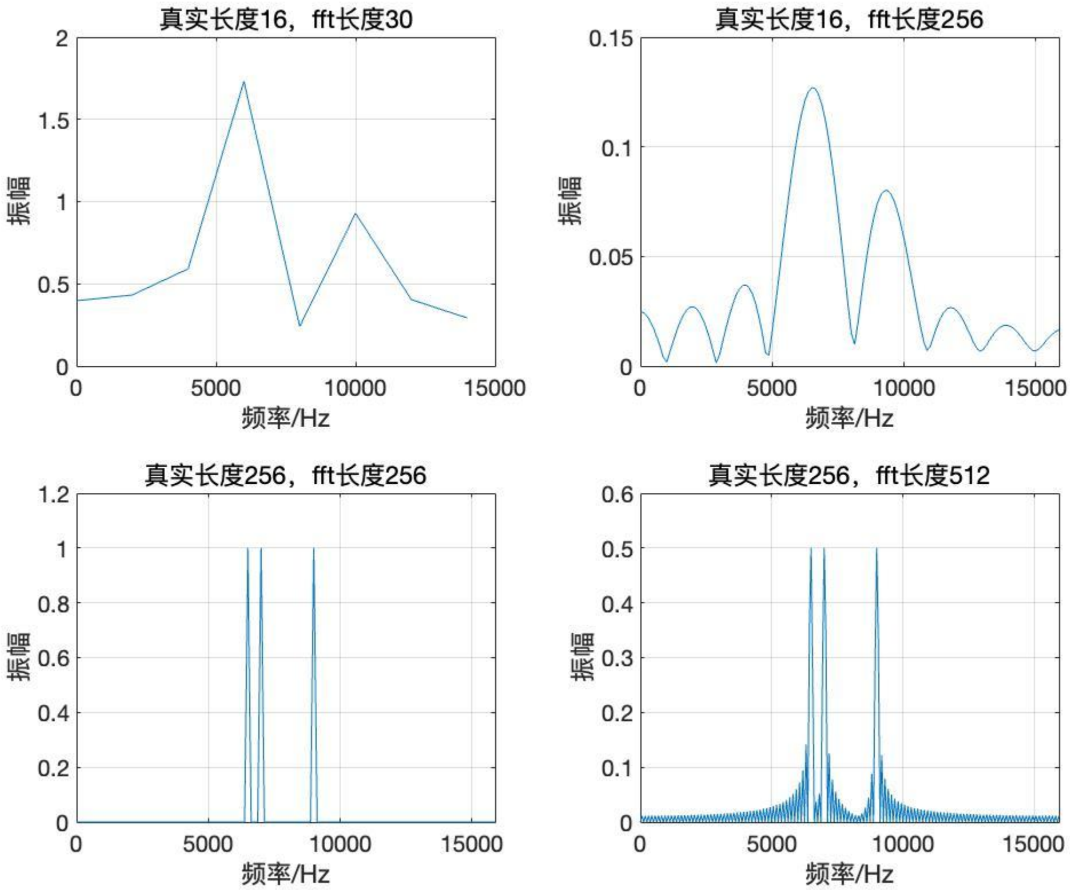
订

线

```
Ndata=256;
n=0:Ndata-1;
N=256;
t=n/fs;
x=cos(2*pi*6.5*1000*t)+cos(2*pi*7*1000*t)+cos(2*pi*9*1000*t);
y=fft(x,N);
mag=abs(y);
f=(0:N-1)*fs/N;
subplot(2,2,3);plot(f(1:N/2),mag(1:N/2)*2/N);
xlabel('频率/Hz');
ylabel('振幅');
title('真实长度256, fft长度256');
grid;
```

```
% fs=100;
Ndata=256;
n=0:Ndata-1;
N=512;
t=n/fs;
x=cos(2*pi*6.5*1000*t)+cos(2*pi*7*1000*t)+cos(2*pi*9*1000*t);
y=fft(x,N);
mag=abs(y);
f=(0:N-1)*fs/N;
subplot(2,2,4);plot(f(1:N/2),mag(1:N/2)*2/N);
xlabel('频率/Hz');
ylabel('振幅');
title('真实长度256, fft长度512');
grid;
```

图片结果：



当数据个数和 FFT 采用的数据个数均为 16 时，频率分辨率较低，但没有因为添零而导致的其他频率部分。当 FFT 采用的数据个数补零到 256 点时，由于添了零，导致出现了其他频率，并且其振幅因为加了 0 而减小了一部分。

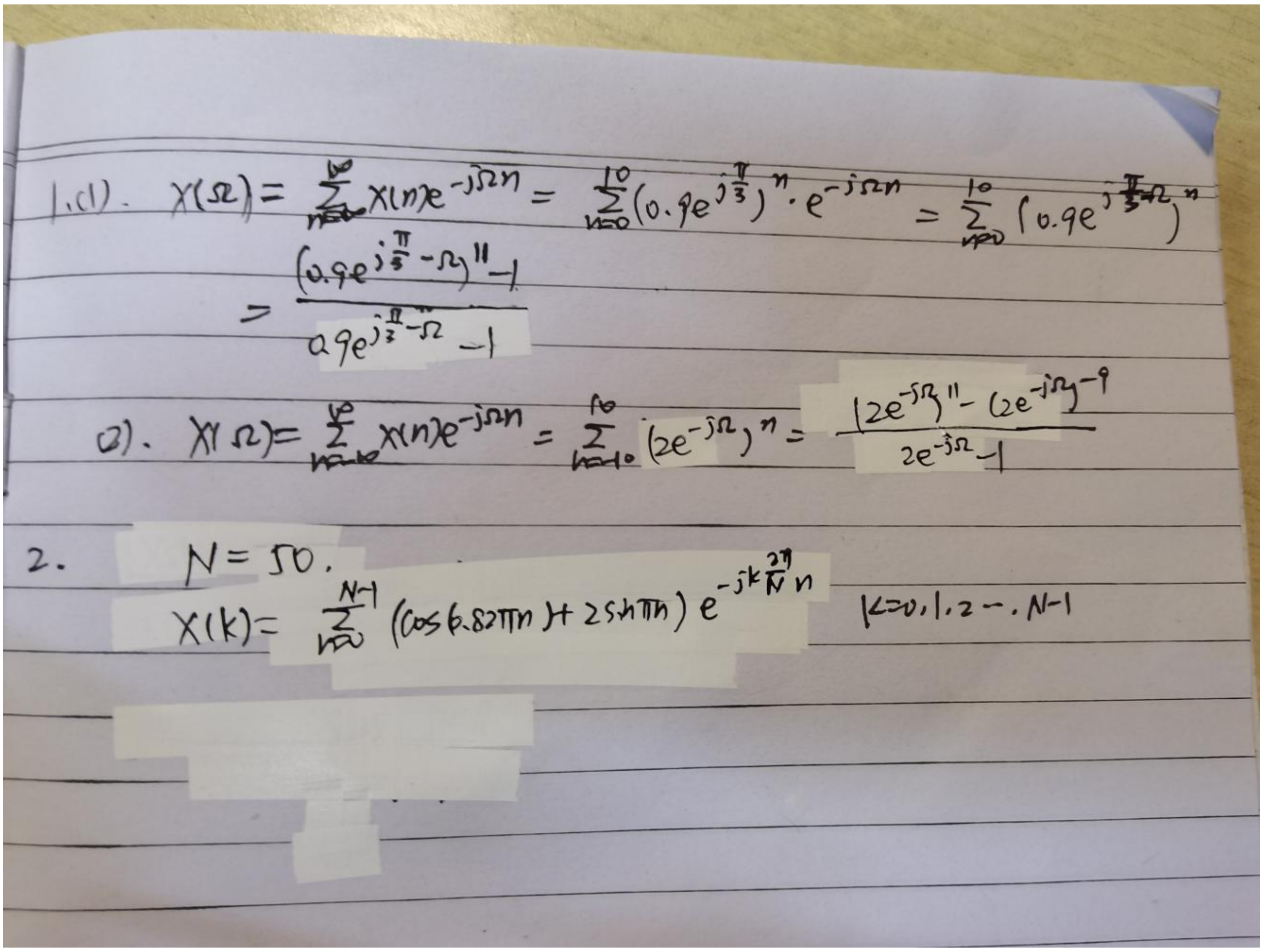
当数据个数和 FFT 采用的个数均为 256 是，频率分辨率很高，可以在图中明显看出有 6500、7000、9000 三种频率；而当 FFT 采用的数据个数补零到 512 点时，和上述类似，出现了其他频率，并且其振幅因为加了 0 而减小了一部分。

六、实验总结

1、实验结果与分析

进行理论计算：

实验名称：离散傅立叶变换 姓名：朱少廷 学号：3200104845



1.(1). $X(\Omega)=\sum_{n=0}^{10}x(n)e^{-j\Omega n}=\sum_{n=0}^{10}(0.9e^{j\frac{\pi}{3}})^{n}\cdot e^{-j\Omega n}=\sum_{n=0}^{10}(0.9e^{j(\frac{\pi}{3}-\Omega)})^{n}$

$$=\frac{(0.9e^{j(\frac{\pi}{3}-\Omega)})^{11}-1}{0.9e^{j(\frac{\pi}{3}-\Omega)}-1}$$

(2). $X(\Omega)=\sum_{n=-10}^{10}x(n)e^{-j\Omega n}=\sum_{n=-10}^{10}(2e^{-j\Omega})^{n}=\frac{(2e^{-j\Omega})^{11}-(2e^{-j\Omega})^{-9}}{2e^{-j\Omega}-1}$

2. $N=50$.

$X(k)=\sum_{n=0}^{N-1}(\cos(0.82\pi n)+2\sin\pi n)e^{-jk\frac{2\pi}{N}n}$ $k=0,1,2\cdots,N-1$

验证了实验结果。

2、讨论、心得

通过本次实验，我复习了离散傅里叶变换，学会了 DFT、FFT、DTFT 等常用离散傅里叶变换的 Matlab 实现方法，并对具体问题进行了实际变换操作。